ICT in Business and the Public Sector

Explainable AI Techniques for ML-based Financial Forecasting

Rose Browne
Student no: S1652834

Supervisors:
1^{st} supervisor: Dr. G.J. Ramackers
2^{nd} supervisor: Prof.dr.ir. J.M.W. Visser

Company supervisors:
K. Vu
B. Eekels

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Borhweg 1
2333 CA, Leiden
The Netherlands

June 28, 2021

Abstract

The lack of transparency found in many Machine Learning (ML) models results in a lack of trust in and understanding of ML models, and impedes the successful implementation of ML applications in many organisations. Explainable Artificial Intelligence (XAI) is focused on overcoming these issues by making ML models explainable without compromising their performance. The application of XAI techniques depends to a great extent on the specific context in which they are used. Firstly, from a technical perspective, the specific type of ML models used determine which XAI techniques are applicable. Secondly, the users of an AI system determine whether the chosen XAI techniques are successful at explaining the system in question. This research focuses on the effectiveness of XAI techniques in the domain of Financial Forecasting. Predictive analytical tools for Financial Forecasting are considered to offer great benefits for both Finance departments, as well as for organizations as a whole. We explore which XAI techniques are suitable for the ML models typically used in Financial Forecasting. These techniques then form the basis for a prototype implementation demonstrating their suitability. Subsequently, the qualitative and quantitative effectiveness of these techniques is validated by using the prototype in an extensive experimental set up. This experiment measures the increase in the level of trust in and understandability of Financial Forecasting solutions, the quality of insights gained, and the efficiency with which insights are gained. Furthermore, we measured the explanation satisfaction for each of the individual XAI techniques selected. It was found that the use of XAI techniques significantly improves both trust and understandability for ML applications in Financial Forecasting. Furthermore, we found a small, positive effect of the use of XAI on the quality of the insights gained. Overall, we conclude that the use of XAI techniques significantly addresses the black-box issue of ML applications in the context of Financial Forecasting. With respect to the provision of insights for Financial Forecasting solutions, we conclude that more research is needed to validate whether the use of XAI techniques can significantly increase the quality of insights gained.

Acknowledgements

I owe great thanks to a number of people for their participation in this research. A great thank you to the people at Deloitte for offering me the possibility to conduct my thesis internship within their Finance & Performance team. Their network has allowed me to conduct many interesting conversations with Finance professionals from the F&P team that helped form the direction of this thesis. In particular, I would like to thank Eric Merrill, the initiator behind and project leader of PrecisionView™, for sharing with me his knowledge on Financial Forecasting and the details on PrecisionView™ that enabled me to develop the XAI prototype proposed in this research. Also, my deepest appreciation to Kristi Blaisdell and Michelle Lahaije for their help and assistance in improving the developed dashboard from a visual perspective. Furthermore, I would like to sincerely thank José Bulder, Ewout Noyens and Ramon Vermeer from ABN Amro, Marcel Agterberg, Marc Cats and Jacob de Glee from Rabobank and Reynier van Voorst tot Voorst, Alie van Davelaar, Karan Gandhi, Pietro Moradei, Ferry Terpelle, Ayham Alkwai and Eric Merrill from Deloitte for their participation in the validation experiment of this research.

Furthermore, I would like to express my great gratitude to three people in particular, who not only contributed to the research, but also taught me some important personal lessons. My company supervisors, Khanh Vu and Bram Eekels and my first supervisor Dr. Guus Ramackers, thank you for your valuable feedback and extensive commitment in this research. But more importantly, thank you for your understanding during difficult times and teaching me that asking for help is not a bad thing. Also, thank you to Prof.dr.ir. Joost Visser for his valuable feedback in the early phases of this research and inspiring the idea of feature grouping, discussed in the conclusion. Lastly, I would like to thank my family and friends for offering the much needed distraction from time to time. A special thanks in particular to my boyfriend Casper for always making me laugh. You are without a doubt the most patient person I know!

Contents

1 Introduction

In this section we provide some background information on Machine Learning and Explainable Artificial Intelligence and discuss the problem statement. Next, we outline our research with a research statement and define the scope. Lastly, we provide an overview of the remained of this thesis.

1.1 Background

Artificial Intelligence (AI) and its subfield Machine Learning (ML) are emerging technologies that find increasingly more applications in both the industry and the public sector. AI and ML have proven to be powerful technologies to automate tasks that previously required the intelligence of human beings [BM17]. However, the implementation of these techniques also comes with difficulties. ML models in particular are often perceived as black boxes [AB18]. It is unclear to the users of an ML model how the model works and why it arrived at a certain decision, prediction or conclusion. This leads to trust issues and negatively impacts the acceptance of ML [Mol20], which creates barriers for their effective deployment in many organisations. Choosing not to make use of ML technologies can be a missed opportunity for companies because it offers a number of great advantages. Firstly, an important benefit of ML is that it can make the part of a job still performed by humans more valuable and satisfying. This is because ML frequently replaces only part of a person's job, allowing that person to focus on the part that humans do best [DR18, Pis18]. Furthermore, automating tasks that previously required a human being increases the efficiency with which that task is performed, as a machine is often much faster [WS19, BM17]. As a result of increased efficiency, the use of ML can also reduce the costs of performing a task. Lastly, the use of ML can improve the safety or accuracy of a task because it makes fewer mistakes in performing them [WS19]. Consider for example Google's research into self-driving cars that appear to be safer than vehicles driven by humans [TK17]. However, ML applications also suffer from a number of disadvantages. In addition to the trust issue mentioned above, the black-box characteristic of ML applications can give rise to other issues as well. Firstly, if the inner workings of a model are unclear to its users, then the level of understanding that users have of a model is negatively impacted as well. Secondly, as ML models are frequently trained using historic data that may contain human bias, this bias can be mimicked by those models. This becomes especially troublesome when ML models are black-boxes, causing the bias to remain undiscovered. This algorithmic bias can have both social and legal implications when, for example, it leads to unfair decision making that disadvantages individuals based on legally protected traits such as age, religion, race or gender. Given the aforementioned advantages that ML can bring, it is important to overcome these issues relating to ML in order to successfully implement ML applications.

Explainable Artificial Intelligence (XAI) is a research field that recently gained much attention and is aimed at solving this black-box problem. More specifically, XAI is a set of techniques used to make models explainable without compromising their performance [AB18]. To validate whether a proposed XAI technique is successful, two aspects can be verified. Firstly, research can verify whether an XAI technique is applicable to a specific type of model, e.g. does it work for Neural Networks, Support Vector Machines, Tree Ensembles, etcetera. Secondly, the XAI's effectiveness at explaining a model amongst a certain target audience can be tested. The reason for validation of an XAI technique amongst its target audience is that the desiderata for an explanation largely depend on the users of that explanation [GMR+18, ADRDS+20]. Although the applicability to different models, as well as the effectiveness on target audiences amongst several industries have been tested for many XAI techniques, the use of XAI in the specific context of Financial Forecasting has largely been left uninvestigated. Financial Forecasting in the context of the Finance Function is defined as the prediction of future values of financial line items. The task of the Finance Function comprises specialist services such as auditing and tax, the provision of transactional financial data, and management accounting activities including decision support, planning and budgeting and information analysis [Tha12]. Financial Forecasting assists in the later, by aiding Finance and Business controllers in the delivery of this decision-making support and financial analysis [Sam15]. Financial Forecasting applications typically use time-series data [JBL19, HGP11]. However, research into the use of XAI techniques for time-series data is currently limited [SAEA+19, KRPG18]. The small amount of research that is available focusses primarily on financial classification tasks [SAEA+19, KRPG18], whereas Financial Forecasting constitutes a regression task. Other research proposes novel, model-specific XAI methods for time-series regression models [IGCB20], rather than validating the established model-agnostic XAI techniques. Not only are time-series models underrepresented in XAI research, but research into its effectiveness amongst finance professionals, the target audience of Financial Forecasting solutions, is lacking as well.

1.2 Problem Statement

It is important to investigate XAI techniques in the specific context of Financial Forecasting. Apart from the general advantages that ML can offer, there is another important driver, namely Finance Business Partnering (FBP). FBP is defined as Finance's role in delivering decision-making support throughout the business [CD13, ZNGL20] through the provision of information and insights [Qui14, ZNGL20]. Although there has long been a demand for FBP [vdV14], today this demand has become more urgent as complexity and uncertainty grow amongst many industries [Tha12]. Decision making now requires managers to obtain timely, more and better information in a quicker fashion than before to deal with this uncertainty. The current Covid-19 pandemic is a good example that emphasizes the importance of this. As a result the business increasingly relies on the Finance Function for the provision of this information [Tha12]. A number of strongly related requirements to successfully implement FBP are emphasized in the literature. Namely, improved systems, a reduction in manual activities, improved data analytics and a stronger future focus [ICA18, CGM15, CIM09, Sch19, Ven15, Oli91, CD13]. Quinn states that the aspects of the future role of the Financial Controller can be summarised in a single word: analysis [Qui14]. This means that tools assisting in this analysis will become of greater importance in order for financials to not only gather and process the large amounts of available data, but also to provide more qualitative data analysis. Therefore, it is important to overcome the aforementioned pitfalls relating to ML driven data analytics, as well as optimize its effectiveness. Predictive analytics in particular is said to assist in the reduction of manual activities [Sch19], improve data analytics in general and form the basis for better decision-making [Del14, PWC17]. This in turn enables Finance to anticipate digital disruption, measure performance and respond swiftly [You15], all in support of the business. Concurrently, the amount of available data has increased significantly, further strengthening the importance of data analytics in Finance [Qui14].

Hence, predictive analytics capabilities, such as Financial Forecasting solutions, promote a forward-looking approach, allow the Finance Function to reduce manual activities, improve data analytics and thereby improve the overall decision-making process. However, predictive analytics tools, and Financial Forecasting solutions in particular, frequently make use of ML techniques for the creation of the complex forecasting models needed to model the VUCA world. As discussed in section 2, the use of ML techniques can give rise to several issues that obstruct the successful implementation of these solutions. XAI helps overcome these issues, but its applicability to models commonly used in Financial Forecasting, as well as its effectiveness amongst the target audience of Financial Forecasting solutions has largely been left uninvestigated. Therefore, it is important that we investigate the use of XAI in Financial Forecasting.

1.3 Research Scope

In this research, we will investigate the use of XAI in the specific context of Financial Forecasting. In particular, we are interested in a number of factors. Firstly, we want to investigate the applicability of XAI techniques to Financial Forecasting from a technical perspective, i.e. which techniques are suitable for the time-series models typically used in Financial Forecasting? Secondly, we are interested in whether XAI increases the level of trust and understandability that Finance professionals have in Financial Forecasting solutions and thereby contributes to solving the black-box problem relating to ML. Lastly, we want to verify the expectation that XAI increases the effectiveness of Financial Forecasting solutions by improving the derivation of insights and enhancing the decision-making process. The motivation for this last assumption is that XAI techniques provide detailed information on the relations between the independent and dependent variables. This is particularly useful in situations where forecasts and the variables on which they depend are an important source for decision making, such as in Finance. Insight in these relations supports the exploration of alternative scenarios and helps signal potential opportunities and threats, thereby enhancing the decision-making process.

To this end, the following research question was formulated: *Does the use of XAI solve the black-box problem and increase the effectiveness of ML-based Financial Forecasting?* In order to answer this, the following hypotheses were derived.

H1. The use of XAI helps overcome the black-box problem by

- **H1a.** increasing the understandability of Financial Forecasting solutions.
- **H1b.** increasing the trust in Financial Forecasting solutions.

H2. XAI increases the effectiveness of Financial Forecasting solutions by improving the derivation of insights by

- **H2a.** increasing the quality of the insights gained.
- **H2b.** increasing the efficiency with which insights are gained.

The academic contribution of this research is twofold. Firstly, we seek to investigate the applicability of XAI techniques for Financial Forecasting from a technical perspective, in terms of which XAI techniques can be applied to the specific types of time-series models typically used for Financial Forecasting. Secondly, it aims to reduce the gap between the ML research community and the Finance department as a means to enable business partnering. This gap is said to hinder the deployment of the latest ML developments [ADRDS+20] within the financial sector. A survey performed by Deloitte amongst CFOs in 2019 indeed found that 64% states their Finance Function is only slightly or not at all prepared regarding the implementation of AI solutions and 77% are currently not using cognitive tools of any sort [Pro19]. We aim to bridge this gap by investigating the use of XAI techniques in the specific context of Finance. We will do so by developing a prototype and verify its effectiveness amongst a target audience of Finance professionals. This prototype is based on an existing Financial Forecasting solution, called PrecisionView™ [MA21], which we will extend with multiple XAI techniques. Although the prototype is based on an existing tool, the approach taken in its development is aimed to be as generic as possible, so it can easily be applied to and implemented in any ML-based forecasting tool within the Finance domain.

1.4 Outline

As briefly mentioned above, a combination of two research methodologies is applied in this research. The main methodology is design science research, as the research delivers a conceptual framework and corresponding prototype describing and show-casing the implementation of XAI techniques for Financial Forecasting solutions. To this end, we firstly performed a review of the literature to investigate which XAI techniques are potentially suitable for Financial Forecasting solutions. Next, we tested the selected techniques on a small, open-source dataset to validate their applicability to such data. After having selected the most suitable XAI techniques, they were implemented in an existing Financial Forecasting solution to build a prototype used for the validation of aforementioned hypotheses. The existing Financial Forecasting solution used for this is called PrecisionView™ and was developed by Deloitte U.S. The main advantages of using an existing tool is that it assists in the validation of the applied XAI techniques by providing the participants with a clear comparison of an ML-based Financial Forecasting solution with and without XAI. Furthermore, it provides some level of insights into the data structure as well as ML models typically used in Financial Forecasting, although it should be noted that these may differ between Financial Forecasting tools.

Next, we collected feedback on the intermediate versions of the prototype amongst a small group of domain experts and fine-tuned it accordingly. Once the prototype was finalized, a combination of quantitative and qualitative research methodologies was used to validate the hypotheses stated in section 1.3. The validation method entailed a combination of survey questions and two case study experiments, relating to a Financial Forecasting solution without and with XAI respectively. Both the survey questions and the case studies were based on a live demonstration of the prototype. The participants that partook in this validation are Finance professionals who would be affected by the implementation of the tool developed in this research or similar tools.

The remainder of this thesis is structured as follows. First, section 2 provides an overview of the related work on XAI, the use of ML in Finance, as well as Financial Forecasting in particular and the importance of such predictive analytics to enable FBP. Next, in section 3, we describe from a high level perspective, the steps involved in the development of our XAI prototype for Financial Forecasting solutions in general, focusing on the generic issues and specifications of our prototype.

In section 4 we elaborate on the specifics of PrecisionView™, the Financial Forecasting solution used in this research, and illustrate how the steps in section 3 were applied to this specific tool. Next, in section 5 we elaborate on the experimental setup for the validation of the prototype and provide the results thereof, followed by a discussion. Finally, in section 6 we provide a summary of the research and discuss its generalisability and limitations as well as provide recommendations for future work.

2 Background

As discussed in the introduction, the use of ML can offer companies great advantages. However, the successful implementation of ML applications is often obstructed by the black-box issue relating to ML. XAI can help overcome this issue, but its application in the specific context of Finance has largely been left uninvestigated. It is important to do so, as the use of XAI is context-dependent. This means that in order to determine which XAI techniques are best suited for Finance, its applicability to financial data and models, and its effectiveness amongst Finance's target audience needs to be validated. Furthermore, the importance of ML and hence the use of XAI in Finance is strengthened further by the concept of Finance Business Partnering, which illustrates the benefits that predictive analytics can have in Finance. Therefore, we will first define this financial context and discuss why FBP forms such an important driver for the use of XAI. Next, we discuss the concept behind ML and its applications in Finance, as well as elaborate on the so-called black-box problem. Then, in section 2.3, we discuss what Financial Forecasting entails, provide an overview of the existing software solutions and elaborate on the tool used in this research, PrecisionView™. Lastly, we describe the different approaches, methods and explanations in the area of XAI, and the current state of research into its use in Finance.

2.1 Finance

As mentioned in section 1.3, one of the aims of this research is to bridge the gap between the ML research community and Finance by investigating the use of XAI techniques in the specific context of Finance. Therefore, in this section, we will first discuss what we mean when referring to the "financial context". Next, we will elaborate on the definition of Finance Business Partnering, the activities that the role entails, why the demand for it increased recently and the obstacles in implementing it, in order to illustrate why it is an important driver for the use of Financial Forecasting.

2.1.1 The Finance Function

Previously, research has been conducted into the use of ML, and to a lesser extent the use of XAI, within the specific context of Finance. However, in the majority of these works, the financial context refers to something different than we are focusing on in this research. Finance can refer to either the Finance Function or the financial sector, also called the Financial Services Industry (FSI). In the context of our research we focus on the Finance Function, which indicates a functional area within a company concerned with the financial management of that company. Its task is to ".. provide transactional financial accounting (general ledger accounting, payables and receivables, external reporting), specialist services such as tax, internal auditing and finance systems and management accounting (decision support, planning and budgeting, information analysis).." [Tha12]. The FSI on the other hand is comprised of a range of businesses that offer financial services such as insurance companies, investment funds, stock brokerages, credit-card companies and banks. As we will see, much of the research on both ML and XAI in Finance is aimed at services in the financial sector, such as stock price prediction, insurance fraud detection or credit risk assessment. It is important to note the distinction between these two meanings of Finance in order to determine the relevance of previous XAI research in a financial context. Research concerning the technical aspects of XAI in the context of the FSI is to a limited extent relevant for the Finance Function as well. The data involved in ML applications for the FSI is similar to that of ML applications within the Finance Function, namely transactional time-series data. The models they use, on the other hand, can differ greatly since the purpose of their ML applications does as well. Furthermore, research into the effectiveness of XAI amongst the target audience in the FSI can not simply be extended to the target audience of the Finance Function. The knowledge of and tasks performed by a banker or a broker differ greatly from that of an accountant, business controller or CFO. In this research, we are interested in the use of XAI in the context of the Finance Function.

2.1.2 Finance Business Partnering

The concept of Finance Business Partnering emphasizes the importance of ML applications like predictive analytics and the use of XAI in Finance specifically.
Improved data and predictive analytics enables the Finance Function to meet a number of requirements deemed of importance for the successful implementation of FBP, as well as overcome some of the obstacles to FBP [Del14, PWC17, You15, Gin11]. The purpose of XAI is to improve both the level of trust in and understanding of these ML driven analytics solutions [ADRDS+20]. Therefore, we expect XAI enabled predictive analytics to further improve the successful implementation of FBP. In this section, we will discuss what these requirements and obstacles are and why FBP has gained much importance recently. Before doing so we will first provide a definition of FBP.

The literature on FBP extends beyond the scientific literature. Much information can be found in both managerial accountancy institute reports and company studies. In our review on FBP, we will use this grey literature as complementary to the academic literature. The reason for this is that, although they lack academic theory, they provide valuable insights on FBP and its implementation in practice due to the extensive network of accountants and companies that partake in their surveys [Tyn16]. More specifically, the sources include:

1. Company studies by Deloitte, KPMG, PwC and EY,
2. Managerial Accountancy Institutes reports by CIMA, ICAEW & CGMA, and
3. Academic literature

2.1.2.1 Definition

The majority of the academic sources define FBP as Finance's role in providing decision-making support on a strategic level. Oliver and Cooper & Dart both emphasize the importance of a forward-looking approach in doing so [Oli91, CD13]. Cooper & Dart not only include decision support on a strategic level but on an operational level as well. Providing information and insights is recognized as the main way to deliver this decision support [Qui14, ZNGL20]. Both Cooper & Dart and Quinn stress that FBP is not a replacement of the traditional role and tasks of a financial, but rather an extension. However, part of these traditional tasks is increasingly performed by information systems [Qui14]. The definition of FBP maintained by the company studies as well revolves around the delivery of decision-making support. Both Deloitte and PWC define FBP as the role Finance undertakes in supporting and challenging the business to ensure their strategic decision making delivers the desired value at an acceptable level of risk [Del13, PWC17]. In line with Cooper & Dart, KPMG not only includes strategic but also operational decision support. Furthermore, they also emphasize the importance of a forward-looking and commercial view, as well as to help articulate different opinions [icwC11]. Although EY does not give a concrete definition on the meaning of FBP, the title of their report "Partnering for Performance" [You15] suggests that driving business performance is, according to them, the main goal of a Finance Business Partner. The managerial accountancy institute reports define FBP as the provision of information and insights in order to inform and influence decision making and drive performance through opportunity and risk management with the ultimate objective of generating value for shareholders [CIM09] or all stakeholders [CGM15, ICA18]. Furthermore, both CIMA and CGMA stress that Finance Business Partnering begins after standard reports have been produced, taking the information produced by the accounting operations as their starting point. Based on these definitions we here define Finance Business Partnering as: Finance's role in improving decision-making, and supporting and challenging the business through the provision of insights to increase the business' performance.

2.1.2.2 Activities

Now that we have defined FBP, we can look into how a business partner achieves this objective of improved decision-making. As we will see, this decision-making support is delivered through a number of activities, including the timely identification of risks and opportunities, the provision of insights and the promotion of a more forward-looking approach. These activities illustrate the importance of improved data analytics in general and predictive analytics in particular.

- **Risk management and opportunity identification**
  Risk management and opportunity identification is recognized as an important activity for FBPs as it is crucial to timely identify and assess potential risks and opportunities in the current hypercompetitive business environment [Ven15]. Furthermore, as a consequence of this hypercompetitive and rapidly changing environment, plans can become outdated quickly and hence several alternative scenarios need to be developed and accounted for [Ven15]. Predictive analytics capabilities assist is this need for scenario planning as it enables financials to forecast the expected outcome of different possible scenarios.

- **Looking forward**
  Furthermore, Business Partners should take a more forward-looking approach [icwC11], or at least find a better balance between hindsight and foresight focus [You15]. Indeed, in a survey performed by Deloitte, it was found that 82% of finance executives surveyed assigned the highest priority to 'forecasting and demand planning' activities [Del19]. In his case study on implementing FBP within a design company, Tynkkynen as well highlights forecasting as one of the main activities of an FBP [Tyn16].

- **Providing insights**
  All sources stipulate the importance of providing insights as part of an FBP's role. The Finance function should enrich its financial systems with both internal and external data in order to transform into a strategic intelligence centre [Oli91]. According to CIMA, the provision of information to deliver decision support is "at the heart of the role"[CIM09]. The importance of providing insights is emphasized further by the amount of attention given

to the performance of data analysis activities. In fact, 85% of the respondents in KPMG's global CEO outlook survey say that analytics will increasingly drive profits as "applying financial data to achieve profitable growth is the greatest strategic value a CFO can bring to an organisation"[Wat17]. In addition to that, EY's report identifies data analytics as a means to anticipate digital disruption and respond quickly [You15]. Hence, data analytics supports the conduction of the aforementioned activity of risk and opportunity identification.

To summarize, the main task of an FBP is to provide decision-making support to improve the overall business' performance. This support is delivered through the identification of risks and opportunities, a forward looking approach and the provision of insights. Data and predictive analytics are an important enabler to gain these insights, and combined with a more forward-looking approach enables the timely identification of potential risks and opportunities.

2.1.2.3 Increasing Demand

We have looked into the definition of FBP, the activities it constitutes to achieve improved decision-making support and why data and predictive analytics capabilities are important for the successful execution of these activities. In this section we will look into why the demand for FBP and improved decision-making support, has increased so much recently in order to understand the importance of improved predictive analytics solutions, such as Financial Forecasting. Deloitte's, KPMG's and EY's reports all signal a greater demand from the business for Finance Business Partnering. In a survey amongst 75 senior Finance executives from U.K. based companies, it was found that 75.7% experienced an increase or significant increase in the demand for Business Partnering activities between 2012 and 2014 and 82.5% expected that demand to increase even further [Del14]. CEO's indicated to rely more and more on their CFOs for both decision making and strategy development [You15]. We found a number of reasons for this increasing reliance on the Finance Function.

- **The increasingly competitive and faster-changing business environment**
  The first reason for the increasing demand found throughout the literature is the increasingly competitive and faster-changing business environment and economic turmoil. The economic crisis of 2007-2008 has brought about a change in the requests and demands placed upon the Finance Function [Smi15, icwC11, You15], as economically difficult times increase the need for financial knowledge [vdV14]. It resulted in enhanced reputations for CFOs who were asked to step up and find ways to reduce costs [You15] and led many Finance teams to re-evaluate their role [icwC11]. Furthermore, the increasingly complicated and competitive business environment, commonly referred to as the VUCA (Volatility, Uncertainty, Complexity, Ambiguity) world, is said to have increased the demand for information to adapt and evolve timely [Smi15]. The use of ad-hoc analysis and forecasting to inform decision-making is required by the ambiguity presented by this VUCA world [CGM15].

- **A shift in focus from operational to business finance**
  Secondly, a shift in focus from operational to business finance is taking place [PWC17, Del19]. CFOs have extended their focus beyond the traditional scorekeeper role [You15]. According to CIMA, the reason for this is that the possibility for further cost reductions has shrunk and

hence Finance's focus has shifted to value creation and the protection of the interests of the shareholders instead [CIM09].

- **The increasing availability of data and the digital transformation**
  Thirdly, all sources point out the vast and increasing availability of data and digital transformation as drivers of the demand for FBP. The large amounts of data available nowadays contain valuable business intelligence, and financials are said to have the ability to translate these insights into business impacts [Qui14, PWC17] and deliver evidence-based decision-making [CIM09]. EY's report even claims business partnering should be made a priority if organisations want to succeed in using this vast amount of data for better decision-making [You16]. They find that by not only looking at internal data but also including external metrics, financials can provide better risk and performance management support.

  Furthermore, digital transformation in general is seen as an important driving force behind the demand for FBP, as it automates the recording and provision of data, allowing financials to move away from these more mundane tasks [Gin11, Qui14]. Apart from the opportunities that this presents to the Finance Function [Del12, Wat17], it also drives the demand for FBP, because it presents new threats that require the CEO to have business partners by its side more than ever [You15].

- **Finance's unique position**
  The majority of the factors behind the increasing demand for FBP mentioned above affect not only the Finance function but every department within a company. Yet it is Finance in particular who is put forward for the role of a business partner. Within the literature, a consensus exists regarding Finance's unique position that enables them to best fulfil this role. Firstly, they are said to have a central role within the business, which enables them to form close bonds with data analytics specialists and technology managers [Qui14]. They have an overview of the business and the ability to gain broader insights that are more difficult to obtain from the functional perspective of other departments [vdV14, CGM15]. Secondly, financials are said to be objective and known for their integrity, which allows them to challenge the business, ensure short term objectives are weighed up against long term objectives [CIM09], and promotes decision-making on the basis of relevant information and thorough analysis [CGM15]. This in turn means financials can protect the interests of all stakeholders [CGM15]. Lastly, Finance's technical skills and affinity with data and analysis ensures the collection and interpretation of the correct data and makes them well-positioned to unlock the potential value of BI [CIM09] that drives decision-making [Qui14]. In addition to that, management accountants already have access to a wide range of data sources that can further enrich data analysis as well as ensure its objectivity [CIM09].

All in all, the drivers behind the increasing demand for FBP and decision-making support emphasize the importance of data analytics, and predictive analytics in particular, within Finance. The use of ad-hoc analysis and forecasting is required to deal with the ambiguity of the VUCA world and translate the vast amount of available data into valuable insights. The Finance Function is considered to be best positioned for this task because they have a central role in the business, are able to gain broader insights, are objective and have integrity, and possess the technical skills and affinity with data analysis that is required to do so.

2.1.2.4 Obstacles

We now know why data and predictive analytics capabilities are important for the Finance function to improve their decision-making support and successfully implement FBP. Nevertheless, many companies still fail at successfully implementing FBP for a number of reasons. Firstly, Finance's relationship with the business and the organisational structure frequently poses difficulties [ICA18, Del14, PWC17, Gin11, CD13]. Secondly, an unclear definition of FBP and the lack of an appropriate skill set are identified as common pitfalls [CGM15, Del13, CD13, vdV14, Ven15, CIM09]. However, the two most frequently mentioned obstacles again point at the lack of automation of repetitive tasks, but most importantly the need for improved data analytics in combination with a more forward-looking approach.

- **The focus on manual and repetitive tasks**
  Finance is said to lack focus on or understanding of value-adding activities. The Finance Function is spending most of its time on traditional finance and accounting activities such as the recording of transactions and the generation and distributions of reports, limiting the time available for value-adding activities [Ven15]. Indeed, 41% of the respondents in Deloitte's Business Partnering Survey indicated that their Business Partners spend 30% to 50% of their time on gathering data and using spreadsheets, whereas 19% indicated their Business Partners spend even more than 50% of their time on this [Del14].

- **Sub-optimal data analytics and the lack of a future focus**
  Probably the most mentioned barrier to successful FBP is that of poor systems and the ineffective use thereof [PWC17]. According to Smith, despite being surrounded by disruptive innovation and technology, the Finance function has yet to innovate [Smi15, Wat17]. The consequences of this are suboptimal data analytics and the considerable amount of time spent on data manipulation and other non-value adding activities as mentioned above. These activities are mainly focused on measuring past performance and cause Finance to be primarily occupied with looking back, rather than looking forward [PWC17]. In addition to that, this leads to inconsistent data and information, causing the need for manual adjustments that further constrain the time available for value-adding activities [PWC17]. In a recent survey, performed in 2019, this problem still seems prevalent in many companies: "Even though the importance of insight activities is well recognised, Business Partners are still tied up with manual tasks."[Del19]. In addition to limiting the time available for value adding activities such as data analysis and looking forward, it also influences the quality of the data analysis activities that are performed. The lack of understanding of how to put analytics to use to improve business performance is found to be the number one challenge [Gin11]. According to Deloitte, the rise in the amount and complexity of available data has not resulted in a comparable rise in the amount of insights obtained. EY found the same in their survey amongst CFOs, who voted 'Lack of effective data analytics to provide business insights' to be one of the main barriers to a partnering relationship with their CEO [You15].

Finance leaders should change the way data and technology are used, and recognize the importance of advanced data analytics [Gin11] as the quality of IT systems available affects the Financial's ability to offer insights [Qui14]. In order for organisations to provide valuable insights, having proper data, processes and systems in place is required [Gin11, PWC17]. These systems needs to be improved in such a way that accessing information can be done easily and quickly [icwC11] and in such a way that it facilitates the gathering and recording of data consistently and in the right way [PWC17]. Furthermore, not only do the systems themselves need to be improved, but companies also need to use them more effectively [Del19]. The improvement of systems in turn contributes to a reduction in manual and repetitive activities, which are seen as a distraction from the value-adding activities of FBPs [Del14, icwC11, PWC17]. Together, improved systems and a reduced amount of manual work, enhance the quality of data analysis. Furthermore, improved data analytics also requires improved data quality [ICA18], which entails data being credible and accessible, but also timely and rich in content[Del12, Del13]. Improved data quality is as well in part accomplished through the use of improved systems that capture the data and ensure its consistency. The improved data analytics and the insights it brings is said to form the basis for better decision-making [Del14, PWC17], anticipating digital disruption, measuring performance and responding swiftly [You15].

To conclude, the increasingly competitive and faster-changing business environment and the vast amount of available data have led to a demand for more and better insights and improved decision-making. As Finance is well-positioned to deliver this decision-making support, the business increasingly relies on the Finance Function to fulfil this task. However, Finance remains primarily tied up in non-value adding activities, that are primarily past rather than future-focused and, most importantly, result in suboptimal data analytics. Improved predictive analytical capabilities, such as Financial Forecasting, support this need for a more future focus, reduce the time spent on manual activities and improve the quality of data analytics as well as promote the acquisition of insights. Therefore, we expect the use of Financial Forecasting solutions to improve the derivation of insights and the overall decision-making process for Finance Functions, whether FBP is the goal or not.

2.2 Machine Learning

As discussed in the introduction, the use of ML can bring companies great benefits. However, its successful implementation is frequently impeded by the so-called black-box problem. In this section, we will look into what exactly is ML in order to understand how it works as well as why it presents this black-box problem. Next, we provide an overview of the previous research into the use of ML for applications within the Finance domain in general, as well as for forecasting purposes in particular.

2.2.1 Definition

Machine Learning refers to a set of models or algorithms that learn from past data and are able to enhance their performance over time, i.e. learning [GCR19]. Laurence Moroney described the process of ML by comparing it with traditional programming, see Figure 2.1 [Mor19]. Where traditional programming takes an amount of data and then programs a set of rules to derive answers from that data, ML does the reverse. It takes the data and the corresponding answers as its input and then lets a machine infer rules that correctly map the data to the corresponding answers. Hence, ML is the study of algorithms used by computers to perform certain tasks without the use of explicit instructions. Instead, the performance of these tasks relies on the deduction of rules and the discovery of patterns that compose a mathematical model used to make decisions or predictions [Alp14].

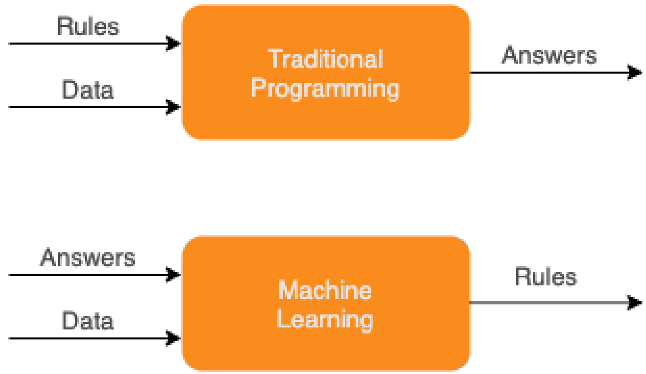


Figure 2.1: The process of Machine Learning [Mor19].

We distinguish between three types of ML, namely supervised learning, unsupervised learning and reinforcement learning [QWD+16]. The above explanation in Figure 2.1 best describes the process of supervised learning as it involves the training of a model using so-called labelled data. With labelled data the desired or expected output (answers) for each of the input instances (data) is available. An example of supervised learning is the development of a model that detects spam e-mails. Such a model is trained using a dataset consisting of historical e-mails in which for each e-mail a label `spam` or `not-spam` is given [SSBD14]. Using these labels, the ML algorithm can then extract patterns by looking at which characteristics are common for e-mails labelled as spam and which are not. Based on these patterns, a number of rules are deducted that form the algorithm that determines whether an e-mail classifies as spam or not.

Furthermore, ML techniques can be further divided based on the purpose of the task at hand [QWD+16]. Supervised learning is used for either classification or regression problems. With classification, the model tries to find the class to which a data instance belongs, e.g. True/False, red/purple/green or small/medium/large. Unlike classification, with regression the model's output is a continuous value, i.e. it is numeric. With regression, the model tries to find a relation between the input variables that best predict the expected output variable. An example of this is the prediction of a person's weight given their height, age and gender. The unsupervised learning approach is most frequently used for anomaly detection and clustering tasks [QWD+16]. The aim is to analyse each data instance and group them based on certain characteristics or features. Ultimately, this results in a division of the data instances in which similar instances are grouped together and dissimilar instances are separated [SSBD14]. When the problem at hand is a decision-making problem, reinforcement learning is the preferred approach. An example of this is a robot waiter that has to learn the best course of action to make the restaurants' guests happy. It does so based on feedback it receives from its environment, the amount of tip in this case.

2.2.2 The black-box problem

In the previous section we provided a definition of ML and discussed the three most common techniques. Based on this definition, we can now explain why ML poses the aforementioned black-box problem. The black-box problem is directly related to the level of transparency of a model. In section 2.4.3 we will go deeper into the definition of transparency and the characteristics a transparent model needs to possess. Here we will only illustrate the black-box problem by means of two examples.

First of all, with ML it is the machine that infers the rules that make up the model, as illustrated in Figure 2.1. Therefore, unlike with traditional programming, we do not know beforehand how the model makes a certain prediction or decision. In order to gain an understanding of how the model operates, we need to take a closer look at its inner workings. In a number of cases, this suffices to understand what the model does, and hence makes it a transparent model. This is the case for linear models, for example. For other types of models, such as neural networks, it remains nearly impossible for a human to follow the reasoning of the model as a whole even if we have information on its inner workings.

1. **Linear Models**
   Linear models are obtained using linear regression. The assumption behind linear regression is that our dependent variable y has a linear relation with our independent variables [HA19]. The simplest form of linear regression contains only one independent variable x and is called a simple linear model. A simple linear model looks as follows:
   $$y = \beta_0 + \beta_1 x + \varepsilon \tag{2.1}$$
   where x is our input, β_1 is the weight given to this input x, β_0 is the intercept and ε is the error. For example, if we want to predict the daily number of ice creams sold based on the temperature, the linear model could look something like: nr_ice_creams $= 8 \times$ temp $+ 50$. This model is considered transparent as the reasoning behind the model is relatively easy for humans to follow: *"on average we sell 50 ice creams a day, and for every degree increase in temperature, we sell 8 additional ice creams"*. However, if this model contained 20 inputs rather than 1, it is called a multivariable linear model, and we would obtain a function of the following form:
   $$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + .. + \beta_{20} x_{20} + \varepsilon \tag{2.2}$$
   The impact each of the individual inputs $x_1, x_2, ..., x_{20}$ has on the outcome of our model y is still interpretable. Namely, x_1 has a weight of β_1, x_2 a weight of β_2, etcetera. However, the reasoning of the model as a whole already becomes more difficult for humans to follow due to the large number of inputs involved.

2. **Neural Networks**
   An example of models for which it becomes nearly impossible to follow their reasoning as a whole is Neural Networks (NN) [AB18]. A NN consists of multiple layers that connect the inputs to the outputs. Consider, for example, a deep learning algorithm that has to detect the hand-written number that is depicted in an image. The input of this NN is the array of pixel values contained in the image. If this image is 28 by 28 pixels, then the first layer of our NN contains 784 inputs. These inputs are called the neurons and each neuron contains a number between 0 and 1, indicating the brightness of the pixel in that neuron. Neurons close to 0 indicate pixels that are not activated and neurons close to 1 indicate pixels that are activated.

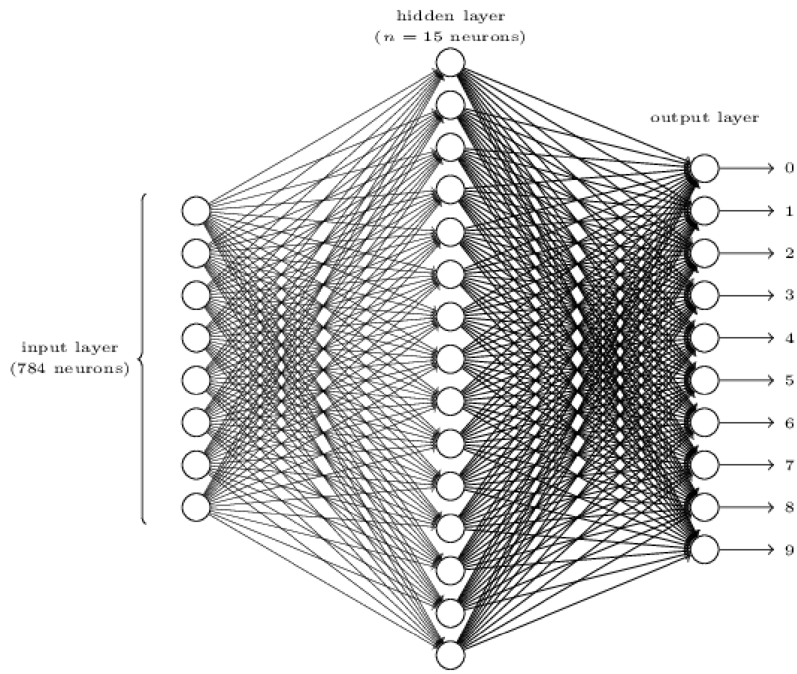


Figure 2.2: Neural network consisting of an input, output and a single hidden layer [Nie15].

After the input layer follow several so-called hidden layers and eventually the output layer. All these layers again consist of a number of neurons. The output layer in this example contains 10 neurons, one for each of the possible digits our image can contain, namely 0 to 9. If the neuron in the output layer that represents the digit 3 has a value close to 1 and the other neurons all have values close to 0, then the algorithm predicts that the hand-written number in our image is a 3. To understand how the values in the input layer are translated to the values in the output layer, we need to look at what happens in the hidden layers. The values of the neurons in the hidden layers are computed using (1) the values of the neurons in the previous layers and (2) the weights given to the connections between the current neuron and the neurons in the previous layers. Figure 2.3 shows what the computation of a single neuron consists of.

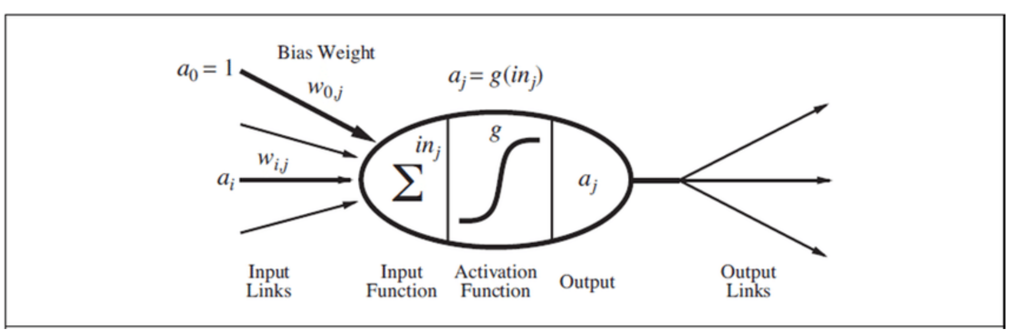


Figure 2.3: The components of a neuron [SGY+15].

The input links $a_0, a_1, .., a_i$ are the values from the neurons in the previous layer. All input links are assigned a weight $w_0, w_1, .., w_i$ that determines the strength of that input. The inputs are multiplied by their weights and the resulting products are added to obtain the input function for a neuron j: $in_j = a_0 \times w_0 + a_1 \times w_1 + ... + a_i \times w_i$. Next, we obtain the actual output a_j by applying the activation function, which transforms the number from the input function to a value between zero and one. This is done because, as explained above, we want our neurons to contain values between 0 and 1 to indicate the level of pixel activation. This process is repeated for all the other neurons in the current layer.

It becomes evident that it is nearly impossible for humans to follow the reasoning of this NN as a whole. Not only does it contain many inputs, namely 784, but these also have to pass through multiple layers before we obtain the outputs. Furthermore, for every neuron in every layer, we not only apply an input but also an activation function. And although all neurons in a certain layer use the same inputs (the neurons from the previous layer), each combination of neurons is assigned its own weight. In the example described above, this results in 784 (#neurons in the input layer) $\times 15$ (#neurons in the hidden layer) $\times 10$ (#neurons in the output layer) = 117.600 weights. The number of weights alone makes it impossible for a human to perform this computation in his head.

From the explanations of the models above, it can be observed that for the first class of models the information on their inner workings suffices to follow the reasoning of the model. However, when the number of inputs of a linear model increases, we see that interpreting the model already becomes more difficult. For the second class of models, neural networks, even when information on the inner workings is available, following the reasoning of the model as a whole becomes nearly impossible due to the great number of weights and functions involved, illustrating the cause of the black-box issue.

2.2.3 Financial Applications

In this section, we provide a review of previous research into the use of ML for financial applications in general and for forecasting purposes in particular, and discuss their main contributions. We found that that the majority of these works focus on the FSI, rather than the Finance Function. However, the research performed within the FSI does illustrates the benefits ML can bring when applied to financial data.

One area in which the use of ML has been investigated is that of portfolio management. Examples include the allocation and optimization of R&D budgets [Jan19], and the selection of stock portfolios [JMK96]. Another frequently studied ML application within the banking sector in particular concerns that of credit risk assessment [CRC16, KSAZ13]. Interestingly, even for this research field where the number of studies is overwhelming, Chen et al. found that the black-box problem remains an issue due to the lack of user involvement [CRC16]. ML is also commonly used for the analysis of financial data using anomaly detection. Examples include the detection of credit card fraud, fraudulent insurance claims, insider trading or money laundering [AMI16, HMYC18]. Lastly, ML is commonly used for forecasting purposes within Finance. One example is that of forecasting bankruptcies of companies. Research performed by Lahmiri and Bekiros found that ML models, Generalized Regression Neural Networks (GRNN) specifically, can outperform traditional statistical methods in forecasting bankruptcy [LB19]. They explain that this superior performance is caused by the fact that the investigated ML models do not suffer from the restrictions that traditional statistical models pose on both the input and the output. These restrictions include requiring interdependence amongst the input variables or limiting the possible output models to be of linear form only. Another area where ML models are used for forecasting purposes is the stock market. Perhaps the most researched topic regarding stock markets is the use of Neural Networks for the prediction of stock indices and share prices [DP10, VSO+13, CD17]. Finally, a Financial Forecasting application that is particularly useful in support of the Planning, Budgeting & Forecasting process within Finance departments is the forecasting of financial items, e.g. revenue and costs. Gajewar and Bansal investigated a number of forecasting models for their performance in predicting a companies' revenue [GB16]. Unfortunately, due to the confidentiality of the data that was used, they do not report the actual observed errors, but only the MAPE of the models relative to each other. Nevertheless, they state that their ML-based forecasting models forecast revenue at a quarterly interval with reasonable accuracy and that in some cases the observed error was only 0.1%. Two years later they continued their research by adding additional data to the historical seasonal and trend patterns used in the initial research, as well as automate their approach through an end-to-end pipeline [BGG+18]. This improved, automated solution reached an accuracy of 98-99% on average and is adopted throughout Microsoft's Finance organization and fulfils an important role in their forecasting processes, "from providing Wall Street guidance to managing global sales performance" [BGG+18]. Although written and developed by Microsoft employees, it does provide interesting insights into the benefits of this type of Financial Forecasting for Finance departments.

2.3 Financial Forecasting

As discussed in the introduction, the validation of XAI techniques not only depends on the target audience that will use them but also on the ML models for which the XAI has to provide an explanation. Furthermore, the type of ML models used, determine to a great extend which XAI techniques are potentially suitable. In order to get a clear picture of the techniques and models used for Financial Forecasting, we will firstly look into what Financial Forecasting entails, followed by an overview of well-known models for Financial Forecasting. Next, we will discuss a number of existing software solutions for Financial Forecasting and elaborate on the tool used in this research, PrecisionView™.

2.3.1 Definition

Forecasting refers to the process of producing future output values for a certain item of interest, based on a given set of inputs. The assumption behind forecasting is that the value we are trying to forecast, to some extent, depends on past events or events that we may currently observe [JT01]. Amongst these events, we look for patterns that we expect to continue to see in the future and utilize them to extract rules that map the inputs to the output. Providing an exact definition for Financial Forecasting is somewhat difficult as it depends on the context of its usage [Sam15]. In the broadest sense of the word, Financial Forecasting refers to the process of predicting the future value of a financial item. However, what these financial items might be, depends on the financial domain in which Financial Forecasting is used. As discussed in section 2.1.1, Finance can refer to either the

Finance Function or the Financial Services Industry. The majority of the research on Financial Forecasting is conducted in the context of the latter, the FSI. Financial Forecasting in this context most commonly refers to the forecasting of stock market values [LLC09, CT01, CLL19, JT01]. However, in this research, we are interested in Financial Forecasting in the context of the Finance Function. In his book on Financial Forecasting and Modeling, Samonas defines Financial Modeling in the context of the Finance Function as follows: ".. the preparation of detailed company-specific models used for decision-making purposes and financial analysis." [Sam15]. He identifies Financial Forecasting as a specific type of Financial Modelling and describes its purpose as the prediction of items in the financial statements for future years. The emphasis, when predicting financial line items, is on the prediction of sales. This is because nearly all of the other financial items are driven by sales [Sam15]. The majority of the research on Financial Forecasting within the Finance Function indeed focuses on the prediction of revenue [GB16, BGG+18, PE14]. However, the Financial Forecasting solution upon which our XAI prototype is to be implemented, consists of forecasts for several financial line items. Therefore, in this remainder of this research the term Financial Forecasting refers to the prediction of any financial line item.

Now that we have defined Financial Forecasting, we can take a closer look at how it works in practice. As discussed in section 2.2.1 there are different ML techniques to generate a model, depending on the purpose of that model. In the case of forecasting financial line items, the purpose of the model is to return the predicted numeric value for a specific line item. Therefore, with Financial Forecasting the ML task at hand is a regression task. Theoretically speaking, it is possible to use other ML techniques for Financial Forecasting, but this would be less informative. If, for example, we would use classification, we would only be able to predict in terms of categories, e.g. "low revenue", or "high revenue". This is ambiguous, difficult to interpret and not very informative: *How high is "higher"? Higher than the current revenue? With what percentage? 105%? 150%?* With regression, we train a model using both inputs, i.e. the variables that potentially influence our output, as well as their corresponding outputs. Suppose, for example, that we want to predict a company's revenue and we expect the revenue to be influenced by the marketing spend. In that case, we could train a model by feeding it both historical data on the marketing spend and the corresponding historic revenue, in order to search for a relation between marketing spend and revenue. This means that Financial Forecasting models are trained using labelled data and hence fall under the supervised learning category, as explained in section 2.2. Exactly what this training process works depends on the model chosen. We will elaborate on this in the following section.

2.3.2 Models

In this section, we will explain the process of Financial Forecasting in more detail by looking at a number of commonly used models for Financial Forecasting. As we have shown, the task at hand is regression. Furthermore, Financial Forecasting models frequently use time-series data. These two characteristics, regression and time-series data, determine the class of ML models that can be used for a specific Financial Forecasting tasks. Furthermore, which of these models is best suited for a specific forecasting goal also depends on the number and type of inputs. The possible forecasting models can be divided into models that use a single input and models that use multiple inputs. Furthermore, we make a distinction between models that are suitable for time-series data and non time-series data. In this section, we will discuss the most common ones.

Non Time-series Data
A well-known approach for forecasting using non time-series data is linear regression. We distinguish between two types of linear regression, based on the number of inputs of the model, namely simple linear regression and multiple linear regression. Linear regression is one of the most straightforward forms of regression and is deemed relatively easy to interpret due to its linearity [Mol20, Sam15]. It models a linear relationship between the input(s), also referred to as the independent variable(s), and the output, also referred to as the dependent variable.

- **Simple-linear regression**
  As explained in section 2.2.2, a simple-linear model has the form: $y = \beta_0 + \beta_1 x + \varepsilon$. The training process for a simple-linear model is similar to that of the multi-linear model explained below.

- **Multi-linear regression**
  As shown previously, a multi-linear is given by a function that computes the dependent variable y as the weighted sum of the independent variables $x_1, x_2, ..., x_n$:

  $$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + .. + \beta_n x_n + \varepsilon \tag{2.3}$$

  Through the learning process, the model tries to find the optimal weights $\beta_0, ..., \beta_n$ that best predict y. This entails findings weights that minimize the difference between what the model predicts and the actual outcome. Note that β_0, referred to as the intercept, is not multiplied with any variable. The intercept is the point where the line given by the function crosses the y-axis. It tells us what our model predicts when all independent variables are set to zero. This interpretation of the intercept, however, is often not very relevant as the case in which all independent variables are set to zero is generally quite unrealistic. The last term ε is called the error term and accounts for the error in the prediction, i.e. the difference between what is predicted and what should have been the actual outcome. Its purpose is to explain anything that affects y that cannot be explained by the inputs x_i [HA19].

Time-series Data
As mentioned above, models for Financial Forecasting frequently use time-series data. In fact, it is often referred to as Financial Time-series Forecasting. Time-series data is data that contains timestamps indicating the date and/or time at which a certain instance was recorded, e.g. the revenue on May 22, 2020. Timestamps help use discover patterns relating to dates and provide us with additional information that helps train a prediction model. More specifically, a time-series can contain 4 components: the trend, the seasonality, cycles and outliers [HA19]. A trend indicates an upward or downward pattern in the data. Seasonality refers to repeating patterns of highs and lows that are caused by seasonal factors such as the month of the year or the day of the week. Cycles are similar to seasonal patterns but differ in that their fluctuations are not of a fixed period and are caused by circumstances, e.g. a recession. Lastly, outliers are data points that lie relatively far away from the rest of the data and are the consequence of unforeseen or unpredictable events. There exist a number of forecasting models specifically for such time-series data. One main

difference between these time-series models and linear models explained above is that time-series models take previous observations of the value to be forecasted as an input to their forecasting model, as we will see below. Furthermore, within the set of models suitable for time-series data we can again distinct between the numbers of inputs a model takes. We refer to models with a single input as non-dynamic models and models with multiple inputs are called dynamic models [HA19].

Non-dynamic models

Well-known examples of time-series models that only take one input are moving average, exponential smoothing and ARIMA [HA19]. As an input, they take the historical values of the item they are trying to predict, i.e. the output. For example, if we are trying to predict future revenue, then the output is the future revenue and the input is the historical value of the revenue.

- **Moving Average**

  The moving average (MA) is a method for time-series decomposition that forms the basis for many other time-series models. The idea behind the MA is that it averages the m most recent data values. This helps to smooth out any randomness and outliers in order to detect trend and cycle patterns [Sam15]. A moving average of the m most recent values, also called an m-MA, is given by:

$$\hat{T}_t = \frac{1}{m} \sum_{j=-k}^{k} y_{t+j} \tag{2.4}$$

  where $m = 2k + 1$ and k indicates the period window of our MA. Hence, for a moving average or order 5, i.e. 5-MA, $k = 2$, indicating that we look at the two periods before and the two periods after our period of interest. If we want to know the 5-MA at period t we then compute the sum of the values at times t_{-2}, t_{-1}, t_0, t_1 and t_2 and divide it by 5 to obtain their average. Figure 2.4 gives an example of the result of applying a moving average or order 5 to a dataset of electricity sales. In the figure, it can be seen that the moving average of the data (red line) the trend of the original data (black line).

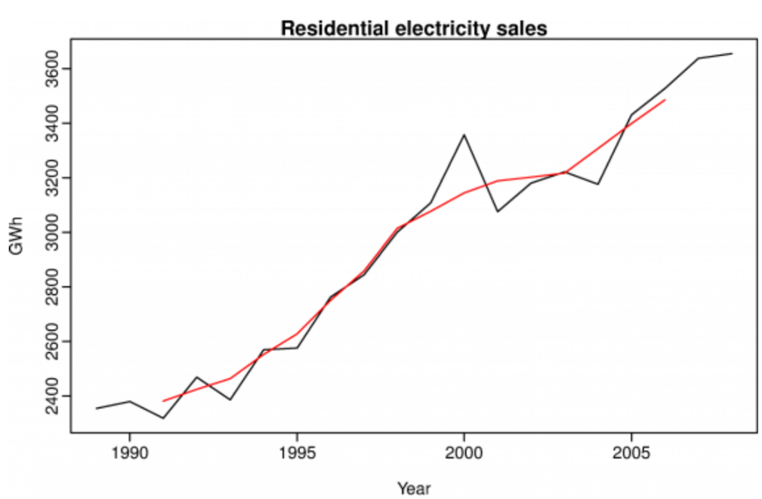


Figure 2.4: The volume of electricity sold to residential customers in South Australia(black) and the MA estimate of its trend-cycle(red) [HA19].

- **Exponential Smoothing**
  Exponential smoothing is similar to the moving average method in that it takes the average of past observations. The difference, however, is that not all past observations weigh equally in the average. The past observations are assigned weights that increase exponentially as the observations become more recent. This means that more recent observations count more heavier towards the average. A simple exponential smoothing model looks as follows:

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1-\alpha)y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + ... + \alpha(1-\alpha)^{T-1} y_1, \tag{2.5}$$

  where α is the smoothing factor, i.e. the weights, and has a value between 0 and 1, y_T is the last observed value, $\hat{y}_{T+1}$ is the first future value and $y_T, y_{T-1}, y_{T-2}, ..., y_1$ are all past observations. From equation (2.5), we can see that the weights assigned to each past observation decrease as $\alpha < \alpha(1-\alpha) < \alpha(1-\alpha)^2$.

- **ARIMA**
  Autoregressive Integrated Moving Average (ARIMA) applies a combination of autoregressive and moving average models on an integrated version of the time-series in question. The reason for taking the integrated version of a time-series is to deal with non-stationarity [Abr19]. In a stationary time-series, the properties of the series are independent of the time at which it is observed. Hence, the mean and variance of such a series should not change over time. This means that time-series that contain trend or seasonality are by definition non-stationary. ARIMA deals with this by taking the integrated version of the time-series, which is obtained by differencing the time-series in order to make it stationary. Differencing entails taking the differences between consecutive time stamps [HA19]. Next, the autoregression model finds a relationship between the output variable, i.e. what we want to predict, and the previous values of that variable. For example, when predicting future revenue, autoregression models use historic revenue as the inputs to forecast this future revenue. The term autoregression indicates that the function is a regression of the variable against itself. We define an autoregressive model as $AR(p)$, where p denotes the number of historical values used. $AR(p)$ can be written as:

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + .. + \phi_p y_{t-p} + \varepsilon_t \tag{2.6}$$

  Here, y_t indicates the value we want to predict at time stamp t and $y_{t-1}, ..., y_{t-p}$ are the p historical values of y, also referred to as the lagged values. Furthermore, similar to multi-linear regression, $\phi_1, ..., \phi_p$ are the weights assigned to these inputs, ε_t gives the error term of the model and c is a constant representing the intercept. Then the moving average model is factored in. This is as well a regression-like model, but instead uses the past values of the forecasts' errors. It is defined as $MA(q)$, where q indicates the number of historical errors to be used. $MA(q)$ can be written as:

$$y_t = c + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + ... + \theta_q \varepsilon_{t-q} \tag{2.7}$$

  Here, $\varepsilon_{t-1}, ..., \varepsilon_{t-q}$ represent the q lagged forecast errors and $\theta_1, ..., \theta_q$ their corresponding weights. ε_t is the error of the current prediction and hence can never actually be observed until we know y_t. Finally, we can combine the $AR(p)$ and $MA(q)$ model and apply it to an integrated time-series to obtain the ARIMA(p,d,q) model. Here, d indicates the order of integration, i.e. the number of times we have differences the time-series to make it stationary. ARIMA(p,d,q) can then be written as follows:

$$y'_t = c + \phi_1 y'_{t-1} + .. + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + ... + \theta_q \varepsilon_{t-q} + \varepsilon_t \tag{2.8}$$

The occurrences of y have been replaced with y' to indicate the differenced series. The prediction y'_t is now given as a function of the lagged values of y' and its error terms.

- **Prophet**
  Prophet was developed by Facebook [TL18]. The idea behind Prophet is that it extracts certain components of the time-series, namely seasonality, growth and holidays. Growth indicates the increasing or decreasing change over time, i.e. the trend component as discussed above. The idea behind the holiday component is that it captures the effect of holidays and other events. Hence, the holiday component is comparable to the cycle component discussed above. These components are then combined in a so-called Generalized Additive Model (GAM) to obtain the forecast function [Abr19]:

$$y_t = g_t + s_t + h_t + \varepsilon_t, \tag{2.9}$$

  where g_t indicates the linear or logistic growth curve for modelling non-periodic changes in time-series. s_t indicates the seasonality, i.e. periodic changes (e.g. weekly/yearly seasonality). Lastly, h_t gives the effects of holidays and special events (user-provided), and ε_t is the error term that accounts for any unusual changes not accommodated by the model.

Dynamic models
The time-series models discussed above take only one input, namely the previously observed values of the item we are trying to predict. However, in many cases, we might want to include additional information in our model. These additional input variables can be anything from external factors such as weather or economical data to internal factors like employee productivity, the price of a product, marketing spending, etcetera. We will discuss two dynamic models that are extensions of a non-dynamic model discussed above.

- **ARIMAX**
  There is an extension to the ARIMA model that allows for the inclusion of additional input variables. This extension is called ARIMAX, where the X stands for exogenous variables. To obtain an ARIMAX model, we extend the ARIMA model in equation (2.8) with the integrated value of the additional variable x at time t, namely x'_t and multiply it by its weight β.

$$y'_t = \beta x'_t + c + \phi_1 y'_{t-1} + .. + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + ... + \theta_q \varepsilon_{t-q} + \varepsilon_t \tag{2.10}$$

- **Prophet with Regressors**
  Similar to ARIMA, there is an extension to Prophet that allows for the inclusion of

other variables in addition to the three main components in 2.9 above. This will result in the following function:

$$y_t = g_t + s_t + h_t + x_t\beta + \varepsilon_t, \tag{2.11}$$

where, x_t represents the value of an additional input x at time t, and β the weight assigned to that input.

There are many other models that can be used for the analysis of financial time-series data. Support Vector Regression (SVRs), for example, are a much-used technique used for Financial Forecasting [BGG$^+$18, LLC09, LB19]. The Neural Networks that we already briefly discussed in section 2.2.2 are also frequently used for Financial Forecasting [JT01]. We will not discuss them in detail because they are beyond the scope of this research. The models explained above contain the models used by PrecisionView™, as we will see in section 2.3.4, and hence form the basis for our investigation into suitable XAI techniques for Financial Forecasting.

2.3.3 Existing Software Solutions

As mentioned in section 1.3, the prototype developed in this research is built upon an existing Financial Forecasting solution, called PrecisionView™. However, as we want our prototype to be easily extendable to other ML-driven Financial Forecasting solutions, we aim to develop a prototype that is as generic as possible. To this end, it is important to have an overview of what other software solutions for Financial Forecasting look like. More specifically, it is important to know what models are commonly used by these existing software solutions. Furthermore, we are interested in the current state of explainability of existing Financial Forecasting software. Therefore, in this section, we will discuss a number of existing software solutions for Financial Forecasting, discuss what models they use and whether they use XAI. Existing software solutions do not solely focus on Financial Forecasting, but are aimed at the overall Financial Planning & Analysis (FP&A) process. Today, there exist many software solutions for FP&A. Here, we have selected a few of them to discuss in further detail. The chosen solutions were selected based on Gartner's magic quadrant for cloud FP&A solutions [GL20]. In this report, they assessed a large number of FP&A solutions along two dimensions, namely the ability to execute and the completeness of vision, as shown in Figure 2.5.

Figure 2.5: Gartner's Magic Quadrant for Cloud Financial Planning & Analysis Solutions, (August 2020). [GL20]

The highest scoring FP&A solutions are found in the upper right corner of the quadrant and are called the leaders. In our review, we decided not to select the leaders, but the solutions that scored the highest on completeness of vision. One of the criteria for completeness of vision is innovation. Furthermore, the visionaries are said to "often introduce new technology, services and business models.." [GL20]. These are important criteria for reviewing existing solutions. This is because the use of XAI for Financial Forecasting has largely been left uninvestigated and hence we expect to have the best chance at finding XAI technologies amongst these more innovative solutions.

1. **Oracle Planning**
   Oracles' Planning and Budgeting Cloud Service is part of the Enterprise Performance Management (EPM) Cloud solution [Ora21]. It is used to model, plan and report and provides capabilities for planning, forecasting and scenario modelling purposes. Oracle Planning consists of several modules, such as the 'Dashboard' module for visualising and analysing data and the 'Projects' module for the assessment of the impacts that certain initiatives have on different corporate resources. Here, we are particularly interested in the 'Financials' module, which allows for the creation of driver-based plans for a companies' revenue, expenses, income statement, balance sheet and cash flow. It offers forecasting capabilities to forecast the future values of the financial items in these plans.

   - **Model usage:** There are several possible models to obtain these forecasts. The final forecast model is automatically selected after checking the accuracy for a number of potential models. First, the accuracy for all nonseasonal forecasting models is checked. If it appears that the data is seasonal, then the accuracy for all seasonal forecasting models is checked as well. The forecasting method with the highest accuracy, and hence lowest error rate, is picked. The nonseasonal methods include Simple Moving Average (SMA), Double Moving Average (DMA), Single Exponential Smoothing (SES), Double Exponential Smoothing (DES), Nonseasonal Damped Trend Smoothing (DTS) and ARIMA. The seasonal methods include seasonal additive, seasonal multiplicative, Holt-Winters' Additive, Holt-Winters' Multiplicative, seasonal Damped Trend Additive Smoothing, seasonal Damped Trend Multiplicative Smoothing and SARIMA [Ora21]. There is also an option for users to define their own models using formulas. When it comes to running the forecasts, there are two approaches. The first approach is to manually run a prediction on certain financial data. The second approach uses the so-called Auto Predict functionality and entails automatically running forecasts at specified intervals. The Auto Predict function automatically populates the forecast models with new data upon entering a new planning cycle.

- **XAI usage:** Furthermore, Oracle Planning currently does not incorporate XAI techniques to explain or visualise the inner workings of the selected forecast model. However, it does provide additional information on forecasts that provide users with some insights on the forecasts and their accuracy. This information includes the accuracy percentage, the error rate in terms of the Root Mean Square Error (RMSE), the best performing forecasting model, the values of the weights assigned to the inputs, the number of missing inputs for which the value was adjusted, the number of adjusted outliers and the seasonality. Furthermore, it provides statistics on the historic values of the variable we are trying to predict. Namely the number of values, the minimum, maximum and mean value, as well as the standard deviation (std).

2. **SAP Analytics Cloud**
SAP Analytics Cloud combines planning and business intelligence capabilities to enable users to analyze their business processes and create plans for them within one application. It offers some of the same functionalities as Oracle Planning. Similar to Oracle Planning, it contains options for financial modelling, automated reporting and the creation of plans. In addition to that, it offers predictive analytics and machine learning capabilities. This includes the generation of manual and automated forecasts and the detection of patterns and important drivers in a dataset. Furthermore, ML is leveraged to offer Natural Language Processing (NLP) functionality. Users are able to get information from the system by asking questions in a conversational manner [SAP21].

- **Model usage:** The functionality for automated forecasting is called the "Smart Predict" function. Smart Predict lets users choose between a classification, regression or time-series prediction task. When it's a classification or regression forecast, users can add additional information on the input values, also called influencers in SAP. Namely, they can indicate which inputs are expected to have an influence on the predicted value, which inputs should be excluded as influencers, and set a maximum for the number of influencers. From their documentation, it is not quite clear what models are used to obtain the forecasts. For time-series forecasts, they use several models to find the individual components that make up a forecast [Bru]. Namely, the trend, cyclic and fluctuation components. The cyclic component can consist of both seasonality and periodic movements. 8 models from 2 different methods are used to detect the best trend component. The first set of methods is stochastic and includes Lag_1, Lag_2 and Double Differencing. With the lagged values Lag_1 and Lag_2 the forecasted value is equal to respectively the last and second to last observed value. Double differencing refers to the process of taking the difference between consecutive time stamps, as explained in section 2.3.2. The second set of models is deterministic and involves applying regression on 5 different input combinations consisting of the date and the potential influencer variables. After these 8 trend functions are obtained, the time-series is detrended using each of these 8 functions. Based on these detrended time-series, seasonality and periodic movements are detected. Next, the fluctuation is detected by applying an autoregressive model on the 8 series for which the trend and cycles have been removed. Finally, the resulting 8 models are tested for their accuracy and the model with the highest accuracy is selected, see Figure 2.6. The methods used by Smart Predict are comparable to or extensions of the methods discussed in section 2.3.2.

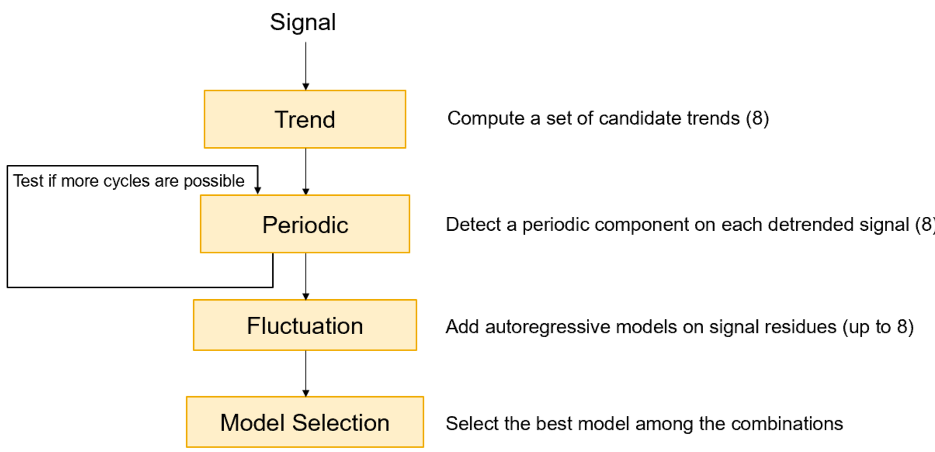


Figure 2.6: time-series models training process in SAP Analytics Cloud. [Bru]

- **XAI usage:** Again, XAI techniques are not available for the forecasts in SAP Analytics. They do present a number of statistics to provide some insight into the forecasts. Namely, the accuracy expressed as the Mean Absolute Percentage Error (MAPE), a plot showing the actual, forecast, outlier and anomaly values. If it is a time-series model, there is also a plot showing the trend, cycles or seasonality and fluctuations. Furthermore, the "Signal Statistics" provide different statistics regarding the target value, i.e. what is being predicted. Namely, the average, minimum, maximum valued and the standard deviation (std). Lastly, there is a so-called "Lagged Predictors Contribution" graph that displays which past values have a relatively high or low influence on the prediction. If the model type is classification, then there is also information on the contribution of each of the input variables to the forecast, an overview of the influence of each category within a single variable, e.g. age categories of the input variable age, and a performance curve that allows for the comparison of the model in question with other models. Interestingly, on their blog, SAP has announced several customer engagement projects where customers can get involved in the development of new techniques. Two of these projects revolve around the use of XAI and hence indicate that SAP might potentially incorporate XAI techniques in the future [May20].

3. **Anaplan**
   Anaplan is a cloud platform offering modelling and planning functionalities. It focuses on the Finance Function, but also the other business functions such as sales, supply chain, human resources and marketing. According to Gartner, Anaplan stands out from the other financial planning solutions by its support for modelling complex financial models [GL20]. The planning capabilities specifically for the Finance Function include long-range, revenue, OpEx and CapEx planning, balance sheet & cash flow, and income statement forecasting [Ana20a]. Forecasts in Anaplan are updated in real-time and are said to make use of AI and ML. Anaplan itself offers around 30 forecasting algorithms within its platform. Furthermore, it allows for integration with a user's custom models, as well as models of other ML platforms, such as Amazon Forecast [Ana20b].

   - **Model usage:** Anaplan distinguishes between four categories of models depending on the purpose of the model [ana20c]. The first category is referred to as curve-fitting models and is aimed at detecting trends in historical data that assist in forecasting future trends. The models used in this category are linear, logarithmic, exponential and power regression and do not take into account seasonality. The second category is called smoothing models and includes extensions of the models discussed in section 2.3.2, such as Moving Average (MA), Double Moving Average (DMA), Single, Double and Triple Exponential Smoothing (ES), and Holt's Linear Trend. The third category of models is also aimed at smoothing but allows for data to be seasonal. It assists in breaking down a time-series into the baseline, trend and seasonality components. The models in this category include Additive Decomposition, Multiplicative Decomposition and Winters' method. The last category, intermittent models, includes a set of models intended for specific forecasting circumstances or to compare other methods.
   - **XAI usage:** Anaplan as well currently does not incorporate XAI techniques in its platform. Nevertheless, they do recognize its importance, stating that: "With the number and frequency of forecasts constantly increasing, forecasting teams need to be able to better analyze forecasts, track accuracy and continuously improve forecasting models." [Ana20b] To this end, it provides a graph with the actual and forecasted values indicating the accuracy. Furthermore, it presents a table with multiple error measurements, namely MAPE, RMSE and Mean Absolute Deviation (MAD) for each of the tested forecast models.

Solution	**Models**	**XAI**	**Other explanation statistics**
Oracle Planning	SMA, DMA, SES, DES, DTS, ARIMA, seasonal additive, seasonal multiplicative, Holt-Winters' Additive, Holt-Winters' Multiplicative, seasonal additive DTS, seasonal multiplicative DTS, SARIMA.	No	Best performing model, accuracy %, RMSE, #adjusted_missing_inputs, #adjusted_outliers, seasonality and input weights. Target variables' min, max, mean and std.
SAP Analytics Cloud	Lag_1, Lag_2, Double Differencing, regression, autoregression.	Currently investigating [May20].	MAPE, actuals vs predicted vs outliers vs anomalies graph. Trend, cycles or seasonality and fluctuations plot. Lagged Predictors Contribution. Target variables' min, max, mean and std.
Anaplan	linear, logarithmic, exponential and power regression. MA, DMA, SEM, DES, TES and Holt's linear trend. Additive- and Multiplicative Decomposition and Winters' method. Intermittent models.	No	Actuals vs. forecasts graph, MAPE, RMSE, MAD.

Table 2.1: Overview existing FP&A solutions and their models and XAI usage.

To summarize, all FP&A solutions reviewed here use models that are similar to or an extension of the well-known models discussed in section 2.3.2, see Table 2.1. Only Facebook's Prophet and the neural network models are not represented in the reviewed Financial Forecasting solutions. Furthermore, all solutions emphasize the importance of analysing and understanding the forecasts they provide. To that end, they do offer a number of statistics aimed at assisting the user in gaining some insight into the forecasts and their accuracy. These include accuracy measurements such as MAPE, RMSE and MAD, plotting the actuals and forecasted values, the components of a time-series and statistics on the target variable. In addition to that, SAP provides an overview of the contribution of each of the inputs to the output, which is similar to a frequently used XAI technique as we will see in section 2.4.5. However, this overview is only available for the classification method. Furthermore, in February 2020 they announced two projects regarding the exploration of XAI techniques for the future. However, until today none of the reviewed solutions actually incorporates XAI techniques.

2.3.4 PrecisionView™

As mentioned in section 1.4, the aim of this research is to bridge the gap between the ML research community and the Finance department by investigating the use of XAI for Financial Forecasting solutions. In order to do so, we will extend an existing Financial Forecasting solution with several XAI techniques and validate their effectiveness through an experiment. The Financial Forecasting solution that was chosen as the starting point for our XAI prototype is called PrecisionView™. In this section we will discuss what the software stack for PrecisionView™ looks like, the information and visualisations it provides, which models it uses and what options it currently offers to provide explainability and transparency for their forecasts.

The Software Stack

PrecisionView™ was developed by Deloitte U.S. and provides financial modelling and forecasting capabilities to forecast, amongst others, items on the Profit & Loss statement and the balance sheet, working capital, or cash flow. It is referred to as a framework rather than a software program. This is because it is agnostic, meaning that it can sit on top of different financial software systems. The software stack for PrecisionView™ consists of four components. Namely, a data platform, the analytics part, a dashboarding component and optionally an Enterprise Performance Management (EPM) system [Del18].

1. **Data platform**
   The data platform delivers the historical data needed to create the forecasts. Examples of data platforms that can be used in PrecisionView™'s stack are Microsoft Azure, Amazon, Google Cloud, SAP HANA, Oracle Analytics Cloud and Workday.

2. **Analytics**
   The analytics part is where the actual analysis of the data from the previous component takes place. In general, the analysis consists of the following steps:

   (a) Driver identification and selection to determine which data inputs have a relation with the target variable that we want to predict.

   (b) Driver analysis to train the actual forecast models using different forecast methods and the drivers identified in the previous step.

   (c) Forecast comparison to compare the accuracy rates of the different forecasting models obtained in the previous step and determine which model has the best performance.

   As discussed in section 2.3.3, some of the data platforms listed above, such as Oracle Planning and SAP Analytics Cloud, also offer part of these data analysis functionalities. If the chosen data platform however does not offer these functionalities or is insufficient for the forecasting goals at hand, either R or Python is used for the analysis process.

3. **Dashboarding**
   Dashboards are used to visualize the forecasts obtained in the analysis step, as well as provide statistics on the accuracy of the forecasts. Furthermore, they are used for scenario analysis, in which the user can model scenarios by changing the values of input variables and see the impact of those changes on the forecasts. Lastly, the dashboard can assist in reporting by allowing users to create snapshots of dashboard visualisations, such as certain scenario analysis. There are again several options for the implementation of the dashboard components. Many EPM systems already have this dashboarding functionality integrated into their systems, such as Oracle's Planning solution, SAP Analytics cloud and Anaplan. If a company does not

use an EPM system, or its EPM does not offer dashboarding capabilities or is insufficient, PrecisionView™ also offers support for standalone dashboarding solutions, such as Tableau, Qlik and SAP Lumira.

4. **Enterprise Performance Management**
   Many companies make use of a so-called Enterprise Performance Management (EPM) system. The purpose of EPM is to monitor, manage and ultimately improve a company's performance on an enterprise-wide level [Glo]. EPM systems support this process by integrating data from different departments within a company and providing analyses capabilities for this data. As we saw in section 2.3.3, many Financial Forecasting solutions are part of an EPM system, such as Oracle Planning. For the purpose of performance management, it is desired for companies to have their financial forecasts directly available within their EPM systems. Therefore, the PrecisionView™ framework offers the possibility to integrate their forecasts in a company's EPM. This includes making the forecasts produced by PrecisionView™ available in a company's EPM. Examples of EPMs for which it offers forecast integration include Anaplan, SAP, Adaptive Insights, which is now part of Workday, and Oracle.

Before elaborating on the information provision and visualisations, the use of models, and the presence of XAI in PrecisionView™, it should be noted that Deloitte is currently in the process of developing an updated version of PrecisionView™. In this research, we worked with the current version that does not yet contain these updated functionalities. Nevertheless, it is important to know what new and updated functionalities are in the pipeline to take this into consideration for the development of our XAI prototype. At this moment, there is no official documentation on this new version of PrecisionView™. However, through a series of interviews with the Business Finance Advanced Analytics Lead at Deloitte U.S. and project leader for PrecisionView™, we were able to gather information on the changes made in this updated version. Below, we will firstly discuss the information and visuals PrecisionView™ provides. Next, we will elaborate on the models it currently offers, and the statistics that are available for the promotion of explainability and transparency of the forecasts. There, we will also address some of the updated functionalities in the new version of PrecisionView™.

Information Provision and Visuals
As discussed above, the software stack for PrecisionView™ is said to be agnostic. This means that the analytics process and the dashboarding can differ greatly between different implementations. Both the specifics behind the analytics process to obtain the forecasts, as well as the information provision and visualisations provided in the dashboard largely depend on the chosen software stack and the specific needs of the company in question. In this research, we made use of the so-called prototype solution. The prototype solution's software stack uses R for the analytics part, Tableau for the dashboarding, and the data is stored locally in excel format. The reason for storing data locally is because it is a prototype. However, as discussed above, the data platform component in the software stack for PrecisionView™ also offers several possibility for the online, centralized storage of data. The information provision for this solution consists of an executive summary, the sales details, a working capital dashboard and the planning overview.

1. **Executive Summary**
   The executive summary provides an overview of the planned, actual and forecasted values for

the most important financial items, as shown in Figure 2.7. These financial items are sales, Adjusted Gross Margin (AGM), General and Administrative expenses (SG&A), Operating Margin (OM) and working capital. More specifically, it contains the following components:

- **Executive KPIs:** provides an overview of the Key Performance Indicators (KPIs) for each of the financial items listed above and the quarterly Earnings Per Share (EPS). For each of these items, it shows the absolute value, as well as the yearly and quarterly growth percentage.
- **Predictive Analytics Scenario:** shows a plot containing the operating plan, the likely forecast and the high- and low driver forecast for the financial items net sales, AGM, SG&A, OM and working capital for the selected year. The high- and low driver forecasts indicate the expected values for the financial item in question when the drivers follow a best or worst case scenario respectively.
- **Detailed Analysis:** provides more in-depth analysis for a selected financial item through a number of plots:
    - The variance between the actuals, the model's forecast and the business's forecast.
    - The cumulative variance between the plan vs. the actuals and the plan vs. the forecast.
    - The predictive analytics scenarios for a 3-year span.
    - The actual versus the forecasted monthly growth.
    - A breakdown showing how much of the financial item in question originates from each of the business segments and sub-segments, for both the current and the previous year.

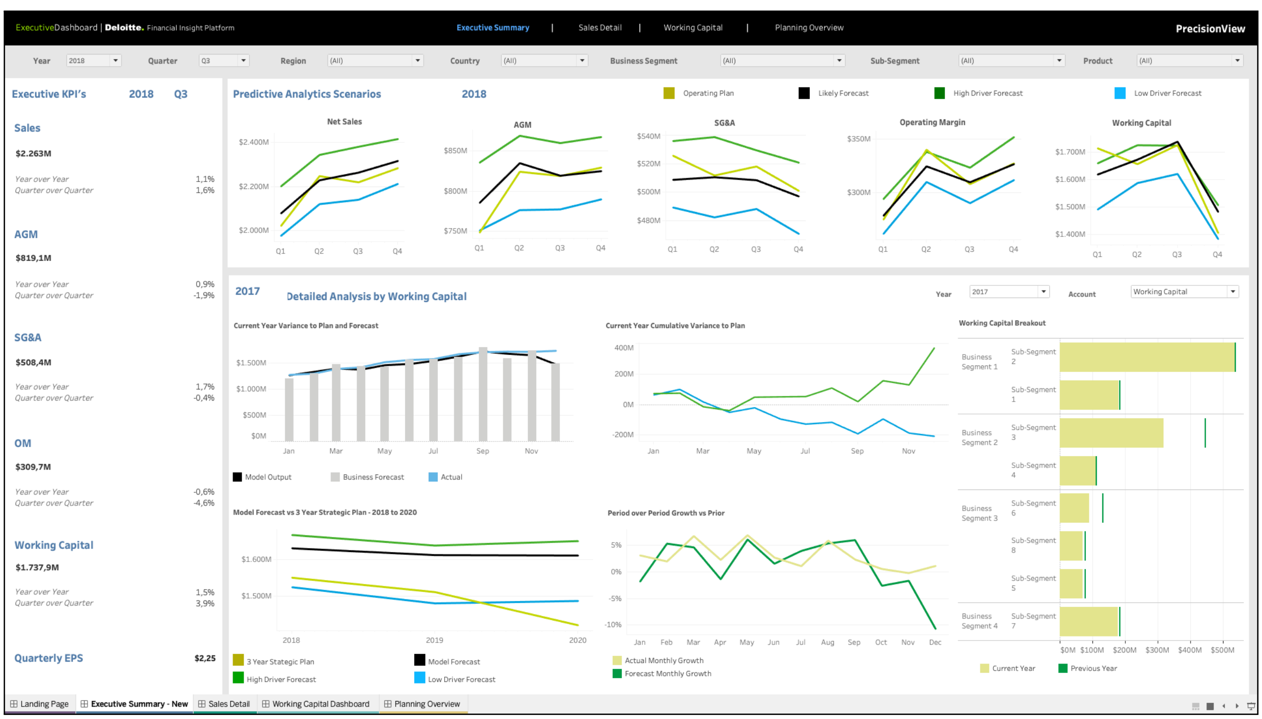


Figure 2.7: The Executive Summary sheet in PrecisionView™'s Tableau dashboard.

2. **Sales Details**
   The sales details zoom in on the sales growth, the change in operating margin and profitability amongst different business segments, sub-segments and products, see Figure 2.8. It contains the following information:

   - **Growth and Margin Trends by Business Segments and Products:** provides a graph in which the % change in net sales and operating margin compared to last year is plotted for each business segment. The size and colour indicate the amount of absolute change and whether this is an increase or decrease, respectively. The same plot is given for either the sub-segments or the products, depending on the option selected.
   - **Profitability Drilldown:** depending on the option selected, it shows which product, sub-segment or segment is most profitable in terms of price and volume. Furthermore, it contains a graph plotting the sales actuals versus forecast, and the operating margin actuals versus forecast on a monthly basis.

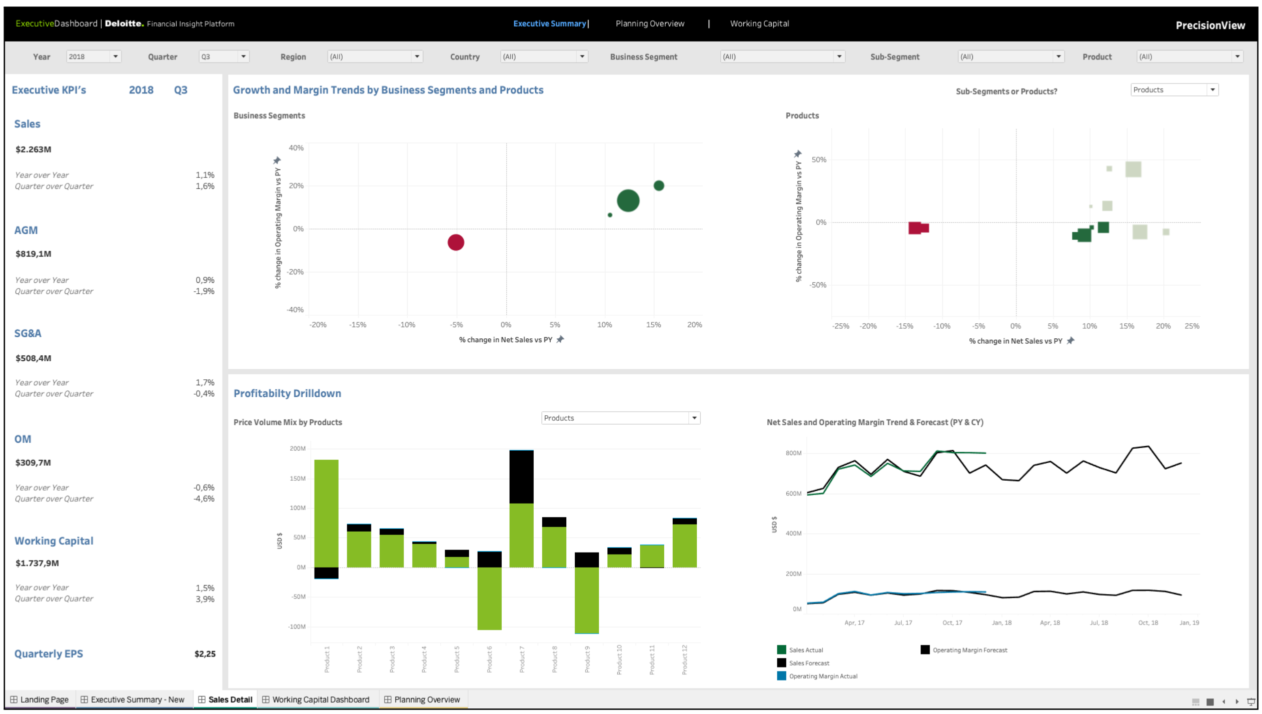


Figure 2.8: The Sales Details sheet in PrecisionView™'s Tableau dashboard.

3. **Working Capital Dashboard**
   This sheet lists the KPIs for working capital and provides insights on the individual financial items that influence the working capital, as displayed in Figure 2.9. These financial items are accounts payable, inventory, accounts receivable, Working Capital (WC) turns and the liquidity ratio.

   - **Business Comparison:** shows the actual value and the growth percentage compared to last year for the selected financial item, e.g. accounts payable. These values are broken down based on business segment and sub-segment.

- **Year over Year Comparison:** provides an overview of the total actual value and growth percentage for the selected item over a 5-year period.
- **Margin Analysis:** the margin analysis plot show the sales actuals or forecast in case a future period is selected, and the OM and AGM percentage on a monthly basis.

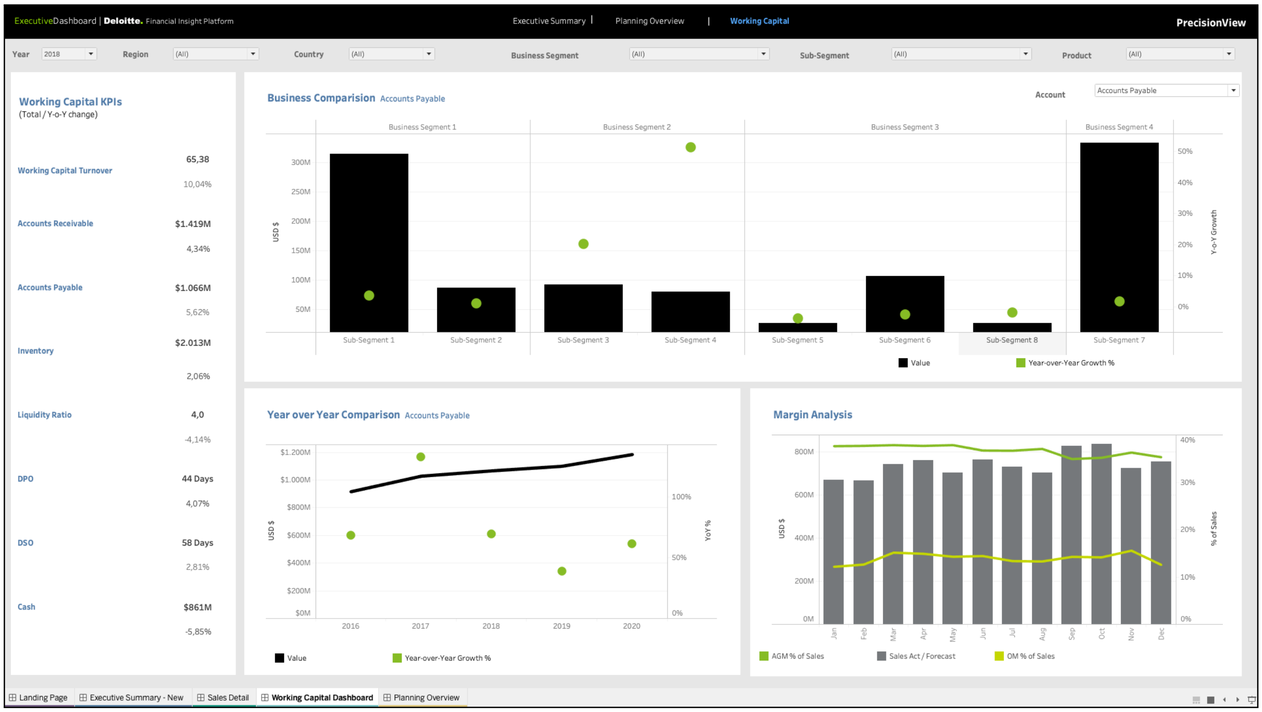


Figure 2.9: The Working Capital sheet in PrecisionView™'s Tableau dashboard.

4. **Planning Overview**
The last component, planning overview, is the main Financial Forecasting component, see Figure 2.10. It contains the accuracy of the different forecast models, the forecasts for the main financial items and the scenario or driver analysis.

- **Model Accuracy:** for a selected forecast model, the model accuracy plots the actuals versus the model's output over a 7-year period. This graph is aimed at providing insights into the accuracy of that model. The closer the outputs of the forecast model are to the actuals, the higher the accuracy of the forecasts. Furthermore, it provides the MAPE for the predictions made on the test set and the training set, as well as the overall MAPE. MAPE expresses the accuracy of a prediction model, by calculating the difference between the actual value A_t and the forecasted value F_t, and divides it by the actual value A_T:
$$\frac{(A_t - F_t)}{A_t}, \tag{2.12}$$
where t is a point in time, e.g. $t =$ Jan 2017. Next, (2.12) is summed for every point t and divided by the total number of points N. Lastly, the number is multiplied by 100% to obtain the percentage. The lower the percentage, the lower the error rate and hence the higher the accuracy.

- **Forecast:** shows the likely, low and high forecast for each of the financial items net sales, SG&A, AGM and operating profit, on a quarterly basis for the selected year. The user has the option to change between a P&L and a working capital view for the forecasts.
- **Drivers' Analysis:** enables the user to adjust the values of each of the drivers that influence the financial items plotted in the 'Forecast' components and see the effect of the changes on the forecast. Furthermore, it contains a plot that shows the distribution of values for each of these drivers. More specifically, it indicated what the low, forecasted and high values for a specific driver are, in order to make informed decisions when changing driver values and performing scenario analysis.

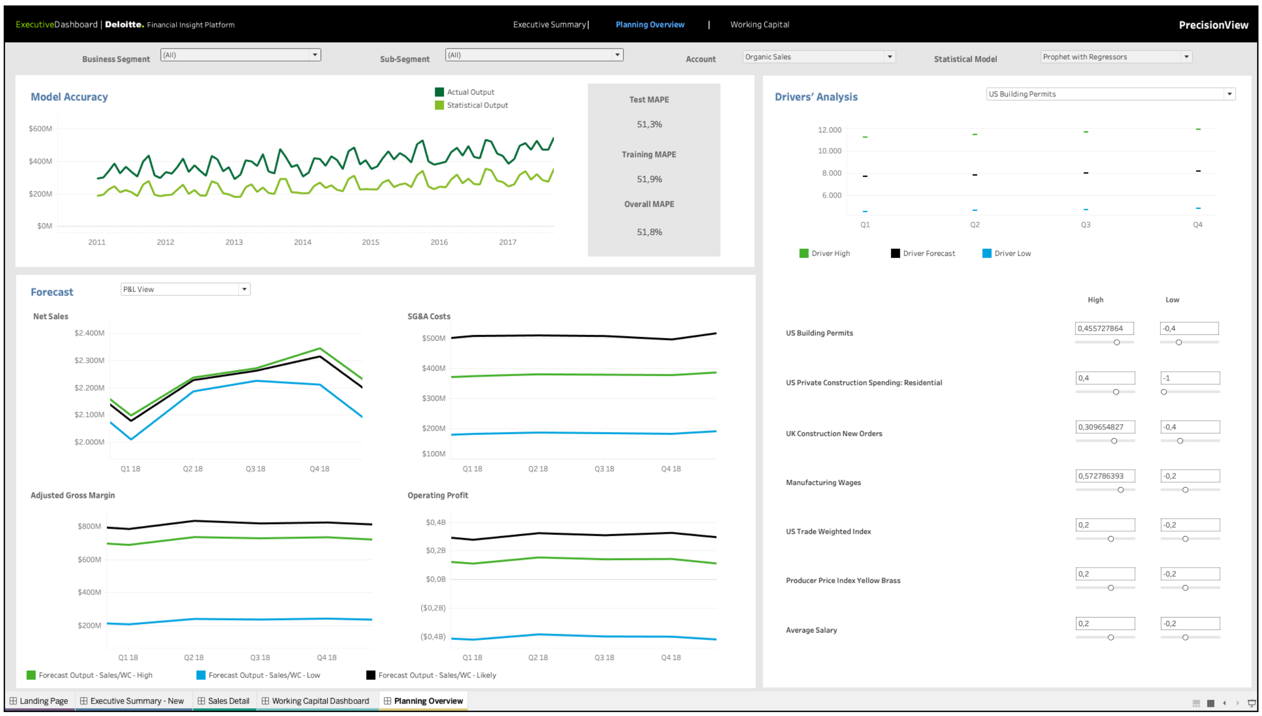


Figure 2.10: The Planning Overview sheet in PrecisionView™'s Tableau dashboard.

Model and XAI Usage

The current version of PrecisionView™ offers 5 different models for forecasting financial items. These models have all been discussed in section 2.3.2. They are multi-linear regression, ARIMA, ARIMAX, Prophet and Prophet with Regressors. In the new version, several models will be added to PrecisionView™, namely E-T-S, Naïve, NNETAR, RWF_Drift and TBATS.

- **Naïve.** Naïve is a method for time-series data and is frequently used to compare complex forecasting methods for benchmark purposes. Naïve simply sets all forecasted values equal to the last observed value [HA19]. That is, $\hat{y}_{T+h|T} = y_T$.
- **RWF_Drift.** Drift is a variation on the naïve method. However, unlike naïve, drift does allow forecast values to increase or decrease over time [HA19]. This change is determined by the average historical change detected in the data, and is called the drift. For h periods into the

future, the forecast is given by:

$$\hat{y}_{T+h|T} = y_T + h(\frac{y_T - y_1}{T - 1}) \tag{2.13}$$

- **ETS.** ETS is another method univariate time-series forecasting method, meaning it accepts a single input, namely the historic values of the variable it is trying to predict. ETS stands for Error, Trend and Seasonality and is an extension of the exponential smoothing models discussed in section 2.3.2.

- **TBATS:** TBATS stands for Trigonometric Box and Cox Transformation, ARMA errors, trend and seasonality. It makes use of, amongst others, the exponential smoothing and ARIMA approaches discussed in section 2.3.2. Note that ARMA models are equivalent to ARIMA models, except for the integration part, meaning no differencing is applied. The TBATS method is used to model time-series data that contain complex seasonlity [HA19]. Examples of complex seasonality include situations in which multiple seasonal patterns exist. Consider, for example, a call centre where there is a daily seasonality regarding the number of calls, i.e. peak times, but also weekly seasonality, i.e. more calls on Fridays.

- **NNETAR** NNETAR stands for Neural Network Autoregression and is a NN approach for time-series data. In section 2.3.2 we saw that autoregression models, such as ARIMA, use the lagged values of a time-series as their inputs [HA19]. With NNETAR, these lagged values are used as the inputs for a neural network. The benefit of using neural networks is that they can model more complex, non-linear relations between the input and the target variable it is trying to predict. NNETAR is similar to the neural network that we discussed in section 2.2.2, as it also has one hidden layer.

PrecisionView™ currently does not incorporate XAI techniques. However, similar to the Financial Forecasting solutions discussed in the previous section, it does offer certain statistics to provide some level of insights into the forecasts and their accuracy. As discussed above when elaborating on the Planning Overview, PrecisionView™ currently provides two sources of information on the forecast accuracy. Namely, the plot containing the actual versus the forecasted values, and the MAPEs for the test, train and overall data set. Furthermore, the Drivers' Analysis as well provides some insight into the forecast models by illustrating, to a certain extent, what happens with the forecasts when the values of their inputs change. In the updated version of PrecisionView™ that is currently being developed, a number of functionalities and statistics will be added that enhance the user's insight into the forecasts. In the current version of PrecisionView™, the driver selection and the training of the forecast models is done beforehand. This means that users are not able to indicate which drivers should be used to forecast a certain financial item. This will change in the updated version, where users can retrain models based on the drivers they themselves have selected. To assist them in this process, the following statistics will be provided:

- **P-values:** a measurement that indicates the probability that we only obtained certain test results purely out of coincidence. The lower the p-value, the higher the confidence level that our results are representative for real-world data.

- **Best fit line:** represents the forecast line of the models that is selected as the best fit, i.e. the best performing model in terms of accuracy. The best fit line is determined by comparing the MAPEs of all forecasting models and selecting the model with the lowest MAPE.
- **Multicollinearity:** indicates the amount of information that two or more input variable have in common [HA19]. Consider, for example, a model for the prediction of house prices that uses both the number of rooms and the amount of m^2 as input variables to predict the price. In this case, there is multicollinearity between the number of rooms and the m^2, because if one increase, then in most cases so does the other. When training forecast models the use of strongly correlated inputs should be avoided because it becomes very difficult to determine the weight of those inputs since it is unclear how much of the observed effect can be assigned to each of the inputs [Mol20]. By providing information on multicollinearity, users can take this into account when selecting the input variable for the creation of a forecast model.

The statistics mentioned above are mostly aimed at guiding the forecast training process itself. However, the new version of PrecisionView™ will also contain a number of statistics to enhance the analysis of forecast models after they are obtained.

- **Driver inspection:** aimed at enabling the client to understand how the input variables, also referred to as drivers, are related and linked to the forecast. Two statistics will be added to promote this driver inspection, namely a driver ranking and the driver correlation. The driver ranking orders the drivers on their importance for the forecast model, which is determined based on their contribution to the forecast output. As we will discuss in section 2.4.5, this driver ranking is similar to the influence method frequently used for XAI purposes. The driver correlation indicates the correlation between each driver and the forecast output, expressed by the coefficient of that driver. The coefficient is the weight assigned to the driver or input variable, as explained in section 2.2.2.
- **Trend inspection:** enables the user to inspect the trend component of drivers or input variables, in the case of time-series data.

2.4 Explainable Artificial Intelligence

In section 2.1, we discussed the importance of predictive analytics tools, such as Financial Forecasting, for the Finance function and the great benefits it can bring. However, as discussed in the introduction, the black-box problem can impede the successful implementation of Machine Learning applications like Financial Forecasting. In section 2.2.2, we illustrated how certain ML models can become difficult to interpret by their users and lead to this black-box problem. Explainable Artificial Intelligence (XAI) is aimed at overcoming this black-box problem and the issues it brings. In this section, we elaborate on what exactly is XAI by looking at the definition, as well as why it is important to use XAI. Next, we will discuss the two possible approaches for applying XAI, the types of methods that exist and the two levels on which XAI can provide an explanation. Finally, we will review the existing work on XAI in the specific context of Finance.

2.4.1 Definition

Explainable Artificial Intelligence (XAI) is a set of techniques aimed at making models explainable without compromising their performance [AB18]. Before elaborating on its exact definition and what it entails, it is important to note that there is a lack of consensus regarding the terminology used in this research field. The most frequently referred concepts, explainability and interpretability or understandability, are used interchangeably. The lack of agreement on what explainable and interpretable models are is a challenge highlighted by several authors [GMR+18, ADRDS+20]. Arrieta et al. address this issue by distinguishing between the concepts of interpretability and explainability as passive and active characteristics of a model, respectively. They refer to interpretability as a passive model characteristic indicating the extent to which the model makes sense for a human observer. They define explainability as an active model characteristic that indicates the actions taken to explain the inner workings of the model [ADRDS+20]. Therefore, they conclude that explainability should be the main objective for the development and use of XAI. In line with this definition, Adadi and Berrada define explainability as interpretable systems whose operations are understandable to humans [AB18]. Furthermore, interpretability is said to indicate that something is understandable [vdBK20] and hence some research measures understandability rather than interpretability [MG00]. Arrieta et al. proposed a reworked definition for XAI to address the lack of consensus that as well focuses on understandability. This definition takes into account both existing definitions and the definition of the word 'explanation' according to the Cambridge dictionary. Furthermore, they argue that the audience of XAI should be a key aspect in this reworked definition for two reasons. Firstly, the reasons for providing an explanation depend on the target audience of the explanation. Secondly, determining whether the used XAI technique has made a certain model explainable or not also depends on the audience it is presented to. Taking this into account, they define explainability as follows: 'Given a certain audience, explainability refers to the details and reasons a model gives to make its function clear or easy to understand' [ADRDS+20]. Forthcoming from this definition of explainability, their definition of XAI is as follows: 'Given a certain audience, an explainable Artificial Intelligence is one that produces details or reasons to make its functioning clear or easy to understand'[ADRDS+20]. The target audience, Finance Professionals, is an important factor for the development of the XAI prototype in this research. Therefore, we follow the definition proposed by Arrieta et al.

2.4.2 The Need for XAI

Above, we provided a definition for XAI and the key factors it aims to achieve, namely explainability and interpretability. Furthermore, we stated that the use of XAI can help overcome the black-box problem. In this section, we will elaborate on why exactly it is important to overcome this problem by discussing the main cause of the black-box problem, and the resulting issues it presents.

There are a number of issues that give rise to the need for XAI. The main cause of the black-box problem is the lack of transparency that occurs for many AI algorithms. This is especially true for Machine Learning (ML) algorithms. According to Adadi and Berrada, promoting a shift towards more transparent AI is the main objective of XAI [AB18]. Here, transparency refers to the degree of insight into the inner workings of an algorithm. The lack of transparency in turn presents several potential threats.

- **Human Bias and Prejudices**

  As ML algorithms are trained on data that may contain human biases and prejudices, there is the risk of algorithms trained on such data to mimic these biases. One example of this, which also illustrates that this problem is not particularly new, is the bias that was found in a computer program for applicant screening used by the St. George's Hospital Medical School in the 1970s and 1980s. The program rejected a great number of applications based on their gender, or on whether they had non-European sounding names. Both biases were not introduced by the algorithm but originated from the original admission procedure [Gar16]. This is only one of the many examples of algorithmic bias. A frequently highlighted example of bias threats within Finance is the automated assessment of credit risks. Companies operating in the credit market increasingly make use of unconventional data on potential lenders to assess their creditworthiness or the risk of defaulting on a loan [PJC19]. ML models trained on this data pose the risk of denying credit to certain lenders based on relations found between the risk of defaulting a loan and a person's postal code or race [DeB18]. When looking at the Finance Function and Financial Forecasting in specific, the risk of bias and consequences thereof are less serious, yet still prevalent. The consequences of bias in Financial Forecasting are less serious because they do not use personal data and hence do not make decisions that could potentially disadvantage individuals based on legally protected traits such as age, religion, race or gender. However, the risk of bias is still prevalent and can have serious consequences when incorrect financial forecasts caused by bias form the basis for a company's decision making. Prior to the rise of ML, financial forecasts were frequently obtained using so-called judgemental techniques, which involves individuals that are very acquainted with the subject matter to provide forecasts based on personal experience, sometimes without the use of any historical data [Sam15]. It has been found that these experts often provide judgements that are too optimistic, driven by motivational factors and the fast-changing and competitive nature of Finance [ÖA98]. Therefore, the forecasts produced by judgemental methods are said to be highly subjective [Sam15]. This problem in part persists with the use of ML due to expert judgement involved in the driver or input identification stage. This involves determining which data inputs are used to train a financial forecast model. As explained in section 2.3.4, driver identification is partly based on statistical methods such as driver correlation. However, it also involves expert judgement to determine which of the statistically correlated drivers actually make sense from a logical business perspective. As discussed in section 2.3.3 and 2.3.4, several Financial Forecasting solutions ultimately leave driver selection up to the user. The overall issue with algorithmic bias is that it can lead to unfair or wrong decisions. In response to this, the GDPR now includes clauses capturing the right to an explanation when automated decision-making is used [GMR+18]. This means ensuring fair decision making is no longer optional and requires companies employing automated decision making to provide explanations to their users. XAI can help avoid bias because it provides insights into the inner workings of a model and thereby exposes errors or reveals that a model is not making decisions or predictions as intended [PJC19], for example, due to bias.

- **Lack of Trust**

  Furthermore, algorithms that are difficult to interpret due to a lack of transparency or high complexity cause humans to have little trust in them [ZLR+18], which impedes the adoption and acceptance of such algorithms. A distinction is made between two types of trust. A user can place trust in a specific prediction made by a model, enough to act on it. Secondly, he or she can trust the model to behave reasonable, resulting in trusting the model as a whole [PJC19]. XAI helps build this trust by offering explanations that justify a certain prediction or decision, even if that prediction or decision might seem illogical or unexpected at first [AB18]. As we will discuss in the following subsections, there are many different approaches to generate an explanation for AI that each provide different types of information on the model or its outcomes. An XAI technique and the information it provides is chosen based on the goal of providing that explanation. However, all types of information contribute to the goal of trust building [vdBK20], illustrating the importance of and emphasis placed on trust. If the users of AI do not trust its predictions, decisions or recommendations, they are often also not going to act upon it [PJC19]. For the Finance Function and Financial Forecasting in specific, this leads to financials not feeling comfortable enough to rely on the presented forecasts and return to manual determining the forecasts. This impeded the shift from operational to business finance and the focus on value-adding activities as discussed in sections 2.1.2.3 and 2.1.2.4. For the Financial sector, the generation of trust is given even more importance, due to the higher regulatory and societal standards this industry is held to [vdBK20]. Here, trust-building does not only include the financials working in the FSI, but also the end-users, i.e. their customers. Those customers want to know if the decisions being made for them by a machine can be trusted.

- **Lack of Understandability**
  With ML algorithms increasingly being used in critical decision making, the issue is no longer only one of fairness and trust. As stated in the previous section, understandability is another important goal of XAI [AB18]. If there is a lack of transparency regarding the inner workings of a model, then the level of understanding that users have of a model is negatively impacted as well. Note that although understanding might help build trust, trust and understandability are not the same nor does understanding directly imply trust or vice versa [vdBK20]. A user might trust a model to make logical or reasonable decisions or predictions, but might still not fully understand how those decisions or predictions were derived. Conversely, understanding might be present while trust is still lacking. For example, although one understands how the algorithm behind self-driven cars works, he or she might still not completely trust it [vdBK20]. Understanding becomes increasingly important, especially for users of algorithms employed in critical contexts. When used for critical decision making, users require a certain level of understanding of how outputs were derived in order to verify and justify those outputs [ADRDS+20]. Examples of such critical decision-making contexts include medicine, military and transportation. However, understandability is also a very important factor for the Finance Function in particular. This is because verification of models goes beyond justifying whether decisions are fair and can be trusted. Understandability aids in controlling unforeseen situations and the improvement of models [AB18]. If a user understands how a model works, he or she can also identify flaws and errors more quickly. Therefore, in the case of the Finance Function, our expectation is that the capability of understanding a model makes a great difference in how forecasts are used for decision making. Without understandability, a user may simply incorporate a predicted significant drop in revenue into his proposed plans and budgets because he or she trusts the model. With understandability, he or she also has the ability to signal why this sudden deviation was predicted and take actions

to mitigate the risk of this actually happening. Furthermore, understandability promotes the improvement of models because if users understand how a system works, they are also capable of making recommendations on how to improve that system's performance [AB18].

There is another issue caused by the lack of transparency concerning the ML research field. It is not stated in the list above as it does not directly impact the users of an AI model but is still interesting to note from an academic perspective. Namely, the use of XAI is said to enable greater transparency amongst scientific discoveries in this ML research field. This is important considering the current gap between the ML research community and business areas such as the Finance Function [ADRDS+20].

In this section, we discussed the main purpose of XAI, namely increased transparency, and the objectives it intends to achieve, namely preventing bias and increasing trust and understandability. As mentioned previously, the audience of AI algorithms or models plays an important part in determining which of these objectives are important [ADRDS+20]. Based on the above, the two main objectives that this research focuses on in the validation of the proposed XAI techniques are increased understandability and trust, as stated in the hypotheses in section 1.3. The reason for not explicitly measuring a reduction in bias is because, as mentioned above, bias is less present in Financial Forecasting because it does not involve personal data. The risk of introducing bias by letting financials retrain models using drivers selected on their own assumptions is also not present in this research. This is because the experiment in this research only involves users working with pre-trained forecasting models and hence they do not have the ability to retrain models.

In the remainder of this section, we will elaborate on the different approaches to develop an explanation, the different types of explanations that can be provided using XAI and the specific XAI methodologies that exist.

2.4.3 Intrinsic vs. Post-hoc Approach

Most taxonomies of XAI methods found in the literature start with a division between approaches based on where in the model development process an explanation is generated. More specifically, there are two approaches, namely the intrinsic and the post-hoc explainability approach.

- **The Intrinsic Approach**
  The intrinsic approach is aimed at developing AI or ML models that are by themselves interpretable for humans. Models that are considered intrinsically interpretable include linear and logistic regression models, decision trees, k-nearest neighbours, rule-based learning, general additive models and Bayesian models [ADRDS+20]. The models that fall under the intrinsic approach are also commonly referred to as transparent models. Arrieta et al. further classify transparent models based on the type of interpretability contained in the model [ADRDS+20]. The first transparency class, algorithmic transparency, expresses the ability of a user to understand how the model maps inputs to outputs. The second transparency class, called decomposability, relates to the explainability of each of the individual components of the model, e.g. the interpretability of the inputs, the parameters and the computations used by the model. The third transparency class is concerned with a user's ability to simulate the line of reasoning of a model, referred to as simulatability. These classes are said to be predecessors

of each other, meaning that decomposable models are by definition also algorithmically transparent, and a model that is simulatable is also decomposable. A linear regression model, for example, is considered relatively simple to interpret and hence users may understand its inner workings (algorithmic transparency) and its individual components (decomposability). However, if the linear model takes in a large number of inputs, the simulatability of the model is impeded. This is because as explained in section 2.2.2, if the relations between the inputs and the output are relatively simple, but the input space consists of dozens of inputs it becomes considerably difficult for a human user to simulate the process of the model.

- **The Post-hoc Approach**
  The other approach taken in developing explainable models is called the post-hoc approach. This approach is concerned with models that are very complex, for example due to a large input space or non-linear relation, and are therefore by itself not interpretable. Post-hoc approaches consist of a set of techniques used to provide an explanation for already developed models without altering their inner workings. This approach is also referred to as reverse engineering, which entails reconstructing an explanation for an uninterpretable model only using the inputs and the given outputs of the black-box model. There are many different forms in which these explanations can be provided. The explanation categories most commonly referred to in the literature are textual explanations, visualisations, example-based explanations, explanations based on influence methods and explanations based on knowledge extraction. Influence methods provide an explanation for a model by looking at the features of the model and their importance, relevance or attribution to a prediction [AB18]. Knowledge extraction methods, also referred to as simplification methods, proceed by extracting some type of knowledge from the black-box model by means of simplification. One way to do this is, for example, to extract rules from the model's inner working that approximates the model's decision-making process. Other knowledge extraction methods include surrogate models and model distillation. Example-based methods aim to explain the behaviour of a model based on a single particular instance in the dataset. One approach to providing example-based explanations is that of counterfactual explanations, which are based on a way of reasoning that is often used by humans. Namely, counterfactuals provide explanations by considering why this instead of that decision was made [Byr19]. They can help identify which conditions had to be satisfied for the outcome or prediction of the model to be different. In section 2.4.5 we will go deeper into the techniques developed for all the above mentioned post-hoc explanation categories.

If there are interpretable models, why then develop complex models that require additional time and effort to be made explainable post-hoc, one might ask. The reason for this is commonly referred to as the accuracy vs. interpretability trade-off. Generally speaking, simple, more transparent models are better interpretable but tend to have a lower prediction accuracy. Conversely, sacrificing transparency by choosing more complex models often brings higher accuracy in return. If, for example, the relationship one is trying to capture with a model is not linear, linear models will result in poor accuracy and hence will not suffice. In such a case more complex models are required in order to accurately capture the complex relationships between the input and the output variables. Explanations of the model can then be developed afterwards using post-hoc techniques.

2.4.4 Model-specific vs. Model-agnostic Methods

The literature on XAI makes a second distinction between XAI techniques based on the applicability of the method used. Model-specific methods can only be applied to specific types of models. Model-agnostic techniques, on the contrary, are applicable to any type of model and hence separates the explanation of the model from the actual prediction, decision or recommendation given by the model [AB18].

- **Model-specific**
  The intrinsic approach discussed in the previous section is by definition model-specific [AB18]. This is because the choice of intrinsic model is dependent on the input data, the desired output and the type of relationship to be modelled between the input and output. Consider, for example, some of the intrinsic models mentioned in section 2.4.3. If we want to create a model that predicts a categorical value, logistic regression could be a good choice, whereas linear regression would immediately be dropped as an option. Conversely, if the desired model is to predict continuous numerical values, linear regression could suffice, but logistic regression is considered unsuitable. However, if we want to create a model for which the relation between the inputs and outputs is non-linear, neither linear nor logistic regression would suffice. Hence, the choice of intrinsic method is by definition dependent on the model to be developed, and hence always model-specific. The downside with model-specific explanation methods is that we are bound to the set of models for which they are suitable. This can lead to the use of models that are sub-optimal for the given problem in terms of the accuracy that can be achieved with it.

- **Model-agnostic**
  Conversely, post-hoc methods are mainly model-agnostic [AB18] However, this does not always has to be the case [GMR+18]. Guidotti et al. discuss a number of techniques identified in the literature aimed at providing post-hoc explanations whose applicability is limited to a specific type or class of algorithms. Examples include rule-extraction techniques specifically tailored to Neural Networks or Support Vector Machines, Saliency Masks suited for explaining Deep Neural Networks and single decision-tree techniques for the explanation of Neural Networks or Tree Ensembles [GMR+18]. Hence, amongst each of the post-hoc explanation types mentioned in section 2.4.3, both model-specific and model-agnostic implementations can be found. However, most recent research is done into model-agnostic techniques [GMR+18], as they provide the obvious advantage of being applicable to any type of algorithm.

2.4.5 Local vs. Global Explanations

Lastly, the existing literature also makes a distinction between explanations based on the scope of the interpretability that is desired [AB18]. More specifically, they distinct between local and global explanations. Given a specific instance, local explanations provide the reasoning behind a single prediction or decision for that instance. Global explanations, on the contrary, are aimed at providing an explanation behind the overall logic of the model. They should provide an understanding of how the input space is mapped to all possible outputs of the model. Global explanations are particularly useful when used to inform decisions on a population level, such as predictions pertaining to climate change [AB18]. This is because in such cases it is more helpful for the user to provide information on the overall reasoning of the model, rather than explaining the output for each instance individually. Interestingly, Guidotti et al. distinguish between three rather than two types of explanations based on the scope. These are the model explanation problem, the outcome explanation problem and the model inspection problem. The outcome explanation problem is concerned with providing explanations for a single outcome or prediction, and hence produce local explanations. The model explanation and model inspection problems are both concerned with providing global explanations but differentiate between the way the explanation is provided. The model explanation problem provides explanations by developing interpretable and transparent models based on the original black box model, i.e. it refers to surrogate model techniques. The model inspection problem, however, provides explanations by providing visual or textual representations of certain properties that allow the inspection of the black box model in question.

In section 2.4.3 we listed the different categories used to create post-hoc explanations. Some of these post-hoc methods provide a local explanation while others are aimed at providing global explanations. We will discuss three of them here.

1. **Knowledge extraction methods**
   As already briefly discussed, knowledge extraction methods can be roughly divided into rule-extraction techniques and surrogate models. One example of a well-known technique for the creation of surrogate models is that of Local Interpretable Model-Agnostic Explanations (LIME) [RSG16], which produces explanations of the local type. LIME investigates what happens with a model's predictions when given permuted samples of the input data and builds a new training set based on these samples and their predictions. Using this dataset it trains an interpretable model that locally mimics the predictions of the original black-box model. There also exist techniques for the creation of surrogate models that provide global explanations. The specific implementation details depend on the chosen technique, but in general such techniques use the original training set complemented with the predictions given by the black box model for each of the test instances. Next, an interpretable model like the ones listed in section 2.4.3 is selected and used for supervised training on the labeled training data [Mol20]. For LIME, for example, there exists an extension called SP-LIME, aimed at creating such global surrogate models.

2. **Influence methods**
   Influence methods approximate the effect or influence of the inputs - referred to as features - on the model's prediction, or the input's importance for the model's predictions in terms of their influence on the accuracy. As with knowledge extraction methods, influence methods as well can be used for the creation of both local and global explanations.

   - **Partial Dependence Plot (PDP).** One example of a technique that provides an explanation for the average effect of a particular feature on the predictions of a model is called the Partial Dependence Plot (PDP). Suppose that we have a model that predicts the number of people at the park based on the temperature that day, the season, and whether it is a work- or weekend day. If we want to know the average effect that the temperature has on the model's prediction, we can construct the PDP for the input variable $x_{temperature}$. This construction starts by collecting all the possible values that

Instance	Temperature	Season	Weekday	Prediction
1	21°C	summer	false	467
2	22°C	summer	true	242
3	17°C	summer	true	119
...	...	...	...	...
100	2°C	winter	true	36

Table 2.2: Example of data instances for a model that predicts the number of people at the park.

Instance	Temperature	Season	Weekday	Prediction
1	2°C	summer	false	98
2	2°C	summer	true	52
3	2°C	summer	true	52
...	...	...	...	...
100	2°C	winter	true	36

Table 2.3: Example of updated temperature values for data instances of a model that predicts the number of people at the park.

$x_{temperature}$ holds, e.g. $x_{temperature} \in \{2°\text{C}, ..., 17°\text{C}, 21°\text{C}, 22°\text{C}\}$. Next, we are going to replace the temperature value for each instance with one of possible values of $x_{temperature}$. An example of what these data instances and their corresponding predictions could look like is given Table 2.2. After replacing the temperature value of each instance with the first possible value of $x_{temperature}$, namely 2°C, we compute the new predictions and obtain the instances depicted in Table 2.3.

We average the new obtained predictions and repeat this process for all other possible values of $x_{temperature}$. Hence, we are computing the average prediction when all instances have $x_{temperature} = 2°\text{C}$, when all instances have $x_{temperature} = 17°\text{C}$, etcetera. Finally, we can plot the average predictions for each of the possible values of $x_{temperature}$ and obtain a visual representation of the effect of feature $x_{temperature}$ on the model's prediction, see Figure 2.11. As PDPs plots compute feature effects by averaging the predictions of a set of instances, it provides a global explanation.

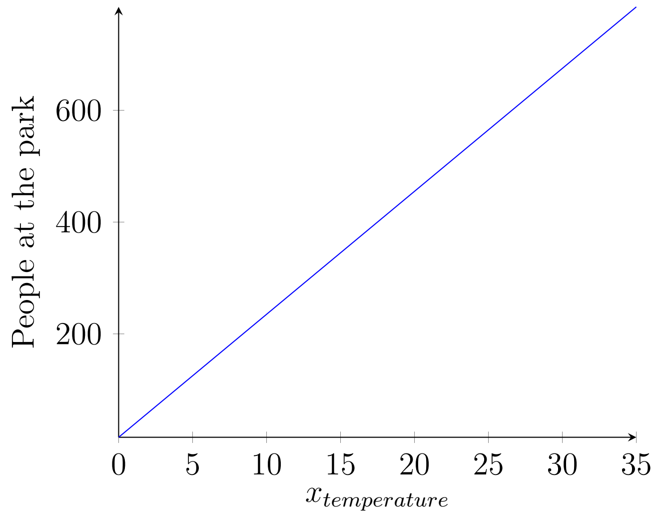


Figure 2.11: Visual representation of the average effect of $x_{temperature}$ on the predicted number of people at the park.

- **Accumulated Local Effects (ALE).** Another technique similar to PDPs is that of Accumulated Local Effects (ALE). However, unlike PDPs, they can deal with correlated features without introducing bias [AZ16]. PDPs introduce bias as a result of the unrealistic instances that are created when changing the value of the feature in question x_i. Assume, for example, that variable $x_{temperature}$ is strongly correlated with some other feature $x_{daypart}$. If we change the value for $x_{temperature}$ in all instances to 37, while leaving $x_{daypart}$ unchanged, we obtain unlikely instances such as $x_{\text{temperature}} = 37°\text{C}$ with $x_{daypart}$ = night. To overcome this issue, ALE only looks at the predictions of the instances with a similar $x_{temperature}$ value, rather than averaging over all instances' predictions. However, this still doesn't fully solve the issue with correlated features. This is because, apart from the problem with unlikely instances, there is also the problem of joint effect between two strongly correlated features. In the above example, there is a joint effect between $x_{temperature}$ and $x_{daytime}$, because the daytime has a direct relation with the temperature and hence we can not be sure which amount of the observed effect can be attributed to either the temperature of the daypart. To overcome this second issue, ALE computes the difference between the predictions instead of the averages. For example, to estimate the effect of $x_{temperature} = 20°\text{C}$, ALE collects all instances with $x_{temperature} = 20°\text{C}$, obtains the predictions for these instances when changing their $x_{temperature}$ value to 21°C and subtracts the predictions obtained when changing their $x_{temperature}$ value to 19°C.

- **Variable Attribution.** Another technique that computes the effects of the features on the model's prediction is variable attribution. Unlike PDP and ALE, variable attribution provides a local explanation. There exist different techniques to compute the attribution of the model's features. A well-known technique is SHapley Additive exPlanations (SHAP) [LL17]. Its aim is to explain the prediction for a specific instance by calculating the amount that each feature contributes to that prediction through the computation of Shapley values which are based on game theory. Shapley values consider the prediction of the model as the 'pay out' of a game, the model's features as the players of the game, and the feature attribution as the distribution of the 'payout' amongst the features [Mol20].

- **Variable Importance.** Lastly, the technique of variable importance provides another way to provide a global explanation. Rather than estimating the contribution of each feature to the model's prediction, it computes the importance of each feature for the accuracy of the model. Feature importance is computed by measuring the prediction error of the model after shuffling the values of a feature. If the prediction error increases greatly, then one can conclude that the feature in question is of great importance for the model's prediction accuracy, and vice versa [Mol20]. Repeating this procedure for each of the model's features presents an overview of the importance of each of the features in comparison to each other. The computation of variable importance is independent of the loss function that is used to estimate the change in prediction error. This means that different loss functions, such as MAPE or RMSE, can be used to determine the loss in accuracy.

3. **Visualisation methods**
   There are a number of ways to provide both local and global explanations in a visual manner. The great majority of model-agnostic visualisation methods, however, are built on top of influence methods [ADRDS$^+$20]. This is because a potential visual explanation needs to be suitable for any type of model, regardless of their type or size of inputs and output. Using model-agnostic influence methods provides a way to extract insightful information from a model, independent of its structure, that also lends itself well to visualisation. PDP's, for example, fall under the category of influence methods but simultaneously provides a graphical representation. The Individual Conditional Expectations (ICE) technique is an extension of PDPs, that plots the feature effect on the prediction for each individual instance, instead of only the average for all instances. Hence, ICE also falls under examples of visualisation techniques that are based on influence methods. Furthermore, SHAP, Shapley values and other variable attribution methods can be visually represented using breakdown plots. Such plots represent the attribution of each feature by means of a bar that, all summed together, add up to the prediction of a specific instance. Lastly, also variable importance values are often graphically represented using horizontal bar charts to allow for easy comparison of the importance of the features amongst each other.

In the current and previous two sections, we provided an overview of the two approaches that can be used for generating XAI, namely intrinsic or post-hoc, the two types of methods that can be used, model-specific or model-agnostic, and the possible scopes for which an explanation can be provided, namely local or global. In sections 3.1 and 3.2, we will discuss the selection criteria for the XAI techniques used in this research, as well as elaborate on those techniques in more detail.

2.4.6 XAI in Finance

In this section, we will discuss the current state of research into the use of XAI in the specific context of Finance. Several of the works discussed here make use of the XAI techniques discussed above in section 2.4.5, while others propose proprietary methods. Furthermore, a distinction is found between works that explore the use of XAI from a technical perspective, and works that focus on the effectiveness of XAI from the target audience perspective. However, we did not find XAI research in the context of Finance that combines both.

One research that proposed a proprietary approach for the explanation of ML applications within Finance is that of Jonker et al. In their research, they provide a method to expose the structure and behaviour of time-series models used in financial services [JBL19]. The developed prototype is a visual analytic approach that incorporates four components to explain the model. These components include the structure of a model, i.e. the relation between the inputs and output, the behavior of time-series components, i.e. indication of the trends and events, scenario analysis functionalities to inspect the change in model response under different circumstances, and integrated commentary to allow for story telling and additional, textual explanations. Based on expert feedback they found that their approach aligns with the mental models users create of the time-series models, and conclude that their visualisation approach therefore facilitates the learning process for these financial models. However, the use cases on which they validated their approach all concern FSI use cases. More importantly, the experts that validated their approach only include users from the financial services domain, such as banks, regulators and funds.

Bracke et al. tested the applicability of a wellknown model-agnostic XAI method in the context of Finance [BDJS19]. More specifically, their research focuses on increasing the explainability of ML models that predict the risk at loan defaulting. The proposed framework uses Shapley values to estimate the influences of features such as interest rates and loan-to-value ratios in order to explain the predicted risk at default. Furthermore, they apply clustering methods to partition the different types of loan requests into similar groups. They validate their approach by testing it on a mortgage defaults data set and demonstrating the possible use case scenarios from the perspective of different stakeholders operating within the FSI. They find that the approach improves the quality assurance, understanding of the models and assists in performance testing. However, no validation amongst the potential target audience of the proposed explanation approach was performed.

Research conducted by van den Berg and Kuiper investigates the use of XAI in Finance from the perspective of the target audience. More specifically, they proposed a framework that helps identify the different types of explanations required by the different stakeholders involved in financial AI applications. Focusing on the perspective of the target audience, the different types of explanations identified in their framework are completely disconnected from the technical methods of techniques that can be used to implement them. Again, the financial context in which the research is conducted concerns the FSI. Hence, the stakeholders identified in their framework include loan applications, financial advisers, loan officers, auditors and bank regulators.

In section 2.2.3, we discussed previous research into the use of ML for financial applications. It was found that the majority of the works was conducted in the context of the Financial Services Industry. When reviewing the current state of research into the use of XAI in financial context, we observed a similar trend. The existing works on the use of XAI in financial context discussed here all focus on use cases in or the target audience of the Financial Services Industry. Indeed, to the best of our knowledge no research has been conducted into the use of XAI in the specific context of the Finance Function. Although the data involved in both financial contexts is similar, the models used differ. For example, as discussed in section 2.2.3, ML applications in the FSI commonly use anomaly detection or classification techniques, whereas Financial Forecasting applications in the Finance domain make use of regression. Furthermore, the target audiences in both differ in terms of the tasks they perform and the knowledge they posses. Hence, XAI techniques found to be effective by the target audience of the FSI, can produce different results when validated amongst the target audience of the Finance Function. This means that findings concerning the use of XAI in the FSI can not directly be extended to the Finance Function. Therefore, the selection of XAI techniques to be used in the developed prototype is based on the research discussed in sections 2.4.3, 2.4.4 and 2.4.5. In sections 3.1 and 3.2, we will discuss the selection criteria for these XAI techniques, and elaborate on those techniques in more detail.

3 System Design

In this section, we will discuss the design of our XAI prototype for Financial Forecasting solutions. This prototype is implemented as part of Deloitte's existing software solution called PrecisionView™. However, the aim of this research is to provide a generic approach for the use of XAI in Financial Forecasting solutions. Therefore, in this chapter we will focus on the logical and technical system aspects of our prototype, disregarding the software-specific implementation details for PrecisionView™. In chapter 4, we will go discuss these implementation details for PrecisionView™ specifically. Here, we will first discuss the design considerations used to determine the set of XAI techniques used in the prototype. Secondly, we elaborate on the specifics of the set of applicable XAI techniques and discuss their advantages and disadvantages. Based on this discussion of applicable XAI techniques and aforementioned design considerations, we elaborate on the selection of the techniques used in our prototype. Next, we present the logical architecture behind our prototype and illustrate how it is linked to the XAI Generation Module. The XAI Generation Module is the main component of our XAI prototype, responsible for the application of the selected XAI techniques. Lastly, we discuss the technical specifications and selection criteria of the library used in this XAI Generation Module, elaborate on its architecture, as well as the usage of the Module.

3.1 Design Considerations

Before we can select the set of XAI techniques that will be used in our prototype for XAI enabled financial forecasting, we firstly define the requirements that the selected set of techniques has to meet. In this section, we outline the four main considerations for our system design.

- **Limit number of techniques**
  In determining the XAI techniques to be used for our prototype, we concluded that the number of techniques has to be limited. This is because a big obstacle in the verification of the effectiveness of XAI techniques is the learning curve. It was found that in order for users to understand how to interpret the explanations and to allow for the generation of trust, they require time and need to use the XAI on multiple occasions and for different tasks [HMKL18]. Incorporating to many XAI techniques could result in overwhelming the use and negatively impact the intended effect of XAI. Therefore, we wish to focus on approximately three to five techniques that are most suitable for the financial forecasting domain and that meet the remaining requirements described below.

- **Complementary techniques**
  In addition to limiting the number of XAI techniques in our XAI prototype, we wish to focus on techniques that are complementary to each other. Since we limit the number of techniques selected, it is important that the final selection of XAI techniques is complementary rather than overlapping, in other to prevent the choosing techniques to overlap in terms of the information they provide. To this end, we decided to incorporate both local and global explanation techniques, because, as explained in section 2.4.5, by definition the two types provide explanations on a different scope.

- **Post-hoc, model-agnostic techniques**
  The third design consideration concerns the type of XAI method and approach. More

specifically, the selected techniques have to belong to the model-agnostic class of methods and follow the post-hoc approach. As discussed in section 2.4.4, model-agnostic methods are applicable to any type of model. Furthermore, they often use the post-hoc approach, meaning that explanations can be obtained after the models were developed. The reason that we focus on post-hoc, model-agnostic methods in this research is due to generalisability. These set of methods enable others to replicate the approach proposed in this research regardless of the specific ML models they use and without the need to retrain or redevelop their models. It is worth noting that a potential disadvantage of model-agnostic techniques is that they accuracy of the produced explanations can be lower than that of explanations produced by model-specific techniques. The reason for this is that model-agnostic techniques approximate the ML model they are trying to explain, whereas model-specific techniques tend to mimic the underlying ML model more directly [AB18]. Nevertheless, model-specific models significantly limit the types of underlying ML models for which it can produce an explanation. This can lead to having to use ML models that are less representative of the task at hand and hence negatively impact accuracy. Therefore, the benefits of model-agnostic techniques outweigh those of models-specific techniques.

- **Computational considerations**
  Another important design consideration concerns the computation time required for the chosen XAI techniques. As we will see in the discussion on applicable XAI techniques in section 3.2, each technique comes with advantages and disadvantages. Several of these advantages or disadvantages relate to the mathematical complexity behind the technique. This complexity influences the computational aspect of the technique, influencing the computation time required to compute the explanation. For certain ML models these more complex computations are required in order to deal with specific characteristics or structures of those models. However, if this is not the case, it is preferred to stay away from these more complex computations to avoid an unnecessary increase in computation time. Keeping in mind the generic characteristic of our prototype, requiring it to be applicable to other Financial Forecasting solutions, it is important to limit computation time. This is because there a several factors in Financial Forecasting solutions that influence the computation time. Firstly, the number of independent variables influences the computation time, as we require explanations for the relationship between each independent variables and the dependent variable. Secondly, the number of forecasted financial line items influences the computation time, since each forecasted financial item requires its own XAI data. Thirdly, in the case of local explanations, we not only require an explanation for each independent variable of a forecasted financial item, but also for every interval for which we provide a local explanation, i.e. weekly, monthly, quarterly, etc. This means that for more extensive Financial Forecasting solutions containing forecasts for many financial line items, forecasts that use many independent intervals, or that uses low intervals requiring many local explanations, the computation time can increase significantly and ultimately become problematic. Furthermore, as financial numbers change when time passes, the forecasting models and hence the pairing XAI data needs to be updated on a regular basis.

3.2 Applicable XAI Techniques

Based on the design consideration discuss in the previous section, there are a number of XAI techniques that qualify for the use in our XAI prototype. They include model-agnostic, post-hoc techniques from both the local and global class of explanation methods. In this section we will discuss them in more details and elaborate on their advantages and disadvantages.

3.2.1 Local Explanations

Biecek and Burzykowski refer to local explanations as instance level explanations [BB20]. They offer 4 methods for instance level explanations. The first 3 methods, Break-down, SHAP and LIME provide an explanation by showing the contribution of the input variables to the prediction for a specific instance. The fourth method, Ceteris Paribus profiles, provide explanations by illustrating how a certain variable affects the prediction of a specific instance.

1. **Break-down**
   Break-down (BD) plots belong to the class of influence methods, discussed in section 2.4.5. More specifically, it is a variable attribution technique. `DALEX` provides two methods for BD plots, namely for additive attributions and for interactions. BD plots illustrate, for each input variable x_i, the contribution to the prediction for a specific instance $f(x)$, in terms of the change in prediction when conditioning on other inputs [BB20].

   Figure 3.1 provides a good example of the idea behind this based on a model for the prediction of survival amongst Titanic passengers. The first sub-figure, A, shows the distribution and average of the predictions of survival amongst different features. For example, we see that when considering all input data, the predictions for the survival rates range between 0 to 1 and an average predicted survival rate of roughly 0.235. The second row only considers data instances, passengers in this case, who are eight years old. It can be seen that the average predicted survival rate now increases to 0.505. From this, it can be concluded that the increase in survival rate of 0.27, can be attributed to the 'age' feature. In the next step, we only look at passengers who are eight years old and were in 1st class. Again, we see a slight increase in the average predicted survival rate, namely to 0.591 exactly. Hence, the 0.086 increase in predicted survival rate is due to the 'class' feature. The method behind the BD plot continues to fix the values of consecutive features, such as fare, gender, embarked, to obtain the BD plot for a specific instance.

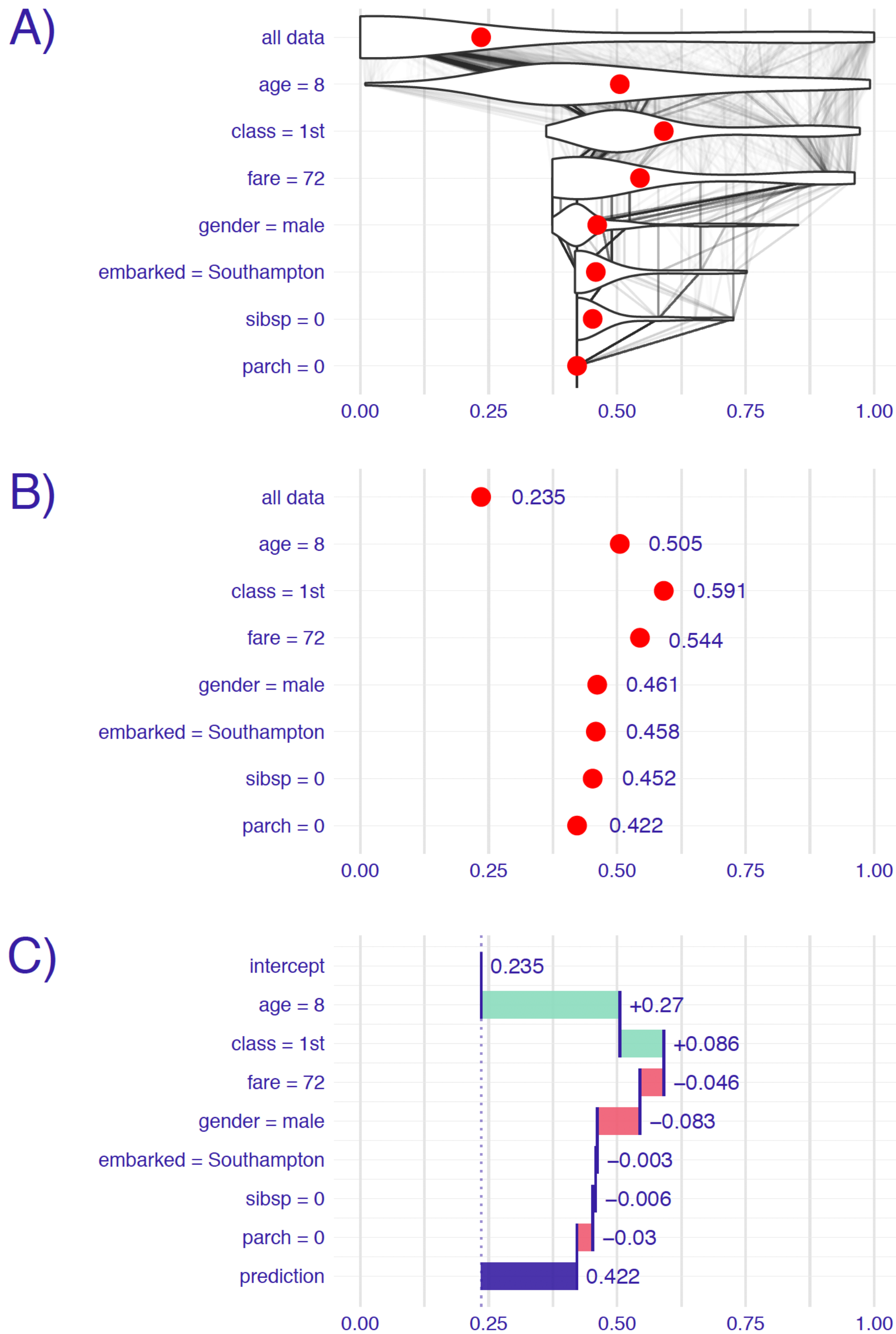


Figure 3.1: Break-down plots and a graphical representation of how they are computed. [BB20]

As mentioned above, DALEX offers BD plots for both additive attributions and interactions. The reason for this is that additive BD plots do not work well on non-additive models with interaction. Interaction indicates the presence of features whose effects on the prediction are dependent on each other. Consider, for example, the joint effect of the day of the week and the temperature on the number of daily bike rentals. In the weekends, bikes may only be rented when the weather is nice enough, but during the week people might rent them regardless of the weather because they have to get to work [Mol20]. As discussed in section 2.3.2, additive models are of the form: $y = \beta_1 x_1 + \beta_2 x_2$, where models with interaction could have the following form: $y = \beta_1 x_1 \times \beta_2 x_2$. If a model contains interactions, then the order in which we change the features of a model to obtain the BD plots affects the influence or effect that is assigned to each feature [BB20]. With additive models it does not matter if we consider feature x_1 or x_2 first, because we add their effects and the result remains the same. Biecek and Burzykowski illustrate the issue by plotting the BD plot for an interactive model, using 10 different feature orderings, see Figure 3.2.

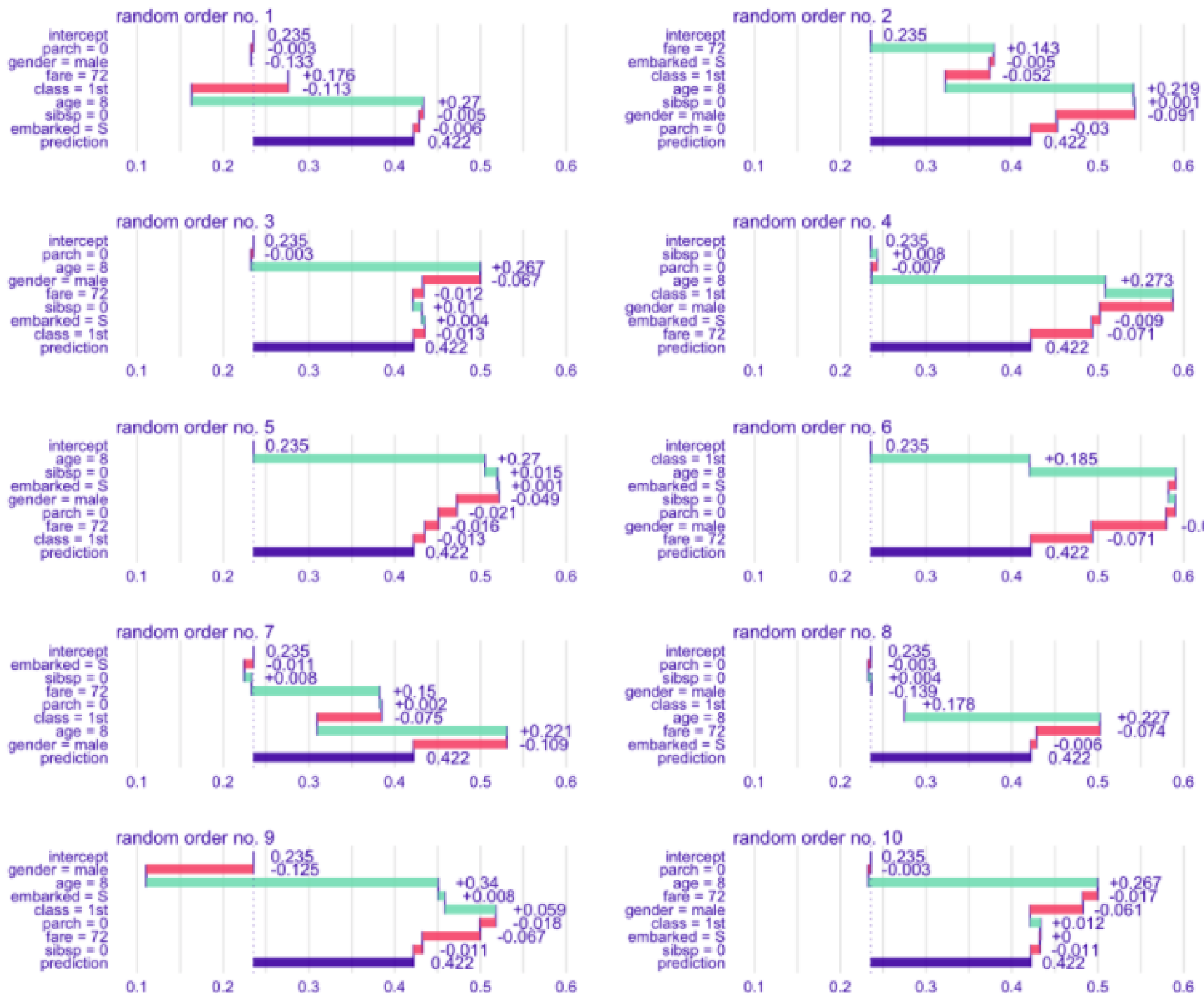


Figure 3.2: Ten different feature orderings and the results BD plots for an interactive model. [BB20]

In Figure 3.2, we see that although the intercept and resulting prediction are the same for every BD plot, the effect or attribution assigned to each feature differs per BD plot. This is due to the change in order and the interaction contained in this model.

2. Shapley Additive Explanations (SHAP)

The idea behind SHAP is founded on game theory, as already discuss in section 2.4.5. Similar to BD plots, it provides an explanation for a specific instance by computing the attribution of each feature to the prediction of that instance [LL17]. Similar to the interactive BD plot, it is an alternative for dealing with non-additive models. SHAP deals with interaction by averaging the attribution values of each feature over all or many possible orderings [BB20].

This is illustrated in Figure 3.3, based on Biecek's and Burzykowski's example of orderings for the Titanic dataset. For example, if we look at the attributions of the feature "age = 8" in Figure 3.2 and compute their sum, we obtain:

$$\frac{0,27 + 0,219 + 0,267 + 0,273 + 0,27 + 0,17 + 0,221 + 0,227 + 0,34 + 0,267}{10} = 0,252$$

This is indeed the average attribution for "age = 8" obtained by the SHAP method, as shown in Figure 3.3. The purple line depicts the box plot for the feature in question and indicates the distribution of the feature attributions of the different orderings, i.e. the minimum, maximum and median of the attributions.

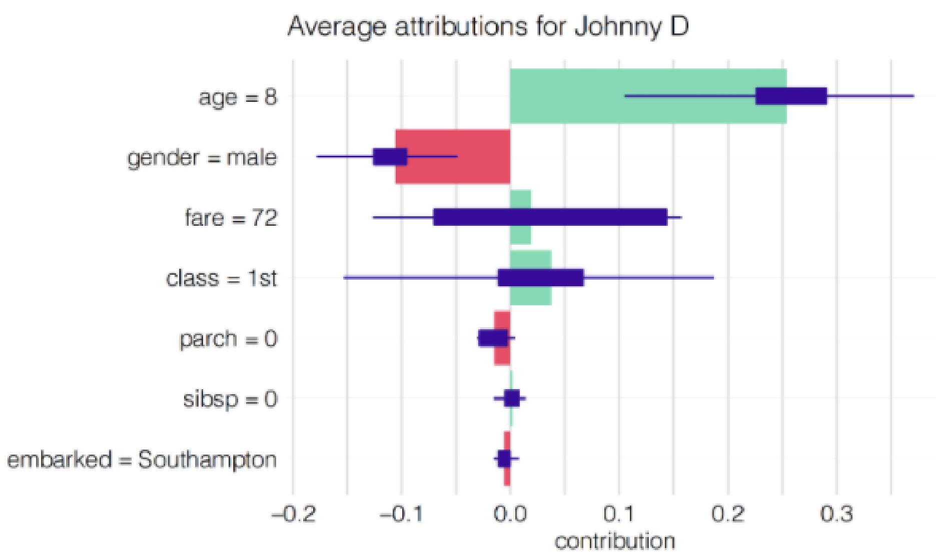


Figure 3.3: The attribution or effect of each feature computed by averaging the attributions of each feature in Figure 3.2. [BB20]

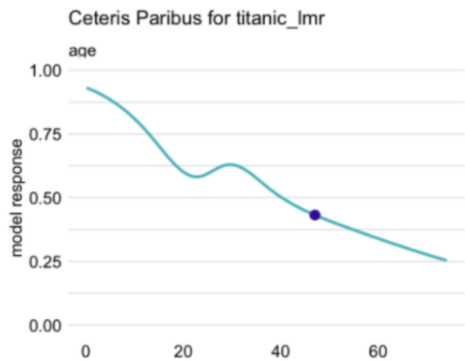
A significant draw back with SHAP is that the computation time can be very long, because it involves computing the feature attributions for many or all possible orderings in order to find the average attribution [Mol20]. Applying SHAP to an additive model generates the same result as applying the additive BD plot method. Therefore, additive BD plots are the logical choice when using additive model without interaction.

3. **Local Interpretable Model-agnostic Explanations (LIME)**
As discussed in section 2.4.5, another category of model-agnostic XAI techniques is that of

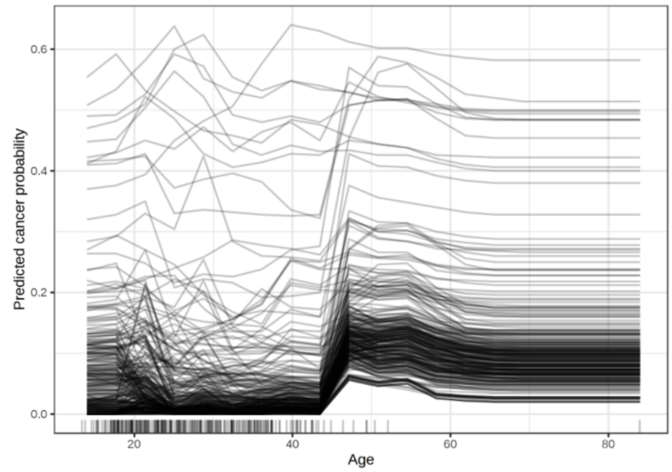
knowledge extraction. LIME is an example of such a knowledge extraction technique and works by generating a local surrogate model of the underlying, more complex prediction model [RSG16]. This surrogate in turn produces explanations similar to those given by the BD plots and SHAP, namely by listing the attribution of the input features to the prediction of the instance in question. Why then use LIME to produce results similar to the BD plot or SHAP? The reason for choosing LIME over BD plot or SHAP method, is that BD plots and SHAP are not suitable for explaining complex models that have a significant amount of input variable or features [BB20]. This is because those methods often assign non-zero attributions to all the features in a model. However, in situations where the model consists of thousands of features, it is desired to set certain features to zero to obtain a simpler, interpretable model. Situations for which this is often the case include models for text or image recognition. Consider, for example, the neural network for image recognition as discussed in section 2.2.2. We saw that this neural network already required 784 neurons for a small image of 28 by 28 pixels. If we would apply the BD plot method to this NN, we would obtain a list of 784 attributions, which is not very intuitive or interpretable anymore to the user [BB20].

Although a suitable XAI technique for explaining models with a large input space, LIME also comes with a number of limitations. Firstly, LIME is said to pose issues when dealing with tabular data containing continuous or categorical explanatory variables [Mol20, BB20]. The reason for this is that LIME zooms in on the instance of interest and compares is to instances or data points close to the instance in question. However, as there are often not many instances that lay relatively close to each other, artificial instances are generated by permuting the values of different features of the instance in question. If the features of such instances are binary, then permuting their values simple entails changing them from 0 to 1 or vice versa. An example of a model with binary features is the NN discussed in section 2.2.2, where each input indicated a pixel in the image and its value, 0 or 1, indicated whether that pixel was present or not. However, when the features of our model are not binary but continuous, determining the values with which we need to permute our features becomes more difficult because continuous values are infinite. A second issue with LIME is that it can produce unstable results, meaning that applying LIME to two closely located data instances can produce very different results [Mol20]. Overall, the use of LIME is said to be limited to model with large input spaces that do not use categorical or continuous tabular data, like text or image recognition [Mol20].

4. **Ceteris Paribus (CP) profiles**
   CP profiles or Individual Conditional Expectation(ICE), as they are also commonly referred to, show the effect of a certain feature's value on the model's prediction. They are particularly useful for what-if analysis, i.e. to see what happens with the prediction if the value of the feature in question changes [BB20]. The difference between the ICE plot and CP profiles in `DALEX` is that ICE plots depict the effect of a certain feature on the prediction for every instance in the dataset, whereas CP profiles only plots the effect for the instance in question.



(a) A CP profile indicating the effect of *age* on the predicted Titanic survival rate for a specific instance [BB20].



(b) An ICE plot showing the effect of age on the risk at cervical cancer for all instances in de data set [Mol20].

Figure 3.4: CP profiles versus ICE plots.

The computation of CP profiles and ICE plots is similar. It takes a specific instance and keeps the values of all features, except the feature for which we want to create the CP profile, constant [Mol20]. We are permuting the feature in question. This entails creating a number of copies of the specific instance and fill in a different value for the feature in question in every copy. Hence, the process is similar to that of computing PDP's, as discussed in section 2.4.5. However, unlike with PDPs, with CP profiles we are only permuting for the specific instance in question. Next, we apply the prediction model to these new, permuted instances to obtain their predicted values. This provides us with a list of predictions for the different values of a certain feature, e.g. *age*, that can be plotted. However, if certain features are correlated, CP and ICE can result in unrealistic plots [BB20]. The reason for this is that when we are permuting the values of a certain feature and that feature is correlated to some other feature, we can generate instances that are very unlikely. Consider, for example, 'the number of people at the park' example used to explain PDP's in section 2.4.5. To compute the CP profile for *temperature* and see its effect on the number of people at the park, we permute the values for *temperature* with the observed values for *temperature*. However, the model also includes the feature *season*, which is correlated to *temperature*, and kept constant during permutation. This results in the generation of permutations for which *temperature* = -10, but *season* = summer, which is a very unlikely instance.

3.2.2 Global Explanations

Biecek and Burzykowski refer to global explanations as dataset-level explanations [BB20]. They offer 4 methods for data-level explanations. We will only discuss three of them, namely Variable Importance (VI) plots, Partial Dependence (PDP) Profiles and Accumulated Local Effect (ALE) plots. This is because the first method discussed is not really a method that provides an explanation, but rather a set of measures aimed at comparing the performance of different models. The second global technique, the VI plot, provides an explanation by illustrating the importance of each of the input variables to the prediction model. Lastly, the third and fourth techniques, the PDP and ALE plot, show what the influence of each of the inputs is on the prediction.

1. **Variable Importance (VI)**
   The idea behind the Variable Importance (VI) plot is to show the importance of each of the input variable for the model, and rank them accordingly. VI can assist the user in a number of ways. For one, it is said to help in the model development phase by either indicating options for model simplification or assist in model validation. Model simplification can take place when certain variables have such a low importance that they might as well be fully removed from the model. Model validation takes place by comparing the variable importance with knowledge by domain experts. For example, VI might indicate high importance for a certain variable, while domain experts know from experience that this variable is not interesting to take into account, because it can not be influence or because the importance is caused by a joint effect with some other variable. There are two approaches for the generation of VI plots, a model-specific and a model-agnostic approach [BB20]. For certain types of models, a model-specific approach can be used to determine the importance of each of the variables. Consider, for example, the multi-linear regression model explained in section 2.3.2 and the corresponding equation 2.3. The weights of each of the input variable $\beta_1, ..., \beta_n$, also called the coefficients, can simply be used as the importance factor of those variables.

   As explained in the introduction of this section, we are primarily interested in model-agnostic techniques to promote the generalisability of our prototype. Therefore, we will make use of the model-agnostic VI approach. Furthermore, the model-agnostic approach has another important advantage over the model-specific approach, namely that the VI of different models can easily be compared amongst each other [BB20]. For the model-agnostic approach, the importance of a variable is determined by measuring the increase in prediction error when the effect of that variable is changed [BB20]. The greater the increase in prediction error, the more important a variable is deemed for the model [Mol20].

   There are several measurements to record the increase in prediction error. We already briefly discussed several of them in section 2.1, because as we saw they are commonly used by existing Financial Forecasting solutions to express the accuracy of a forecasting model. These measurements include Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R squared (R^2), Median Absolute Deviation (MAD) and Mean Absolute Percentage Error(MAPE). As mentioned, to measure the increase in prediction error a change in the effect of the input variable has to be simulated. This is done in a similar was as for the computation of PDP plots as explained in section 2.4.5, namely by permuting the values of that variable. More specifically, the process is as follows:

   - Determine the error rate for the current prediction model, $L(Y, X)$, where $L()$ is the chosen error measurement, Y the predictions of the original model and X are the data instances on which Y was trained.
   - Next, for every input variable j we apply the following steps to obtain its importance:
     (a) For a certain number of instances, we permute the value of j to obtain X^j. Hence, this involves randomly shuffling the values in column j for instances 0, 1, ... i.

(b) Determine the error rate for the permuted instances X^j generated in the previous step: $L(Y, X^j)$.

(c) Compute the difference between the new and the original error rate:

$$\frac{L(Y, X^j)}{L(Y, X)} \text{ or } L(Y, X^j) - L(Y, X)$$

An advantage of VI plots is that they are compact and therefore easy to interpret and follow by the users [BB20]. Furthermore, VI measurements can easily be compared amongst different models, as explained above [Mol20]. Furthermore, the calculation of VI plots by definition takes into account any interaction that is present amongst features. As discussed in section 3.2.1, if a variable interacts with some other variable, then the variable's influence on the model's output can not fully be contributed to that variable. This is because part of its influence comes from the joint effect with some other variable. This interaction should also be considered when determining the importance of a variable, because with interaction, part of a variable's importance might be contributed by the joint interaction effect and not solely by the variable in question. The permutations used to calculate the VI measurement automatically break the interaction relationships between variables, allowing us to measure the effect of interaction when compute the error rate prior to and after permutation. However, this is also a disadvantage, because if two variables interact, then the VI measurement for both of them will include any importance caused by the joint effect of those variable. This can lead to misinterpretation of the results. Lastly, VI has the disadvantage that the results may very between permutation rounds, because the permutations are determined randomly. This problem is somewhat similar to the problem illustrated in figure 3.2 where different permutation orders result in different BD plots when applied to non-additive models.

2. **Partial Dependence Profiles (PDP)**
Partial Dependence Profiles or Partial Dependence Plots (PDP) show the influence of specific variables on the model's output by expressing the output as a function of that variable [BB20]. The process behind obtaining PDP's is discussed in section 2.4.5. An advantage of PDP's is that they are said to be intuitive and therefor are both easy to implement as well as easy to explain to the user [BB20, Mol20]. Furthermore, it is interesting to note that PDP's are very similar to the CP profiles discussed in 3.2.1. With CP profiles, the effect of a certain variable is calculated for each individual instance in the data set. PDP's are essentially the average of these individual effects. This can be illustrated by comparing the PDP in Figure 3.5, with the CP profile in Figure 3.4. This also means that PDP's suffer from the same disadvantages as CP profiles, namely that they generate unrealistic permutations when certain input variables are correlated and hence generate misleading results [Mol20]. There is an extension to PDP's that solves this issue, which is discussed below.

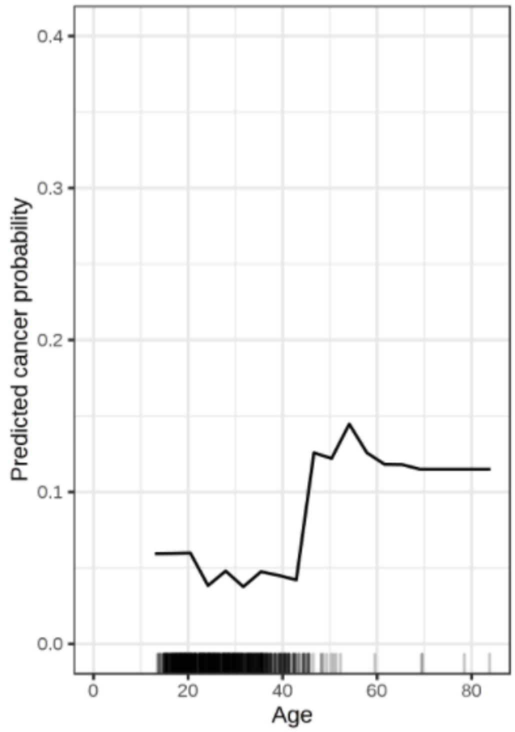


Figure 3.5: The average predicted risk at cancer as a function of the variable *age*. [Mol20].

3. **Accumulated Local Effects (ALE)**
   Similar to PDP's, the Accumulated Local Effects (ALE) plot shows how a certain input variable influences the prediction of a model on average. However, the ALE plot overcomes the issues relating to dependence that PDP's suffer from, and they are faster to compute [Mol20]. The issue arises because, as mentioned previously, PDP's average the model outputs of unrealistic data instances that are generated during permutation. ALE plots solve the issue in two steps. Firstly, they only consider the conditional distribution of the variable of interest, rather than the margin distribution. Secondly, unlike PDP's that compute the average of the model outputs for the values in the distribution, they compute the difference in model output amongst the values in the conditional distribution.

   This is illustrated by the 'people at the park' example from section 2.4.5 that predicts the number of people at the park by using, amongst others, the input variables *temperature* and *season*. We know that *temperature* and *season* are correlated, i.e. the average *temperature* increases in certain seasons and decreases in others. If we permute the values of *temperature* we obtain unlikely instances in terms of the combination of *season* and *temperature*. ALE plots overcome this issue by only permuting values within a conditional distribution rather then permuting the values of all instances in the data set. Figure 3.6 shows what such a conditional distribution might look like. In this case, we created distributions of size 5. This results in only considering instances with *temperature* values close to each other, namely within a range of 5°C degrees difference. By only permuting the *temperature* values of instances that lay close to each other in terms of *temperature*, we avoid creating instances that are unlikely.

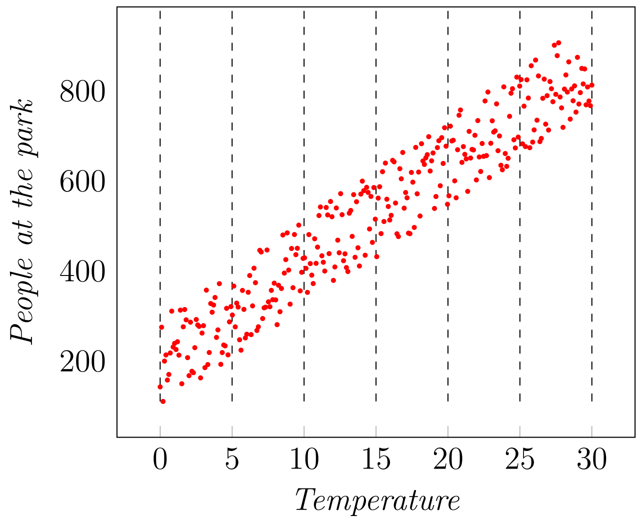


Figure 3.6: Using conditional distributions to avoid averaging unlikely data instances.

However, this does not yet fully solve the problem caused by correlated variables. If we permute the values of instances with similar *temperature* values and average the predictions for those instances, we are still not sure that the change prediction is completely caused by the *temperature* or also by *season* [Mol20]. To overcome this easy, ALE plots do not compute the average of the predictions in a distribution, but the difference. This is done as follows:

(a) The *temperature* of all instances in a distribution are replaced by the upper bound of that distribution, e.g. *temperature* = 25.

(b) The *temperature* of all instances in a distribution are replaced by the lower bound of that distribution, e.g. *temperature* = 20.

(c) Subtract the predictions obtained for the instances in (b) from the predictions obtained for the instances in (a).

By obtaining the difference in prediction between the upper and lower bound, we only obtain the effect of increasing *temperature* from 20 to 25, without incorporating potential effects of other correlated variables like *season*.

The main advantages of ALE plots is that they solve the issues that PDP's suffer from [BB20]. The computation used this issue also results in a faster computation time. This is because for ALE plots permutations are only performed within a distribution rather than for all data instances. For example, consider the conditional distribution in Figure 3.6 and assume it contains 50 instances per distribution and hence 300 instances in total. The computation for this distribution involves permuting the values for 50 instances twice, for the lower and upper bound, in each of the 6 distributions, resulting in 600 permutations in total. However, with PDP's we would not have distributions and hence permute all data instances with each of the determine grid values for *temperature*, e.g. 0, 5, 10, 15, 20, 25, 30. Hence, computing the

PDP would require $300 \times 7 = 2100$ permutations. However, ALE plots also have a number of disadvantages. Firstly, a higher number of distributions causes ALE plots to become wobbly, i.e. the plot contains many small in- and decreases. This can be avoided by using a small number of larger distributions instead. This is however not preferred because it smooths out the actual complexities of the model [Mol20]. Secondly, although the computation time of ALE plots is shorten compared to PDP's, the computation itself is more complex, as illustrated by the involved steps shown above. Lastly, if there is interaction in the prediction model, then ALE plots also will not suffice. This is because the effect of one variable can not be computed in isolation from the effect of some other feature it interacts with [BB20].

3.3 Selected XAI Techniques

Based on the discussion of applicable XAI techniques in the previous section and the design considerations states in section 3.1, we can now elaborate on the final set of XAI techniques selected for our XAI prototype. We will firstly discuss the selection of local explanation techniques, followed by the selected global techniques.

3.3.1 Local Explanations

Based on the above review of the local explanation techniques, we conclude that additive Break-down plots are best suited for our XAI prototype. The reason for this is that, as discussed in sections 2.3.2 and 2.3.3, the majority of the models used for Financial Forecasting are additive models. The other three influence methods, interactive BD plots, SHAP and LIME, are suitable for both additive models and models that contain interactions. However, they each suffer from certain limitations or drawbacks. The SHAP method, for example, requires a very long computation time to deal with interaction. As stated in the design considerations, we prefer the use of techniques with lower computation times whenever possible and hence select the additive BD plot for use in our XAI prototype. Furthermore, when applying either of these three methods to additive models, they produce the same results as the additive BD method. Therefore, incorporating one or multiple of these three methods, in addition to the additive BD plot, conflicts with our design criteria stating that the selected techniques should be complementary to each other, rather than overlapping.

Lastly, the reason to not include CP profiles is again due to our design criteria stating that the chosen XAI techniques should not contain overlapping information. As discussed in section 3.2.2, there exists a global explanation that provides the same type of information as CP profiles, namely ALE plots. However, as explained, unlike CP profiles, ALE plots do not suffer from the risk of unrealistic plots. Therefore, we exclude CP profiles from the final set of XAI techniques.

3.3.2 Global Explanations

Based on the above review of the global explanation techniques and discussed design considerations, we conclude that ALE and VI plots are best suited for our XAI solution. PDP's and ALE plots provide the same type of information, namely they both provide insights into the effect that a particular variable has on the model's prediction. To conform with our design criteria stating that selected techniques should be complementary and not overlapping, we exclude one of them from our selection. As discussed in section 3.2.2, when variables are correlated, PDP's generate biased results due to the generation of unlikely data instances. ALE plots, on the contrary, are able to deal with this correlation by filtering out the joint effect of those correlated variables. Therefore, PDP plots are excluded from our selection and use ALE plots instead.

Variable Importance plots are also selected because they provide a different type of explanation than ALE plots, and hence conform with our design criteria to select complementary XAI techniques. More specifically, VI plots provide insights into the importance of each variable for the prediction model, whereas ALE plots illustrate the influences of each variable on the prediction. VI plots express this importance in terms of its influence on the model's accuracy. As illustrated by our review of the existing Financial Forecasting solutions in section 2.3.3, accuracy related measures are an important functionality in Financial Forecasting solutions. Therefore, the selection of VI plots not only complies with the design criteria relating to the use of complementary techniques, but it is also a suitable shows for the target domain of our XAI prototype.

3.4 Logical Architecture

Based on the design considerations and specifics of the applicable XAI techniques, we selected a final set of XAI techniques to be used in the XAI Generation Module of our XAI prototype. However, before discussing the architecture and technical specifics of this module, we firstly elaborate on the logical architecture for XAI enabled financial forecasting solutions and illustrate how it is linked to the XAI Generation Module. The architectures are defined using the ArchiMate framework. The definition of the ArchiMate symbols is given in figures A.1, A.2 and A.3 in appendix A.1. Furthermore, the dashed lines indicate an access, whereas the solid lines indicate the triggering of a function, i.e. behavioural flow.

The high-level architecture for the generation of XAI enabled Financial Forecasting solutions consists of a number of components, as depicted in Figure 3.7. The blue application boxes indicate the components and the functions they contain. The green boxes indicate the data artifacts that are used by these components. The architecture not only indicates the static elements contained in the architecture, but also the behavioural flow amongst the components. This behavioural flow or process for the creation of an XAI enabled Financial Forecasting solution consists of several steps, starting at the bottom of Figure 3.7.

1. **Creating the Financial Forecasts**
   The first step involves the creation of a forecast model and the collection of the corresponding forecasts. This phase and the involved functions are indicated by the *Financial Forecast Creation* block in Figure 3.7.

   As discussed in sections 2.3.4 and 2.3.3, the creation of forecast models can be done using different types of applications, such as local programming scripts in R or Python, or by making use of analytics and forecasting capabilities in data platforms like Oracle Analytics and Workday. This means that the application in which the *Financial Forecast Creation* process takes place depends on the specific implementation. The *Financial Dataset* data artifact forms the input for the creation of the forecast model. The application that holds this dataset also differs depending on the specific implementation. It can be sources from a data platform such as the ones discussed in section 2.3.4 or simply be stored locally. Lastly, the *Financial Forecast Creation* process produces two data artifacts, namely the created forecast model and the forecasts produced by this model.

2. **Generating the XAI**
   In Financial Forecasting solutions that do not make use of XAI, the output from the previous stage can directly be fed into a dashboarding application. However, for XAI enabled solutions, we must first apply XAI techniques to the results obtained from the *Financial Forecast Creation* phase. For our XAI prototype, this is done in the *XAI Generation Module*. In the following subsections, we will elaborate on the architecture and inner workings of this Module. The architecture and code discussed there are based on the use of the R implementation of the `DALEX` package. However, it should be noted that the *XAI Generation Module* in Figure 3.7 can also be implemented using the Python implementation of `DALEX`, or even through one of the other XAI packages discussed by Arya et al [ABC+20]. The module takes as inputs the *Financial Dataset*, the *Financial Forecasting Model* and optionally the *Financial Forecasts*. The financial forecasts produced in the previous stage are optional, because they can also be obtained using the financial dataset and the forecast model. However, as we will see below, the dashboarding phase does require an input file containing the forecasts. Hence, using the already produced *Financial Forecasts* file saves computation time. The output of the *XAI Generation Module* consists of a number of .csv files that contain the data to create the XAI plots. For every XAI technique and each financial item forecasted, a plot data file is created. We will discuss this in more detail in sections 3.8 and 3.5.3.

3. **Visualising the Forecasts and XAI**
   The last step involves the visualisation of both the Financial Forecasts and the XAI plots, as well as linking the XAI plots to the corresponding financial item, forecast model and, in case of a local explanation, the specific forecast instance. The application used for this dashboarding, again depends on the specific implementation. We saw, for example, that PrecisionView™ facilitates the use of dashboarding functionalities contained in ERP systems, as well as stand-alone dashboarding applications such as Tableau, Qlik or Microsoft's Power BI. The specifics regarding visualising and linking the forecasts and corresponding XAI depends greatly on the selected Dashboarding tool. Therefore, in section 4, we will provide a more detailed overview of this process for Tableau specifically.

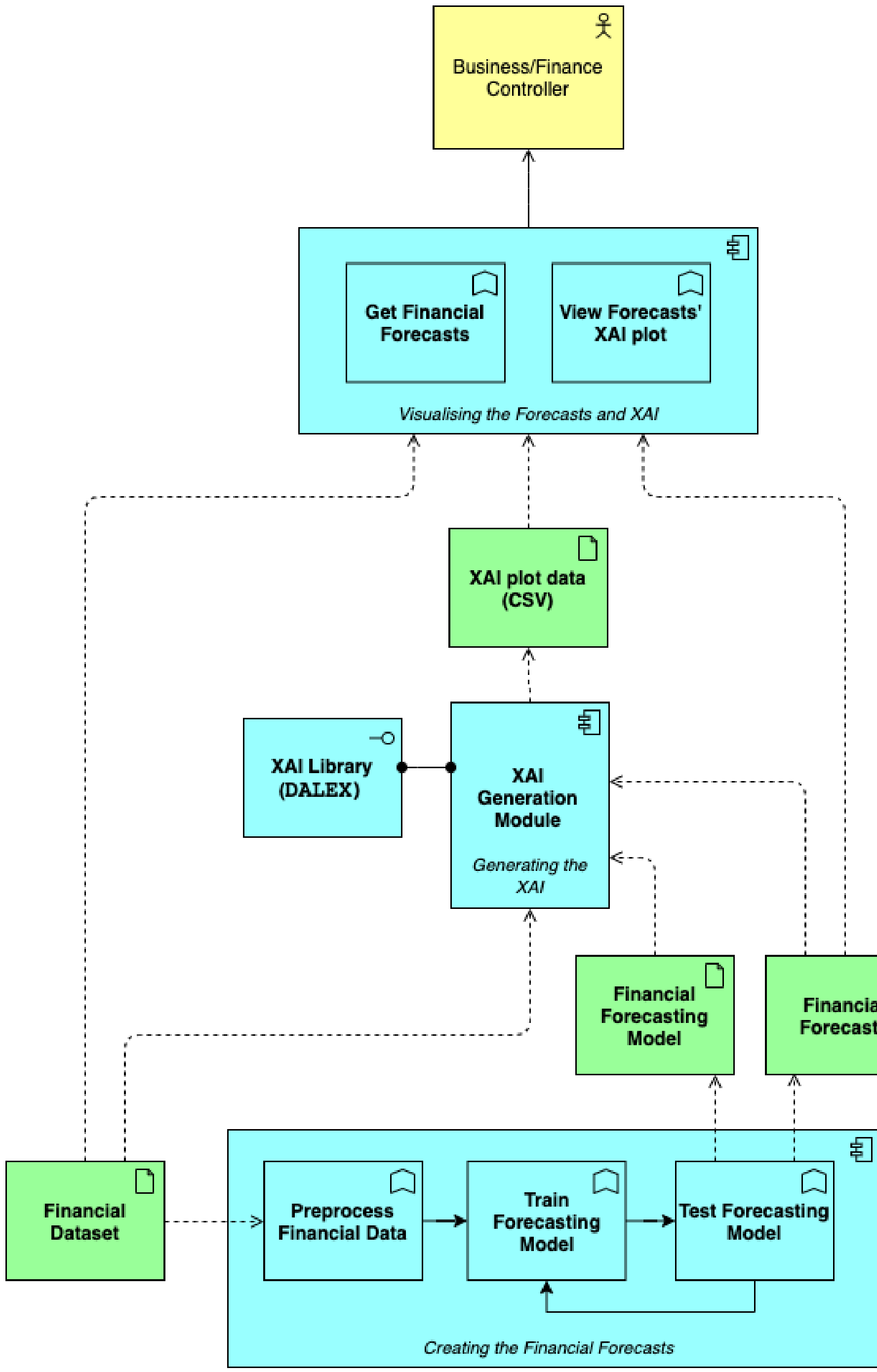


Figure 3.7: The high-level architecture behind XAI enabled Financial Forecasting solutions.

3.5 XAI Generation Module

In the previous section, we discussed the steps involved in the creation of XAI enabled Financial Forecasting solutions and illustrated how they are linked together in the architecture depicted in Figure 3.7. In this section, we will discuss the technical aspects of the XAI Generation Module contained in aforementioned architecture. The XAI Generation Module is `DALEX` specific. We will discuss the motivation behind this technical architecture decision in the discussion on XAI programming libraries. Next, we elaborate on the architecture behind the XAI Generation Module and provide an overview of how the contained functions are linked to each other. Lastly, we discuss the technical implementation aspects of these functions and illustrate the usage of the selected XAI library.

3.5.1 Library Selection

Today, several XAI libraries or toolkits exist that provide functionalities that assist in making ML models interpretable. In this research we will be making use of such a library, because it offers a number of advantages. Firstly, it facilitates the replicability of our prototype due to the reusability of programming libraries. This is because these libraries are open source and hence can be freely used by any reader looking to replicate our approach. Secondly, the use of libraries offer stability, because they are used by other developers. This means that early bugs and issues are often already discovered and fixed by others, offering more assurance that functions will work as described. Lastly, it reduces development time because the provided XAI technologies can directly be used inside the code. Arya et al. provide a comparison amongst 14 of these AI explainability libraries [ABC+20]. In their comparison they consider 6 functionalities a library can offer. These include:

- data explanation functionalities to provide summaries of datasets,
- directly interpretable modelling functionalities,
- local, post-hoc modelling functionalities,
- global, post-hoc modelling functionalities,
- self-explaining modelling functionalities,
- metrics to evaluate different explanations

In this research we are particularly interested in post-hoc, model-agnostic methods. This is because, as discussed in 3.1, these set of methods are model independent and hence enable others to replicate our approach regardless of the specific ML models they use. Furthermore, another design consideration concerned the use of complementary XAI techniques, resulting in the selection of both local and global explanation methods.

From the 14 libraries compared by Arya et al., 9 of them fulfill the requirement of offering post-hoc, model-agnostic techniques. Another distinction between the libraries reviewed concerns the programming language for which it was build. The most frequently used source languages of these libraries are R and Python. In our research, we decided to make use of the `DALEX` library developed by P. Biecek [Bie18]. The reason that we chose `DALEX` is because it is one of the 9 libraries that offers both local and global post-hoc, model-agnostic techniques. In addition to that, it is one of the few libraries that offers support for both Python and R, which facilitates the replicability of our approach for others. Furthermore,`DALEX` has successfully been used in several other studies. For example, it has been used to research the most important ingredients of Cement-stabilized rammed earth (CRSE), a sustainable construction material [ABKN20]. In this research, Anysz et al. trained several ML models using regression and derived the importance of CRSE ingredients using DALEX's Variable Importance functionality. Another example of research that made use of DALEX functionalities was conducted in the field of climate forecasting [THSK21]. More specifically, Tian et al. compared ML models to forecast reservoir inflow and used DALEX's Partial Dependence Plot functionality to investigate the contributions of the inputs in these forecasts. Furthermore, the author of DALEX recently researched the use of several DALEX functionalities for ML applications in the context of the FSI [BCG+21]. In this research, Biecek et al. illustrate the applicability of the VI, PDP, CP and BD plot functionalities for models used in Credit Scoring.

3.5.2 XAI Generation Module Architecture

In the previous subsection, we discussed how the XAI Generation Module is linked to the other steps in the process for the generation of XAI enabled Financial Forecasting solutions. Here, we will elaborate on the architecture inside the XAI Generation Module. More specifically, we will provide an overview of how each of the functions involved in applying the selected XAI techniques are linked to each other. The implementation of these functions is left out of consideration here, as it will be discussed in detail in the following section. The architecture is shown in Figure 3.5.2.

1. **Creating the Explainer**
   Prior to the generation of XAI, DALEX requires the creation of a so-called explainer. The function for the creation of this explainer takes in several arguments, as depicted in the lower left corner in Figure 3.8. Firstly, it requires both the forecast model, as well as the inputs of the train dataset. Using the input data and the model, the explainer can, in a number of cases, compute the forecasts produced by the model. The explainer's ability to do so depends on the type of model that was used to train the forecast model, as we will see below. Furthermore, optionally the outputs of the train dataset can be passed to the explainer. This is the so-called labelled data used in supervised learning, as discussed in section 2.2.1. It is primarily used for to validate models and compare their performance. The Variable Importance (VI) plot discussed in section 3.2.2 is an example of this. It uses the train inputs to create permutations and the train outputs to compute the difference in error rate prior to and after permutation [Bie21]. Lastly, both a label and a custom predict function can be specified. Both of these arguments are optional. The label represents the name of the prediction model and hence is used in all the plots that are generated by the XAI techniques. If no label is specified, it is extracted from the prediction model. Therefore, it is recommended to specify a label to ensure informative naming. The predict function is the function that is used to generate forecasts for given input. As already briefly stated above, for a number of prediction models DALEX explainers can do this on the basis of the specified model and the input dataset. For these models it can obtain the default predict() function that it contained in that model. However, for other models, DALEX can not access the contents of the default predict() function, because these models return classes [BB20]. In such a case, the explainer requires a custom predict function as input. We will discuss this in more detail in section 3.5.3.2. Although the majority of the parameters of the explain() function are not required, it is recommended by the authors to specify them as much as possible to avoid any unexpected behaviour.

2. **Applying the XAI Technique**
After the creation of the explainer object using `DALEX`'s *explain()* function, we can commence the application of XAI techniques. This phase is indicated by the *GetAdditiveBreakdown(), GetAccumulatedLocalProfile()* and *GetVariableImportance()* application functions in Figure 3.8. The functions offered by `DALEX` to obtain the BD, VI and ALE explanations are *predict_parts*(), *model_parts*() and *model_profile*(), respectively. All XAI methods require the explainer object produced in the previous step as an argument. Additional arguments differ per technique, as explained below. Furthermore, note that the data artifacts and parameters indicated as arguments for *GetAdditiveBreakdown(), GetAccumulatedLocalProfile()* and *GetVariableImportance()* in Figure 3.8, differ from the arguments for *predict_parts*(), *model_parts*() and *model_profile*() discussed below. This is due to two reasons. Firstly, *predict_parts*(), *model_parts*() and *model_profile*() are called from within *GetAdditiveBreakdown(), GetAccumulatedLocalProfile()* and *GetVariableImportance()*, respectivaly. Therefore, *GetAdditiveBreakdown(), GetAccumulatedLocalProfile()* and *GetVariableImportance()* take in arguments that are not used for *predict_parts*(), *model_parts*() or *model_profile*(), but for the generation of the explanation plots and tables. Secondly, the XAI Generation Module only uses part of the possible arguments for *predict_parts*(), *model_parts*() and *model_profile*().

   - **Break-down**
   The break-down method requires two additional arguments, namely *new_observation* and *type*. *new_observation* holds the specific forecast instance for which we want to obtain the local explanation. The *type* indicates the approach to compute the variable attribution that is used to obtain the BD plot. As discussed in section 3.2.1, there are different methods to obtain the BD plot, such as the additive and interactive BD methods, as well as SHAP. Furthermore, there are two additional arguments, namely *order* and *keep_distributions*. *order* indicates the order in which the input variables, i.e. the features, are permuted. This is relevant in the case of interaction, as discussed in 3.2.1. *keep_distributions* indicates whether the conditional distributions of the predictions should be computed. The conditional distribution provides insights into the distribution of the values of a certain feature amongst all data instances. An example is given in Figure 3.1, panel A.

   - **Variable Importance**
   The only required argument for the *model_parts*() function used to compute the VI is the explainer object. All other arguments are optional. They include the *loss_function, type, variables, variable_groups, B* and *N*. As discussed in 3.2.2, several methods, such as MSE, RMSE, R2, MAD or MAPE, can be used to compute the error rate that determines the importance of a variable. *loss_function* contains this method. It can either contain the name of a predefined loss function, such as *loss_sum_of_squares*() and *loss_root_mean_square*(), or a user defined loss function. *type* indicates the approach used to compute the variable importance, which can either be the raw error increase, difference or the ratio, as discussed in 3.2.2. Furthermore, *variables* indicates the inputs variables or features for which the importance needs to be determines.*variable_groups* offers the possibility to obtain the joint effect for each of the variables in a group. Lastly, *B* indicates the number of permutations performed in the computation of a variables importance, and **N** represents the number of data instances used in the permutation.

- **Accumulated Local Effect**
  The *model_profile*() function used to compute ALE plots as well only requires the explainer object. Other, optional arguments are *variables, N, type, variable_type, groups* and *k*. *variables* and *N* have a similar purpose as their counterparts used in the *model_parts*() function for VI. The *variables* argument indicates for which input variables or features the ALE profile is to be computed and *N* indicates the number of data instances used for sampling. *type* indicates the method used for profile computation and be either of the values *partial, conditional* or *accumulated*. The *partial* approach is used for the computation of PDP's discussed in 3.2.2. To compute ALE profiles, the required *type* is *accumulated*. *variable_type* indicates for which types of variables the ALE profiles should be computed. This can either be *nummerical*, enforcing the computation of ALE profiles only for continuous variables, or *categorical*, enforcing computation only for categorical values. Furthermore, *groups* offers the possibility to compute several ALE plots in which a distinction is made between the different values of the variable on which the grouping is performed. To return to the example of the Titanic dataset, this means we can obtain the ALE plots for the variables 'age' and 'fare', grouped by the different, possible values for 'class'. Lastly, the *k* argument indicates whether or not to cluster the CP profiles that are used to compute PDP's. If specified, it indicates the number of clusters to be used in clustering. The argument is however not used for the computation of ALE profiles.

3. **Obtaining the XAI Plot and Table**
   Once the *predict_parts*(), *model_parts*() and *model_profile*() for the BD, VI and ALE methods respectively, are obtained, the plots can be created. This is done by applying R's *plot()* function that takes as an argument the object returned by the functions *predict_parts*(), *model_parts*() and *model_profile*(). However, to create XAI enabled Financial Forecasting solutions, in which the forecasts are linked to the XAI explanations using a dashboarding tool, the XAI Generation Module has to return more than only images that contain the XAI plots. In fact, these plot images are not used for the dashboarding, but only for verification purposes. Instead, the dashboarding phase requires files that contain the data used to create these plots to be able to rebuild the plots inside the dashboard tool, as well as to link them to the corresponding financial item and forecast instance. To do so, the XAI Generation Module also contains the *createExplanationTable* function, in addition to the *createExplanationPlot* function. Both functions were defined in such a way that they can be called by all three XAI functions, as depicted in the lower right corner in Figure 3.8.

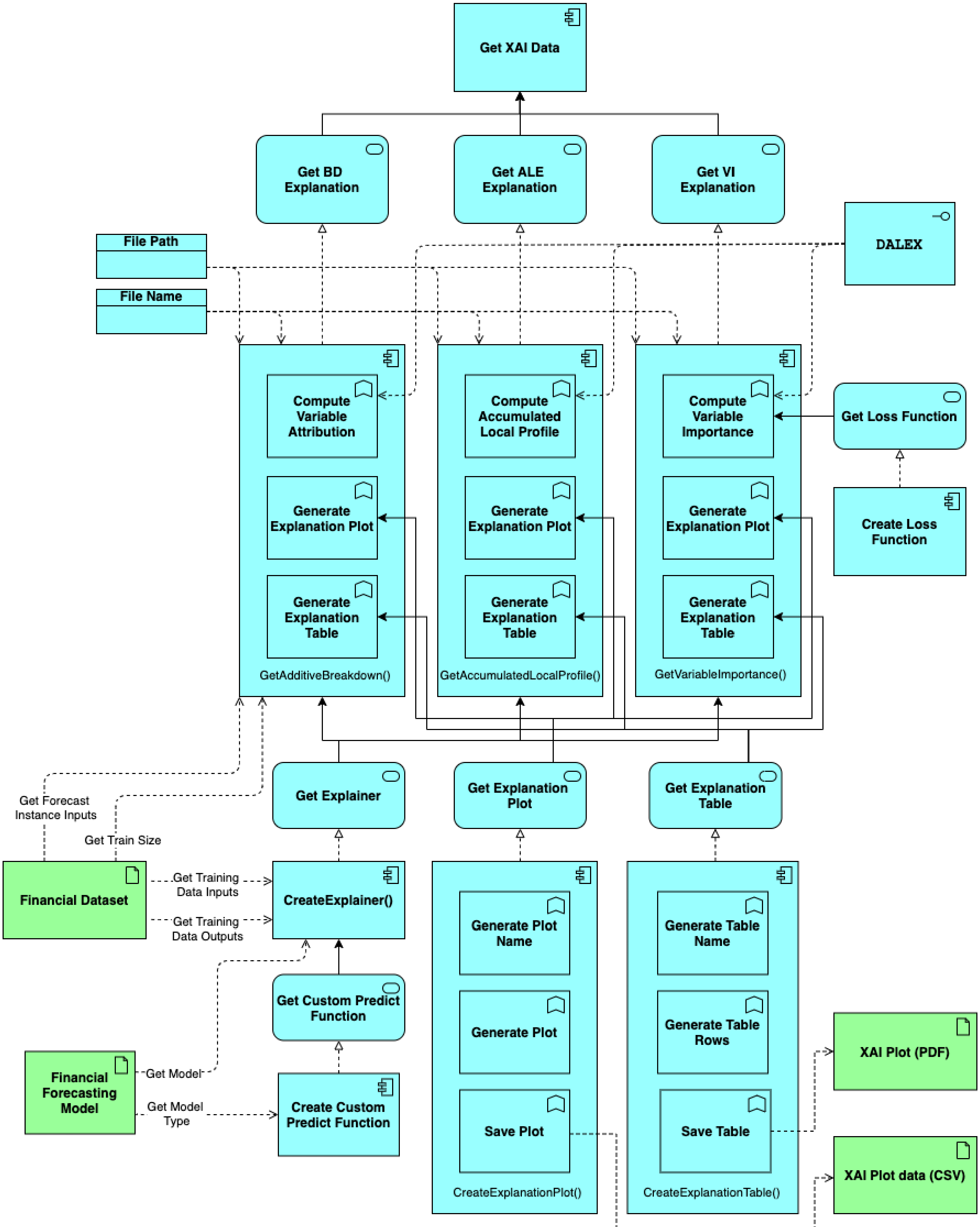


Figure 3.8: The architecture of the XAI Generation Module for the XAI prototype.

3.5.3 Usage of the XAI Generation Module

In the previous sections we discussed the selected XAI techniques and the high-level and Module specific architectures. In the discussion of the architecture behind the XAI Generation Module, we elaborated on the functions involved to obtain the XAI output for the dashboarding phase, the arguments these functions take, and the flow amongst the functions. In this section, we will go into the specific implementation of these functions for the Financial Forecast models used in the XAI prototype developed in this research. This is done through means of code snippets from the developed XAI Generation module. Furthermore, we will illustrate the outputs of the functions through means of the obtained plots produced on a test dataset. To understand these plots, we first briefly elaborate on this test dataset and the forecast models for which the XAI techniques are applied.

3.5.3.1 Test Dataset and Model Development

In order to develop and test the functionalities employed by XAI Generation Module, firstly forecast models were created. These forecast models and the input data on which they are trained can then be used as the input for the XAI Generation Module. The test dataset that was used is obtained from the UCI Machine Learning Repository. The dataset contains information on the daily rental of bikes in a bike sharing system and the accompanying weather data between 2011 to 2012. More precisely, for every data instance the following features are provided:

- *season*, the season in which an instance was recorded, i.e. winter, spring, summer or fall.
- *holiday*, whether the instance was recorded during a holiday period or not.
- *workingday*, represents whether this day was a working day or not.
- *weathersit*, the weathersituation on the day the instance was recorded. This can either be "Good", "Misty" or "Rain/Snow/Storm".
- *atemp*, the feeling temperature on the day of recording.
- *casual, registered* and *cnt*, the number of casual users, registered users and total users, respectively, renting bikes on the day of recording.

Furthermore, the date (*dteday)*, month (*mnth*), year (*yr*), day of the week (*weekday*), windspeed (*windspeed*) and the temperature (*temp*) on which the instance was recorded are provided.

Using this dataset, three different forecast models where created using the multi-linear regression, ARIMAX and Prophet with Regressors models. These models where chosen for two reasons. Firstly, because they are used by PrecisionView™, the Financial Forecasting solutions on which we will implement our XAI prototype, as discussed in 2.3.4. Secondly, they still provide a good basis for the exploration of XAI techniques for Financial Forecast, as these models are commonly used for Financial Forecasting in general, as discussed in 2.3.2. It should be noted that we decided not to include ARIMA and Prophet models in the development of our prototype. The reason for this is, as explained in 2.3.2, that these models are non-dynamic models. This means they only use a single input to train the model, namely the historical value of the item it is trying to predict. However, as we saw in 3.2, both the local and global XAI methods all provide an explanation by means of the input variables or features of the model. Hence, as ARIMA and Prophet do not incorporate such features in the model, applying these XAI techniques is not possible.

3.5.3.2 Explainers

As explained in 3.5.2, the application of `DALEX`'s XAI techniques, firstly requires the generation of the explainer object. As discussed earlier, the *explain()* function's ability to obtain the forecasts using the specified model and input dataset, depends on the type of forecast model used. In the case of a multi-linear regression model created using the *lm()* function from the `stats` package, `DALEX` knows how to obtain the forecasts. However, for both the ARIMAX and Prophet with Regressors models, created using the *auto.arima()* and *prophet()* functions from the `forecast` and `prophet` package respectively, `DALEX` cannot obtain the predictions. Therefore, both ARIMAx and Prophet with Regressors require a custom predict function to work with `DALEX`. Therefore, a wrapper function is used to determine the need for passing a custom predict function, prior to calling `DALEX`'s *explain()* function, as shown in listing 3.1.

```
createExplainer <- function(myModel, inputs, outputs, myLabel,
    myPredictionFunction = NULL){

    if(length(myPredictionFunction)){
        myExplainer <- DALEX::explain(model = myModel, data = inputs, y =
    outputs, label = myLabel, predict_function = myPredictionFunction)
    }
    else{
        myExplainer <- DALEX::explain(model = myModel, data = inputs, y =
    outputs, label = myLabel)
    }

    return(myExplainer)
}
```

Listing 3.1: *createExplainer()* wrapper function used to call `DALEX`'s *explain()* function using the correct predict function.

The custom predict function for ARIMAX models is shown in listing 3.2. It uses the *predict()* function from the `stats` package. The `stats` package recognizes that the model was trained using *arima()* and therefore automatically uses the *predict.Arima()* function in the background. The function requires the ARIMAX model (*object*), the number of future items we want to forecast (*n.ahead*) and the values for the features of the future instances to be predicted (*newxreg*).

```
myPredictionFunction <- function(myModel, newdata)  {
    indices <- as.numeric(rownames(newdata))
    predictions <- vector()

    for(i in 1:length(indices)){
        result <- predict(object = myModel, n.ahead = indices[i], newxreg =
    newdata[i,])$pred
```

```
        predictions[i] <- result[indices[i]]
    }

    return(predictions)
}
```

Listing 3.2: The custom predict function used to obtain the forecasts for an ARIMAX model.

The custom predict function for Prophet with regressors models is shown in listing 3.3. It uses the *predict()* function from the same package used for the model creation, namely the `prophet` package. The function requires the Prophet model (*myModel*) and the regressor values of the future instances to be predicted , i.e. the values for the features (*newdata*). Due to the structure of the *newdata* object containing the regressors, the regressor values need to be stored in a dataframe *inputOrder*, in order for the obtained forecasts to be stored in *predictions* in the correct, chronological order.

```
myPredictionFunction <- function(myModel, newdata)  {
    inputOrder <- data.frame(rowNr = as.numeric(rownames(newdata)), idx =
   1:length(newdata[,1:1]))

    inputOrder <- inputOrder[order(inputOrder$rowNr),]
    results <- (predict(myModel, newdata))$yhat
    predictions <- vector()

    for(i in 1:length(inputOrder[,1:1])){
        predictions[i] <- results[inputOrder[i,]$idx]
    }
    return(predictions)
}
```

Listing 3.3: The custom predict function used to obtain the forecasts for a Prophet with Regressors model.

3.5.3.3 Plot and Table Creation

Before discussing the implementation details of the selected XAI techniques, we will first elaborate on the function used for the generation of the explanation plot and table, discussed in 3.8. The reason for this is that the XAI techniques are obtained using wrapper functions, as we will see below. This is done to enable the XAI Generation Module to call *createExplanationPlot* and *createExplanationData* from within the wrapper for the XAI technique in question. By doing so, *createExplanationPlot* and *createExplanationData* know for which XAI technique they have to create the explanation plot and data. This is necessary because, as shown in listings 3.4 and 3.5, these functions apply a different procedure depending on the XAI technique in question. For the *createExplanationPlot* function, the difference in procedure depending on the XAI technique is minimal. The only difference is that BD plots require different naming, because a plot is created for every future forecast instance, whereas the two global explanations generate only one plot for the model as a whole. To this end, *createExplanationPlot* takes in the *dateNr* as argument to differentiate between the names of the generated BD plots, as shown in listing 3.4, line 4 and 7.

```
1 createExplanationPlot <- function(explanation, filePath, fileName, title,
    dateNr = NULL){
2
3     if(length(dateNr)){
4         graphName <- paste0(filePath, fileName, "_", dateNr, "-table.pdf")
5     }
6     else{
7         graphName <- paste0(filePath, fileName , "-table.pdf")
8     }
9
10    pdf(file = graphName)
11    plt <- plot(explanation) + ggtitle(title, subtitle = NULL)
12    dev.off()
13 }
```

Listing 3.4: The function used to create the resulting visual for the XAI explanation in question.

As is to be expected, the difference in procedure amongst the XAI techniques is much greater for *createExplanationTable*. The reason for this is that the three XAI techniques generate very different plots and hence the data required to reproduce these plots differs greatly. The generation of the table for BD plots is completely distinct from the two global explanations, as can be seen in listing 3.5, lines 6:37. The procedure to obtain the explanation table for the ALE and VI plots, found on lines 39:58 in listing 3.5, does contain some overlap. Furthermore, it is important to note, for readers that wish to replicate the XAI prototype, that certain operations in *createExplanationTable* are specific to the dataset used. The date column for the rows in the BD explanation of a specific forecast is obtained based on the start date of the dataset(line 12, listing 3.5). Furthermore, lines 20:29 and 48:54 are only applicable if renaming of the model's features is desired. This might be the case if the original feature names in the dataset are deemed not user-friendly.

```
1 createExplanationTable <- function(explanation, filePath, fileName, XAIType,
    dateNr, trainSize, originalVars = NULL){
2
3     tableName <- paste0(filePath, fileName, "-table.csv")
4
5     # Create the explanation table for BD explanations.
6     if(XAIType == "BD"){
7         # Round to integer
8         explanation$contribution = round(explanation$contribution, digits = 0);
9         explanation$cumulative = round(explanation$cumulative, digits = 0);
10
11        # Add date of the forecast that the current explanation belongs to.
12        startDate <- ymd(as.Date("01/01/2011", format = "%d/%m/%Y")) %m+%
    months(trainSize)
13        date <- ymd(startDate) %m+% months(dateNr)
14        explanation["date"] <- format(date, format = "%d/%m/%Y")
15
16        explanation$contribution[1] <- 0
17        explanation$contribution[nrow(explanation)] <- 0
```

```
        explanation$contribution[nrow(explanation)] <- sum(explanation$contribution)

        if(length(originalVars)){
            for(i in 1:nrow(explanation)){
                if(!is.na(originalVars[explanation[i, 3]])){

                    explanation[i, 1] <- stringr::str_replace(explanation[i, 1], explanation[i, 3], originalVars[explanation[i, 3]])

                    explanation[i, 3] <- originalVars[explanation[i, 3]]
                }
            }
        }

        if(dateNr > 0){
            write.table(explanation, tableName, sep = ",", row.names = FALSE, col.names = !file.exists(tableName), append = T)
        }
        else{
            write.csv(explanation, tableName, row.names = FALSE)
        }
    }
    # Create the explanation table for VI and ALE explanations.
    else{
        if(XAIType == "ALE"){
            variable <- explanation$agr_profiles[,1]
            label <- explanation$agr_profiles[,2]
            x <- explanation$agr_profiles[,3]
            yhat <- round(explanation$agr_profiles[,4], digits = 0);
            ids <- explanation$agr_profiles[,5]
            explanation <- data.frame(variable, label, x, yhat, ids)
        }

        if(length(originalVars)){
            for(i in 1:nrow(explanation)){
                if(!is.na(originalVars[explanation[i, 1]])){
                    explanation[i, 1] <- originalVars[explanation[i, 1]]
                }
            }
        }

        write.csv(explanation, tableName, row.names = FALSE)
    }
}
```

Listing 3.5: The function used to create the table containing the data to create the plot for the XAI explanation in question.

3.5.3.4 Break-down Plots

As explained above, for the application of `DALEX`'s *variable_attribution* function and generation of the corresponding explanation plot and data, the XAI Generation Module uses a wrapper function, as shown in listing 3.6. The arguments specific to the wrapper function *getAdditiveBreakdown()* are *newInstance, dateNr* and *trainSize*.

- *newInstance* is a dataframe consisting of a single row that holds the feature values for a specific future forecast instance.
- *dateNr* is used in *createExplanationPlot()* for file naming, and in *createExplanationTable()* for determination of the date, as well as to determine whether to create a new table or update an existing one.
- *trainSize* is used to determine the start date of the forecast instances in the BD plot, based on the start date of the training dataset and its size in terms of the number of instances it contains.

```
getAdditiveBreakdown <- function(myExplainer, newInstance, filePath, fileName,
   dateNr, trainSize, originalVars = NULL){

    additiveBreakdownPlot <- DALEX::variable_attribution(myExplainer,
   new_observation = newInstance, type = "break_down")

    createExplanationPlot(additiveBreakdownPlot, filePath, fileName,
   paste("Breakdown plot for", myExplainer$label), dateNr)

    createExplanationTable(additiveBreakdownPlot, filePath, fileName, "BD",
   dateNr, trainSize, originalVars)
}
```

Listing 3.6: The wrapper function *getAdditiveBreakdown()*, used to apply `DALEX`'s *variable_attribution* function and obtain the corresponding explanation plots and table.

The BD plot obtained after applying the *getAdditiveBreakdown()* wrapper to a multi-linear regression model, trained on the bike rental dataset discussed in 3.5.3.1, is depicted in Figure 3.9. It can be seen that the BD plot starts at an average predicted daily bike rentals of ~ 4504. *weathersit* has the biggest impact on the final prediction of ~ 3771 bike rentals, by lowering the prediction with ~ 1719. *days_since_2011* and *temp* have the second and third largest impact on the prediction, by adding ~ 1158 and ~ 1076 to the predicted bike rentals, respectively. After having taken into account the contributions of the remaining drivers, we end up at a final prediction of ~ 3771 daily bike rentals. Furthermore, for this specific forecast instance it can be seen that all drivers together had a negative contribution of 731.

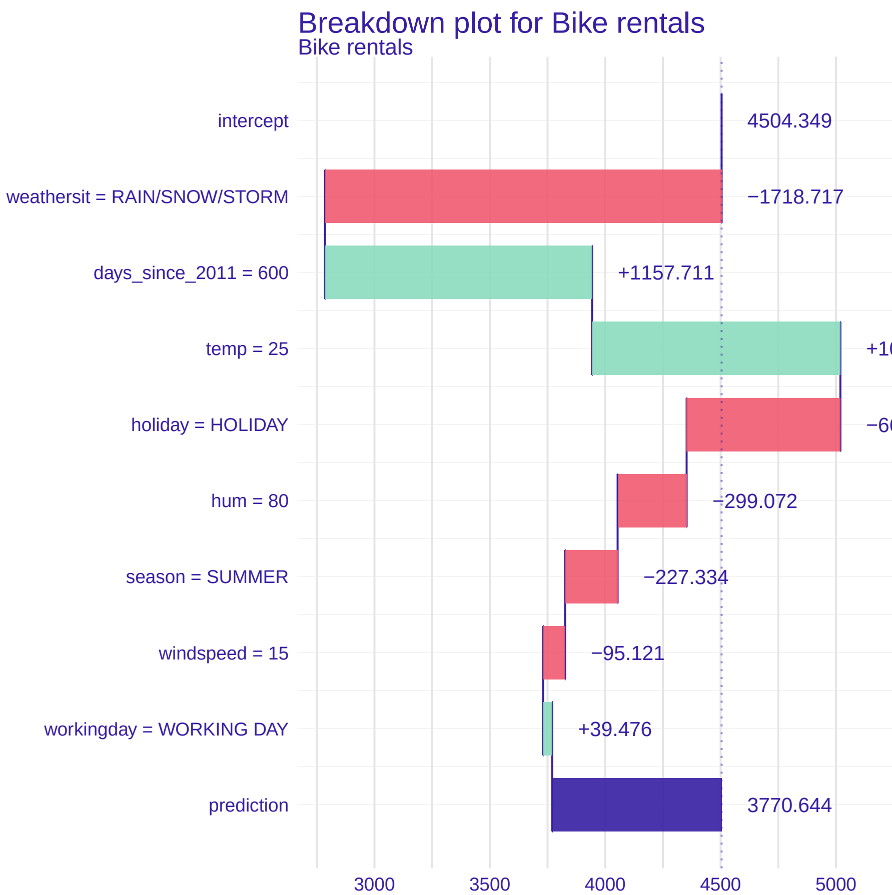


Figure 3.9: The resulting BD plot visualisation obtained by applying `DALEX`'s *variable_attribution* function to the bike rental dataset.

3.5.3.5 Variable Importance Plots

The generation of the Variable Importance plot uses a wrapper function as well, as is shown in listing 3.7, *getVariableImportance()*, for the same reasons as discussed in 3.5.3.4. The only argument specific to the wrapper function *getVariableImportance()* is *myLossFunction*. As explained in sections 3.2.2 and 3.5.2, the loss function is used to compute the error rate before and after permuting the values of a certain feature, in order to determine the importance of that feature. In the XAI Generation Module, we defined a custom loss function, namely the MAPE, rather than using one of the predefined methods. The reason for doing so is that the Financial Forecasting solution on which our XAI prototype is to be implemented, PrecisionView™, also uses the MAPE as the main measure of accuracy. We expect consistency in the use of error measurements to facilitate the learning process for the users of the XAI enabled Financial Forecasting solution, as it requires the understanding of only a single error measurement. The method containing the custom loss function is depicted in listing 3.8.

```
getVariableImportance <- function(myExplainer, myLossFunction = NULL,
   filePath, fileName, originalVars = NULL){
```

```

    if(length(myLossFunction)){
        VIP <- DALEX::model_parts(myExplainer, loss_function = myLossFunction, type = "difference")
    }
    else{
        VIP <- DALEX::model_parts(myExplainer, loss_function = loss_root_mean_square, type = "difference")
    }

    createExplanationPlot(VIP, filePath, fileName, paste("Variable Importance plot for", myExplainer$label))
    createExplanationTable(VIP, filePath, fileName, "VI", NULL, NULL, originalVars)
}
```

Listing 3.7: The wrapper function *getVariableImportance()*, used to apply `DALEX`'s *model_parts* function and obtain the corresponding explanation plots and table.

```
myMAPEfunction <- function (y_true, y_pred){
    MAPE <- mean(abs((y_true - y_pred) / y_true)) * 100
    return(MAPE)
}
```

Listing 3.8: The custom loss function passed to `DALEX`'s *model_parts* function to obtain the error rates prior to and after permutation.

The resulting visualisation obtained after applying the wrapper function *getvariableImportance()* to a multi-linear regression model for the prediction of daily bike rentals, is shown in Figure 3.10. The bars indicate the percentage increase in error rate, measured by the MAPE function defined in listing 3.8. From the VI plot, it can be concluded that *temp* is the most important variable for the bike rental forecast model, followed by *weathersit, days_since_*2011*, hum* and so on. Furthermore, the box plots in each of the bars indicate the distribution of variable importance amongst each of the permutations. The box plots provide information on how representative the importance is for the sample of permutations, i.e. whether the importance of the individual permutations all lay close to the average importance.

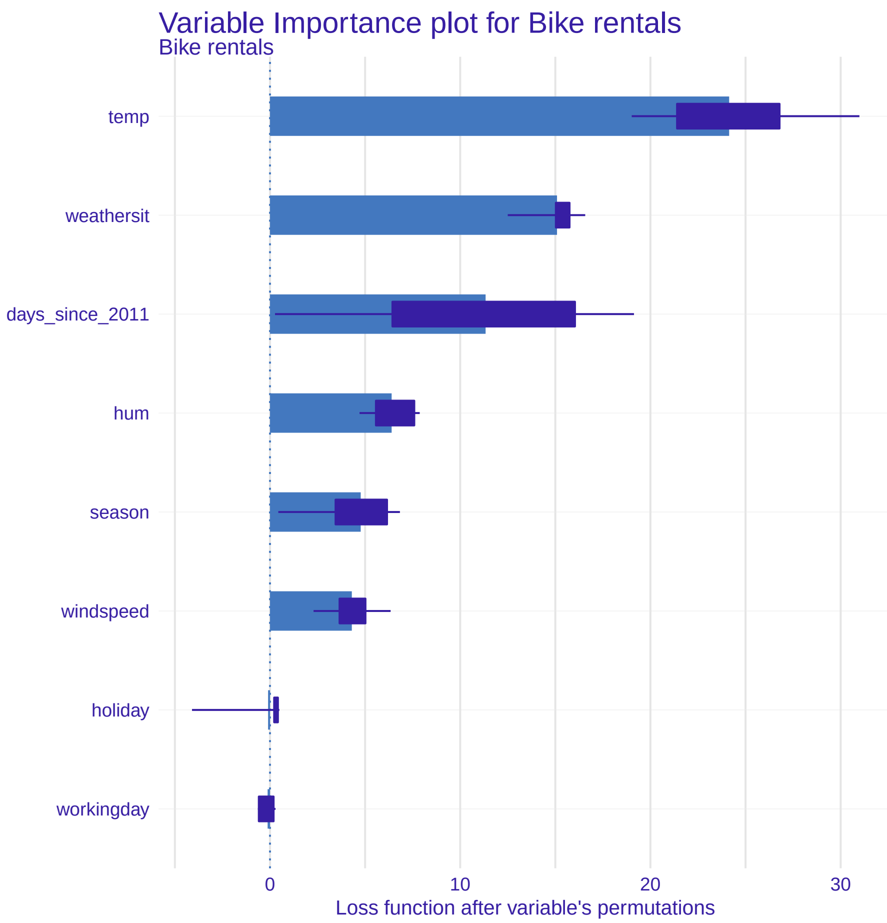


Figure 3.10: The resulting VI plot visualisation obtained by applying `DALEX`'s *model_parts* function to the bike rental dataset.

3.5.3.6 Accumulated Local Effects Plots

The wrapper function used to apply `DALEX`'s *model_profile* function and obtain the corresponding explanation plot and table, *getAccumulatedLocalProfile()*, is given in listing 3.9. The only argument specific for this wrapper function is *myVariables*. It is passed to the *model_profile* function to indicate for which variables the Accumulated Local Effect has to be obtained.

```
getAccumulatedLocalProfile <- function(myExplainer, myVariables, filePath,
    fileName, originalVars = NULL){

     ALE <- DALEX::model_profile(myExplainer, variables = myVariables, type =
    "accumulated")
     createExplanationPlot(ALE, filePath, fileName, paste("ALE plot for",
    myExplainer$label))
     createExplanationTable(ALE, filePath, fileName, "ALE", NULL, NULL,
    originalVars)
}
```

Listing 3.9: The wrapper function *getAccumulatedLocalProfile()*, used to apply `DALEX`'s *model_profile* function and obtain the corresponding explanation plots and table.

Applying the $model\_profile$ and *createExplanationPlot()* functions using the wrapper function *getAccumulatedLocalProfile()* on a multi-linear model for the prediction of daily bike rentals, resulted in the ALE plot shown in Figure 3.11. From the ALE plot, it can be concluded that both *days_since_*2011 and *temp* are positively correlated with the average predicted daily bike rentals, i.e. if the value of either of those variables increases, then so does the average predicted bike rentals. *hum* and *windspeed* on the contrary, are negatively correlated with the average predicted daily bike rentals. Furthermore, from the plots, it can also be found what the amount of predicted daily bike rentals will be for specific values of each of the variables *days_since_*2011*, hum, temp* and *windspeed*. It should be noted that the ALE plot contains less variables than the BD and VI plot. The reason for this is that, as discussed in 3.5.2, `DALEX`'s $model\_profile$ function by default only computes the ALE profiles for the numeric variables of a model. As we will see in section 4, the Financial Forecasting solution on which our XAI prototype is to be implemented, namely PrecisionView™, only contains models with numeric variables. Therefore, in the setup of our approach and the development of the XAI Generation Module, we also focused on the generation of ALE plots for numerical values only. However, for readers wishing to replicate our approach, this can easily be changed through the use of *getAccumulatedLocalProfile()*'s *myVariables* argument.

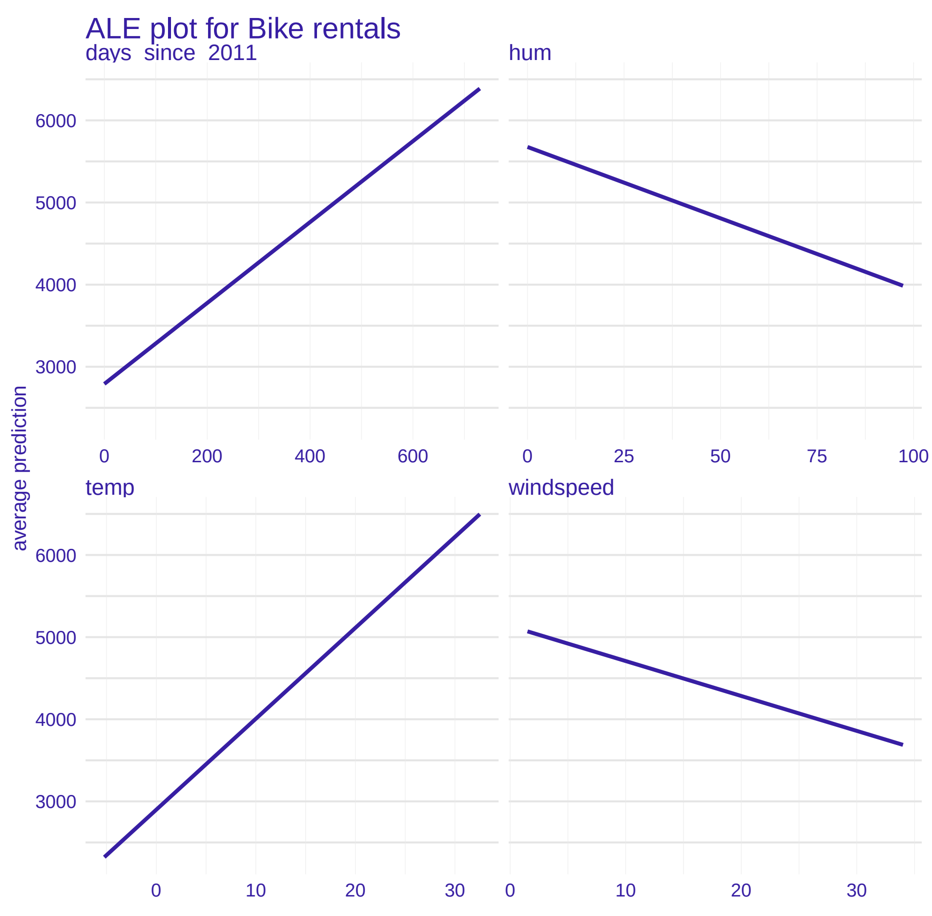


Figure 3.11: The resulting ALE plot visualisation obtained by applying `DALEX`'s $model\_profile$ function to the bike rental dataset.

3.5.3.7 Putting It All Together

In the previous sections we saw the implementation details for each of the XAI techniques incorporated in the prototype, as well as the helper functions they use. Here we will show the main function, *applyXAI*, where everything is put together and which arranges the orchestration amongst these functions. It is shown in listing 3.10. Firstly, it defines the inputs belonging to the training set and the inputs based on which the forecasts were generated. Secondly, it calls the *getTidyNames()* function. This function determines where the XAI data should be stored, how the corresponding files should be named, and generates a user-friendly label based on the current forecast model, model type and variable to be forecasted. Next, the explainer object for local explanations is created and the BD explanation is generated for every forecast instance. Lastly, the explainer object for global explanation is created, and the VI and ALE explanations are created. The reason for creating a separate explainer object for global explanations is because, as explained in section 3.2.2, to compute the Variable Importance, we need to know both the actual and observed values of the dependent variable to measure the change in prediction error. Therefore, we need a separate `DALEX` explainer that uses the train inputs and outputs instead of the forecast inputs and outputs, because for the forecast data we obviously do not have the actuals available.

```
applyXAI <- function(inputpath, inputfile, forecastModel, modelType, inputs,
    trainOutputs, forecastOutputs, myPredictionFunction, t, f, independentVar,
    originalVars = NULL) {

    trainInputs <- as.data.frame(inputs[1:t,])
    forecastInputs <- as.data.frame(inputs[(t+1):(t+f),])

    # 1. get the label, filename and -path for the current forecast model.
    tidyNames <- getTidyNames(inputpath, inputfile, forecastModel, modelType,
    independentVar)

    myLabel <- tidyNames$myLabel
    myFilename <- tidyNames$myFilename
    myFilepath <- tidyNames$myFilepath

    # 2. create the explainer for the forecasted data
    myExplainer <- createExplainer(forecastModel, forecastInputs,
    forecastOutputs, myLabel, myPredictionFunction)

    #3. loop through all forecasted values and get the corresponding breakdown
    plot.
    for(i in 1:nrow(forecastInputs)){
        newInstance <- forecastInputs[i,]
        getAdditiveBreakdown(myExplainer, newInstance, myFilepath,
    paste(myFilename, "additive-breakdown", i, sep = "_"), dateNr = i-1, t,
    originalVars)
    }

    # 4. create the explainer for the training data
    myExplainer <- createExplainer(forecastModel, trainInputs, trainOutputs,
    myLabel, myPredictionFunction)

```

```
    # 5. Define the custom loss function (MAPE in this case) and get the VI plot.
    myMAPEfunction <- function (y_true, y_pred){
        MAPE <- mean(abs((y_true - y_pred) / y_true)) * 100
        return(MAPE)
    }

    getVariableImportance(myExplainer, myMAPEfunction, myFilepath, paste(myFilename, "VI", sep = "_"), originalVars)

    # 6. get the ALE plot
    getAccumulatedLocalProfile(myExplainer, getDependentVariables(forecastModel), myFilepath, paste(myFilename, "ALE", sep = "_"), originalVars)
}
```
Listing 3.10: The main function, *applyXAI*, from where all the other functions discussed in section 3.5.3 are executed.

4 System Implementation

In the previous section, we discussed the logical and technical aspects of our XAI prototype. We elaborated on the design considerations, the selected XAI techniques and the logical architecture. Furthermore, we discussed the technical implementation details of the XAI Generation model, including the library selection and the architecture and usage of the module. In this section, we will discuss the process of implementing the XAI prototype for PrecisionView™. To this end, we will firstly discuss the software stack of the PrecisionView™ version used in this research and the corresponding architecture. Next, we will discuss the modifications made to PrecisionView™'s Tableau dashboard. As explained in section 2.3.4, Tableau is a dashboarding solution used by PrecisionView™ to visualise the final forecasts and their corresponding explanations. A modified version of this dashboard is required in order to support the conduction of our experiment, as we will explain in section 4.2. This Financial Forecasting dashboard without XAI forms the base case with which we compared the Financial Forecasting dashboard with XAI, in order to measure the increase in trust, understanding and aspects relating to the insights gained, as stated in section 1.3. Next, we will discuss the process of applying the XAI Generation Module to PrecisionView™'s forecasting models, the resulting outputs and how to incorporate them in the dashboarding tool. Lastly, we will showcase the resulting XAI enabled Financial Forecasting solution for PrecisionView™ and elaborate on the design improvements made in collaboration with professionals from Deloitte.

4.1 PrecisionView™ Architecture

As discussed in section 2.3.4, PrecisionView™ is an "agnostic framework", meaning that each of the four components in the software stack can be implemented using different software solutions. The software stack used in this research is the so-called prototype solution of PrecisionView™. The first component in this stack, the data platform component, simply consists of locally stored excel files that contain the financial data. The analytics part where the driver selection, as well as the forecast model training and testing takes place, consists of several R scripts. The optional Enterprise Performance Management component is not used in the prototype solution. Lastly, the dashboarding component, in which the forecasts are visualised, is implemented using Tableau. Tableau is a software solution used for business intelligence purposes. It provides data analytics capabilities to explore and manage data through the use of several visual functionalities. A simplified version of the resulting architecture of the prototype solution is shown in figure 4.1. The .xlsm file containing the aggregated model for all segments form the input for the visualisation of the forecasts in Tableau.

As discussed in section 2.3.4, each of the sheets in PrecisionView™'s Financial Forecasting solution offer the possibility to filter the forecasts based on a companies' segments and subsegments. These segments and subsegments are used to group the data based on certain properties. For example, in the dataset used for PrecisionView™'s prototype solution there are two geographical segments, North America and International, that group line item forecasts based on their geographical location. The subsegments belonging to these two geographical segments further divide line item forecasts according to the specific country and continent, respectively. In order to do so, PrecisionView™ generates the forecasts for each Financial line item per segment and subsegment. A detailed overview of the architecture containing the specific segments and subsegments used, is shown in figure 4.2. Within the folder for each segment, N R scripts exist that generate the forecast model for each of the subsegments that fall under the current segment. Here, N is the number of subsegments within the current segment. The input for these R scripts consists of the financial dataset for each of the subsegments contained in the corresponding folder, i.e. $segment/02\_segmentinput$. After applying the R scripts, the resulting forecasts are placed in the corresponding output folder, $segment/04\_segmentOutput$. This folder contains a .csv file with the forecasts for a specific subsegment and financial line item, obtained by applying one of the forecast models multi-linear regression, ARIMA, ARIMAX, Prophet or Prophet with regressors. next, using a macro enabled worksheet, these output files containing the forecasts are loaded into different sheets in the corresponding excel model, i.e. $segment/01\_segmentModel/DriverBasedModelSegment.xlsm$. Finally, through the use of data links, the models for each individual segment are loaded into the overall final model $FinalModels/DriverBasedModelFinal.xlsm$, where the forecasted values of each segment are added to obtain the total forecasted values.

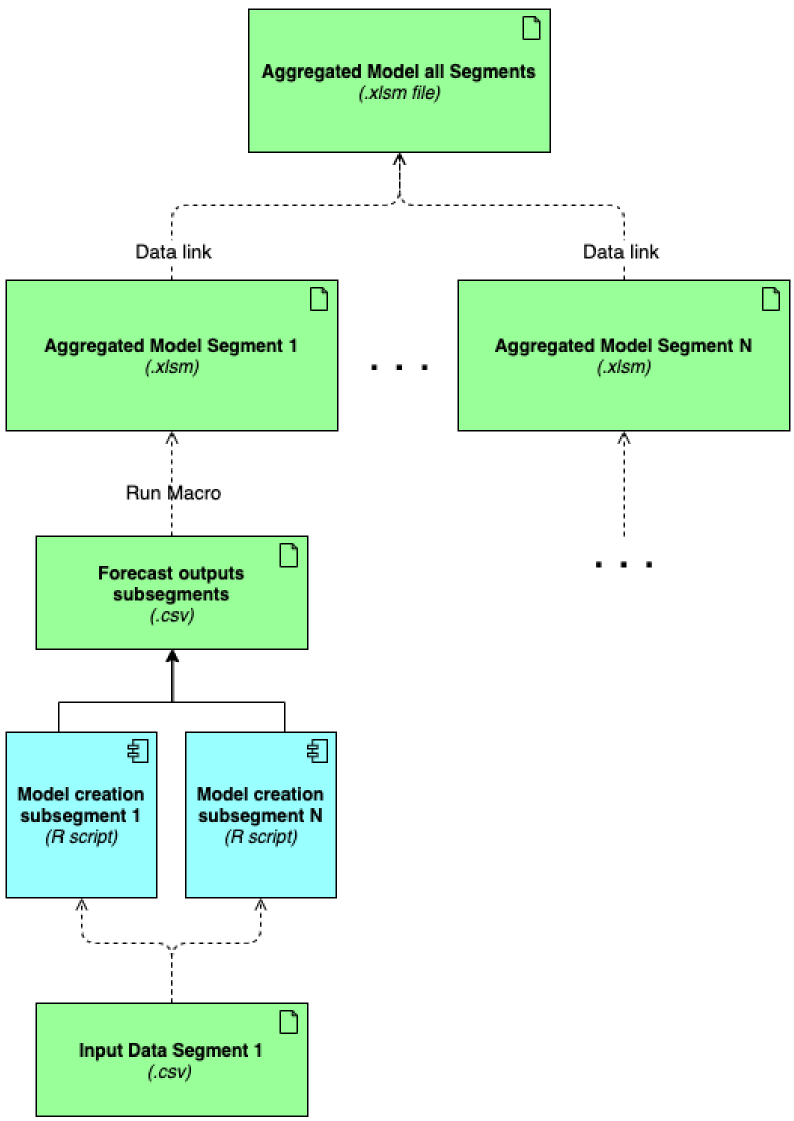


Figure 4.1: Compressed overview of the architecture of PrecisionView™'s prototype solution.

The architecture described in this section forms the basis for the implementation details involved in applying our XAI prototype to PrecisionView™. Firstly, it determines the decisions made in rebuilding the Financial Forecasting dashboard, as described in section 4.2. Secondly, it influences the implementation process described in section 4.3.

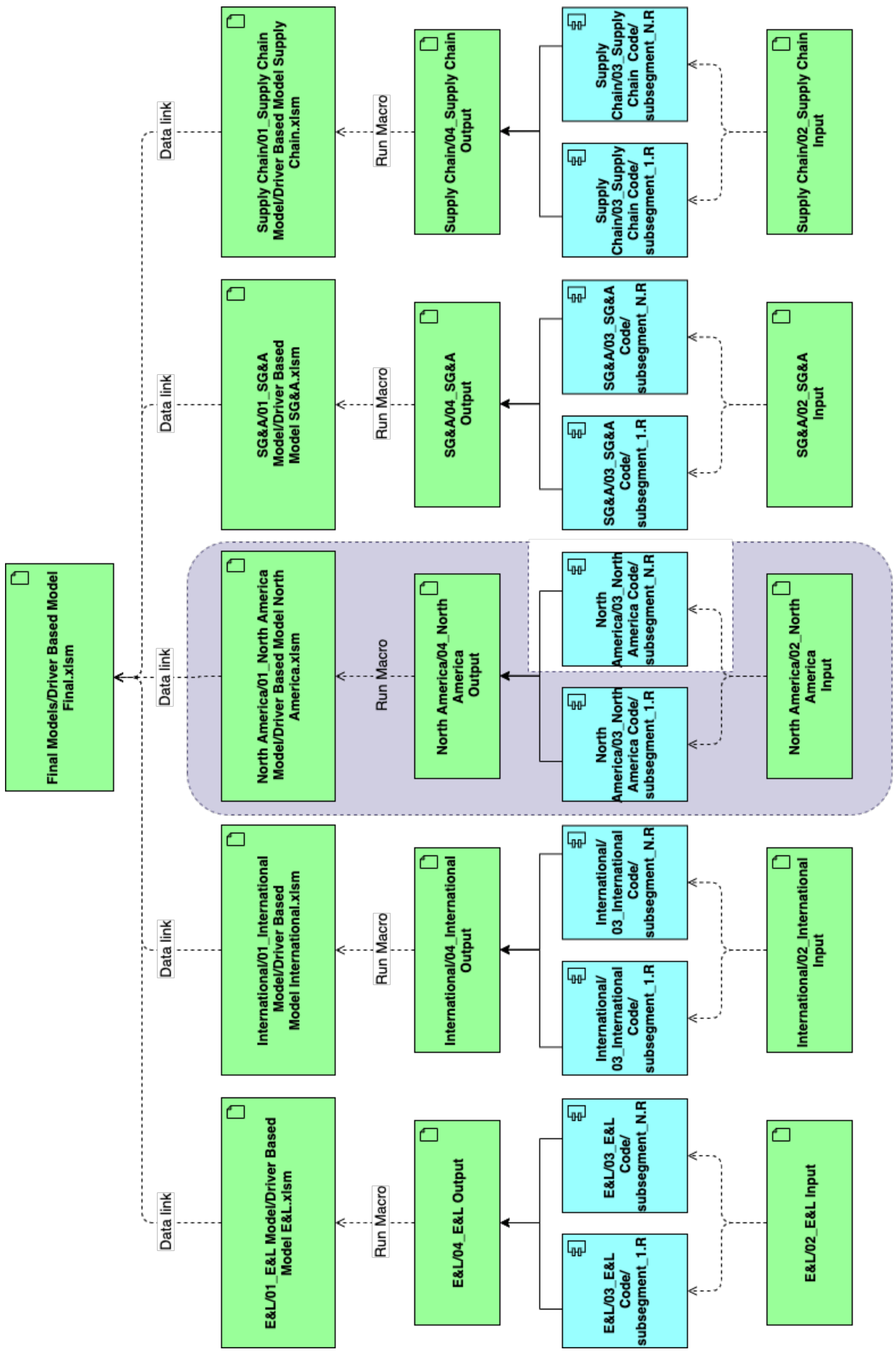


Figure 4.2: Detailed overview of the architecture of PrecisionView™'s prototype solution.

4.2 The Modified Forecasting Dashboard

To test the hypotheses stated in section 1.3, the experimental setup requires the involvement of a Financial Forecasting solution with, and one without XAI. This is because the experiment involves measuring the increase in trust, understanding and aspects of the insights gained, prior to and after the use of XAI. We will elaborate on this in more detail in section 5.1. In this section, we will describe the development of the Financial Forecasting solution without XAI. The reason for rebuilding the forecasting dashboard is two-fold. Firstly, the Tableau dashboard we were provided with did not match the dataset and accompanying forecasts provided. Therefore, we needed to replicate the provided dashboard using the provided dataset. Secondly, as illustrated in section 2.3.4, the original dashboard is fairly extensive. Demonstrating and explaining all sheets and corresponding components would require a significant amount of time from the research participants and makes the experiment unnecessarily complex. Therefore, the dashboard was modified to only incorporate part of the original sheets and components. In this section, we will discuss which sheets and components were selected and why. Furthermore, we will elaborate on the segment and subsegment selection involved in the modified dashboard. Lastly, we will showcase the resulting modified dashboard.

Sheet Selection
As discussed in section 2.3.4, the original Tableau dashboard contains multiple sheets, namely *Executive Summary*, *Sales Details*, *Working Capital Dashboard* and *Planning Overview*. As explained above, showing and explaining all these sheets to the participants requires a significant amount of time and makes it unnecessarily complex. One sheet suffices to demonstrate the XAI techniques and makes the experiment more focused and controllable. Therefore, the dashboard was modified incorporating only the *Planning Overview* sheet, displayed in Figure 2.10. The *Planning Overview* sheet was selected as the focus sheet for a number of reasons. Firstly, it contains the *Forecast* component that provides the forecasts for the main P&L items, namely the net revenue, product cost, the sales, general & administration expenses (SG&A) and the operating profit. The operating profit is not forecasted but derived by subtracting the product cost and SG&A from the net revenue. Secondly, it contains the *Model Accuracy* component that provided two statistics aimed at providing some level of insights into the forecasts and their accuracy. As discussed in section 2.3.4, these statistics are the closest attempt at providing the user with some form of explanation of and insight into the forecasts. Excluding these statistics might paint a wrong picture when evaluating the effectiveness of Financial Forecasting solutions without XAI, and hence should be kept in the dashboard.

Component Selection
For similar reasons as discussed above, not all components of the *Planning Overview* sheet were maintained fully in the modified version of the forecasting dashboard.

- **Forecast Component.** Firstly, the *Forecast* component has been simplified to avoid overcomplication of the validation experiment. As discussed in section 2.3.4, the *Forecast* component contains the option for users to switch between a P&L view and a working capital view of the forecasts. In the modified version this option was eliminated and only the forecasts according to the P&L view are given. Furthermore, alterations were made to the component in order to facilitate the case studies conducted during the experiments, as discussed in section 5.1. Firstly, the forecasts for each of the financial items are given on a monthly basis, instead of on a quarterly basis as is the case in the original forecasting dashboard. Secondly, the high and low forecasts are replaced by the current and last year's actuals to allow for the conduction of the designed case studies, as described in 5.1.

- **Drivers' Analysis Component.** As discussed in section 2.3.4, the *Drivers' Analysis* component consists of a driver inspection functionality and a plot showing the distribution of values for each of the drivers. The driver inspection functionality is removed from the modified dashboard. The reason for doing so is that this functionality makes the conduction of the case studies complicated and potentially lead to unreliable results. The case studies are aimed at testing the participants' understanding of the forecasts and the influences of the corresponding drivers. The driver inspection functionality in the *Drivers' Analysis* component enables users to alter driver values and investigate what happens with the forecasts in the *Forecast* component. Hence, without excluding the driver inspection functionality, it is unsure whether the participant's understanding of the forecasts and influence of corresponding drivers increased due to the use of XAI, or because they had the possibility to inspect driver effects through the driver inspection functionality.

Segment and Subsegment Selection

As stated in 4.1, the architecture for the prototype solution of PrecisionView™ impacts certain decisions made in rebuilding the forecasting dashboard. More specifically, this concerns the selection of segments and subsegments to be used in the modified dashboard. In the modified dashboard, only a single segment and subsegment are used. The reason for doing so is two-fold. Firstly, limiting the forecasts to a single segment and accompanying subsegment is done for similar reasons as for the sheet and component selection discussed above, namely to avoid overcomplication of the dashboard and hence limit the time required for participants to get acquainted with it. Secondly, as discussed in section 4.1, the overall forecasts for a specific line item are obtained by adding the forecasts of each of the individual segments and subsegments for that line item. However, every segment and subsegment uses a different set of drivers for the generation of the forecast model. Therefore, it is not possible to combine the XAI techniques that explain the forecasts of each individual segments and subsegments into a single XAI plot for the overall forecasts. Therefore, in the development of our XAI prototype, we decided to focus solely on the segment *North America* and the corresponding subsegment *U.S.*, as shown in Figure 4.2. This segment and subsegment were selected because the forecasting model for this segment/subsegment combination contained the highest number of drivers, compared to the models of other segment/subsegment combinations. This is an important consideration because we expect that, although models using a higher number of variables are potentially harder to explain, they can also have greater benefit from the XAI techniques. Therefore, we expect models with a higher number of variables to be better suited for our experimental setup.

Worked Example

Based on the above design decisions concerning the sheet, component, segment and subsegment selection, a modified version of PrecisionView™'s prototype solution dashboard was developed.

An overview of the resulting dashboard is shown in Figure 4.13. The modifications made to the individual components are shown in Figures 4.3, 4.4 and 4.5. The figures in the modified version of the forecasting dashboard are based on the data of a toy manufacturing company. The company operates internationally, but as explained above, we limited our modified forecasting dashboard to the U.S. subsegment. Hence, the data in below figures are based on the toy manufacture's sales in the U.S. Firstly, Figure 4.3 shows the Model Error component in the modified version of the forecasting dashboard. As discussed above, the component remained virtually the same as in the original dashboard. The only difference is the location of the *Forecast* and *Statistical Model* select boxes, which were moved to create extra space to display the XAI techniques in the forecasting dashboard with XAI.

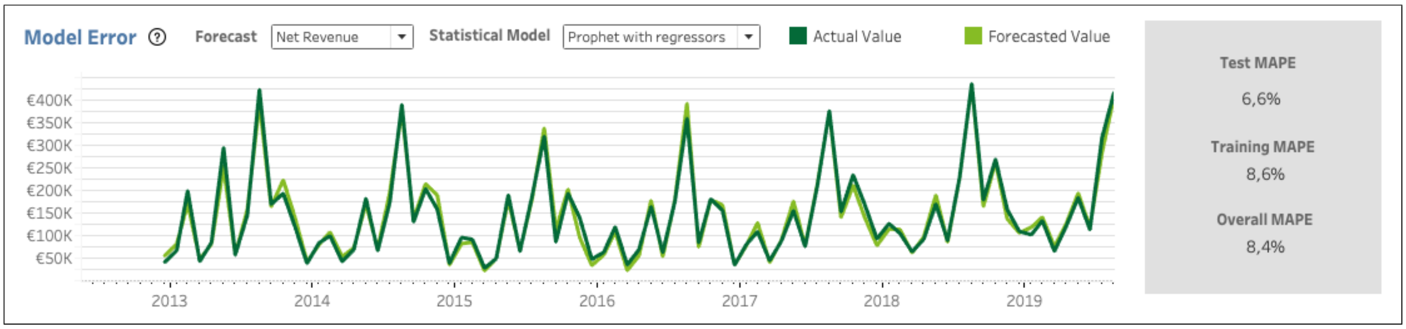


Figure 4.3: The *Model Error* component in the modified forecasting dashboarding without XAI.

The Forecast component of the modified forecasting dashboard is shown in Figure 4.4. The high and low forecast lines have been replaced with the actual values (green) for the financial item in question, as well as last year's value (blue). Furthermore, the forecasts (grey) are given on a monthly rather than quarterly basis. The given forecasts are based on the single segment and subsegment combination, discussed above. This also means that the *segment* and *subsegment* select lists have been removed from the modified dashboard.

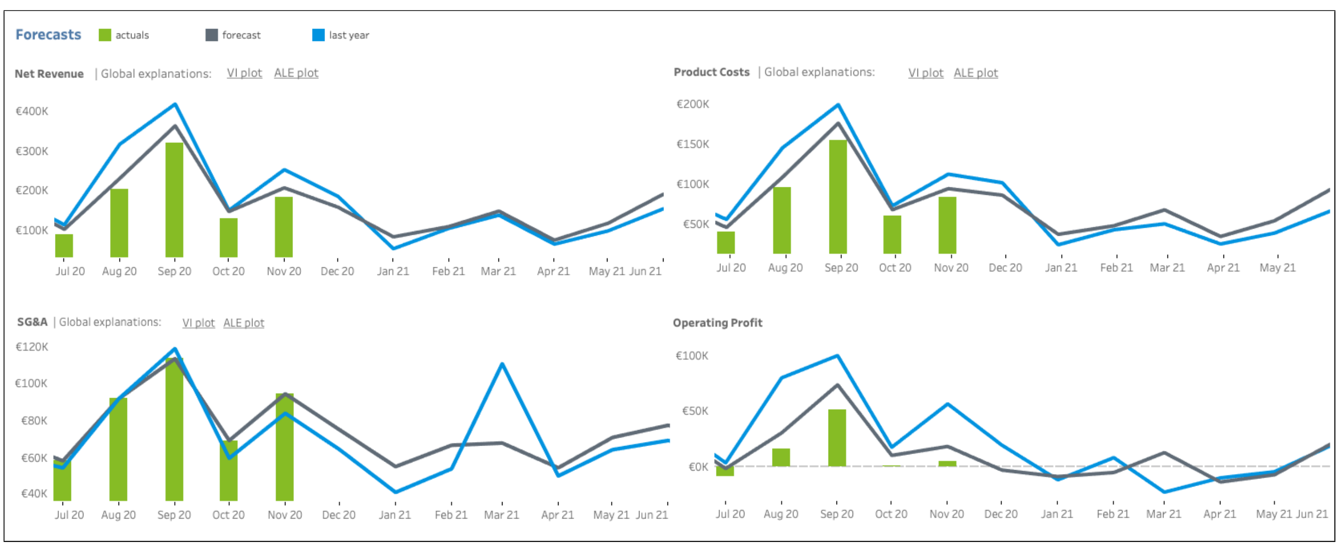


Figure 4.4: The *Forecasts* component in the modified forecasting dashboarding without XAI.

Figure 4.5 illustrates the Drivers' Analysis component in the modified forecasting dashboard. As discussed above, this component was simplified. It no longer contains the driver inspection functionality. Furthermore, the plot showing the distribution of the driver values has been changed to a table format. The reason for changing the visualisation of driver values from graph to table format again was due to create additional space to properly display the XAI techniques in the XAI enabled version of the forecasting dashboard.

Drivers

	Low Driver	Mean Driver	High Driver
Advertising & promotion expenses	€3.654	€12.506	€30.251
EU inflation	2.07	2.65	3.14
External sales	€61.750	€172.088	€399.892
Marketing & sales expenses	€3.495	€4.998	€6.495
Revenue toys and games	€29.150	€29.291	€29.431
Sales allowances	€89.	€906	€4.880
Total administration expenses	€3.247	€14.100	€38.628
Total allowances	€3.847	€10.096	€21.830

Figure 4.5: The *Drivers' Analysis* component in the modified forecasting dashboarding without XAI.

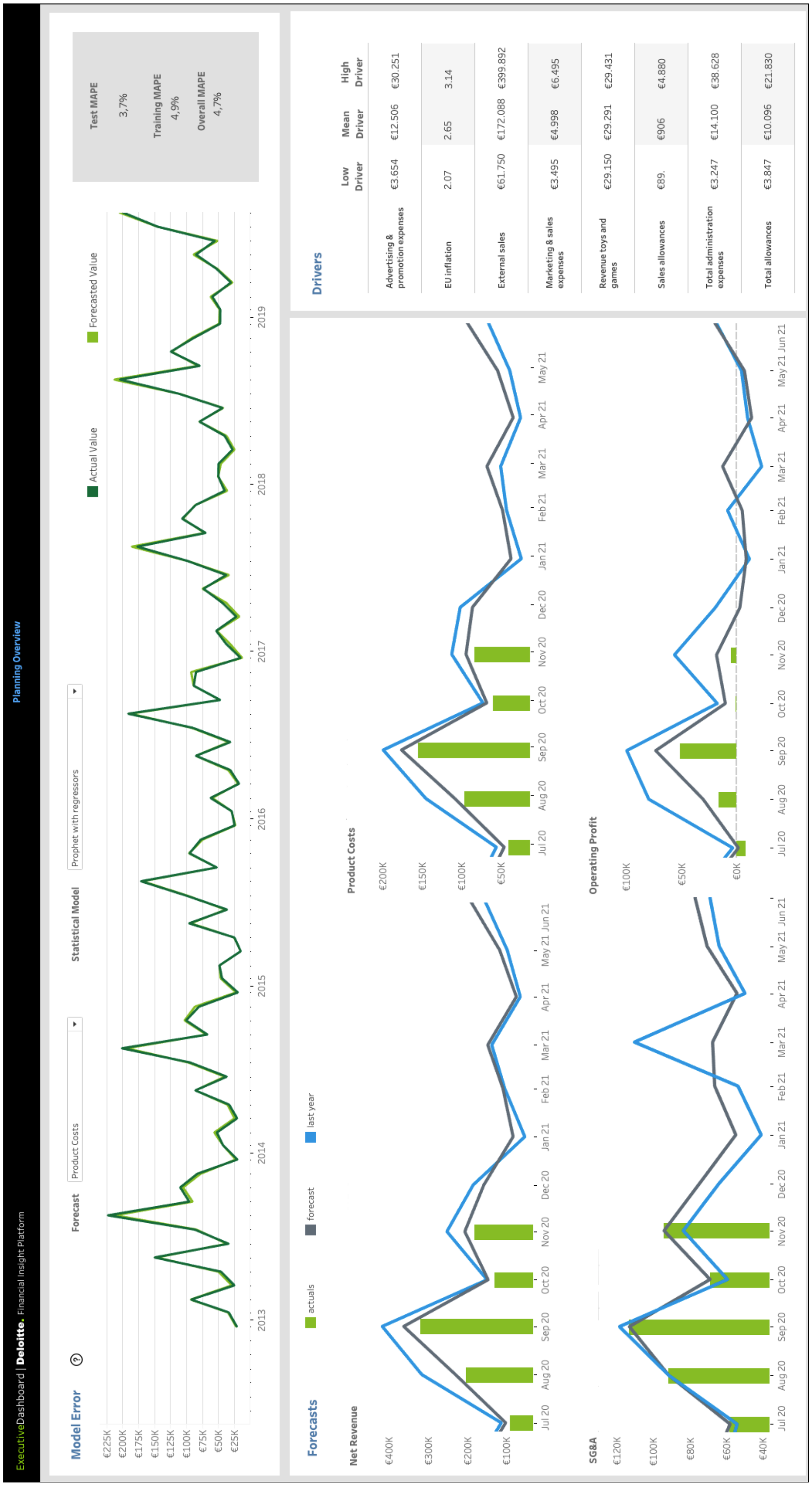


Figure 4.6: The modified Tableau forecasting dashboard without XAI techniques.

4.3 Implementing the XAI techniques

In the previous sections we discussed the architecture for PrecisionView™'s prototype solution, the resulting data structure and the modified forecasting dashboard onto which our XAI prototype is to be implemented. In this section, we will discuss the steps involved in developing the XAI enabled version of the modified Forecasting dashboard discussed above. To this end, we first discuss how the XAI Generation Module is applied to PrecisionView™'s data analytics component, in order to generate the XAI data required to incorporate the XAI in the forecasting dashboard. Secondly, we discuss how the resulting XAI data is used to generate the XAI plots in the forecasting dashboard and how they plots are linked to the forecasts. However, before doing so, we first address the peculiarities that arised with the use of SARIMAX models.

4.3.1 The Impact of SARIMAX on Break-down Plots

Before we can discuss the XAI data generated for the PrecisionView™ specific implementation of the XAI Module, it is important to note certain peculiarities that were discovered regarding the use of ARIMAX models and BD plots. During the implementation of the XAI Generation Module for PrecisionView™, we discovered that the Break-down explanations created for certain ARIMAX models are incorrect. It appears this issue arises when BD plots are created for ARIMAX models using seasonality. This is illustrated by the BD plots depicted in Figures 4.7 and 4.8. In Figure 4.7, where the ARIMAX model does not include seasonality, the intercept and the contributions of the individual drivers correctly add up to the final prediction of €57602. If the ARIMAX model does include seasonality, however, the sum of the intercept and driver contributions does not equal the final prediction. As shown in Figure 4.8, the sum of the intercept and driver contributions amounts to $\sim$ €61585, while the final prediction is lower, namely €58127. We expect this is due to the function used for SARIMAX models, which takes in 4 additional arguments, compared to the function for ARIMAX, discussed in 2.3.2. The form of the function is as follows:

$$ARIMA(p, d, q)(P, D, Q)_m \tag{4.1}$$

where (P, D, Q) is the seasonal part, with P the auto-regressive order, D the degree of differencing and Q the moving average order [HA19]. Furthermore, m indicates the seasonal period, e.g. $m = 12$ is yearly seasonality. Although solving this issue is beyond the scope of this research, for the purpose of future works we will shortly discuss our theory concerning the cause of this problem in section 6.2.

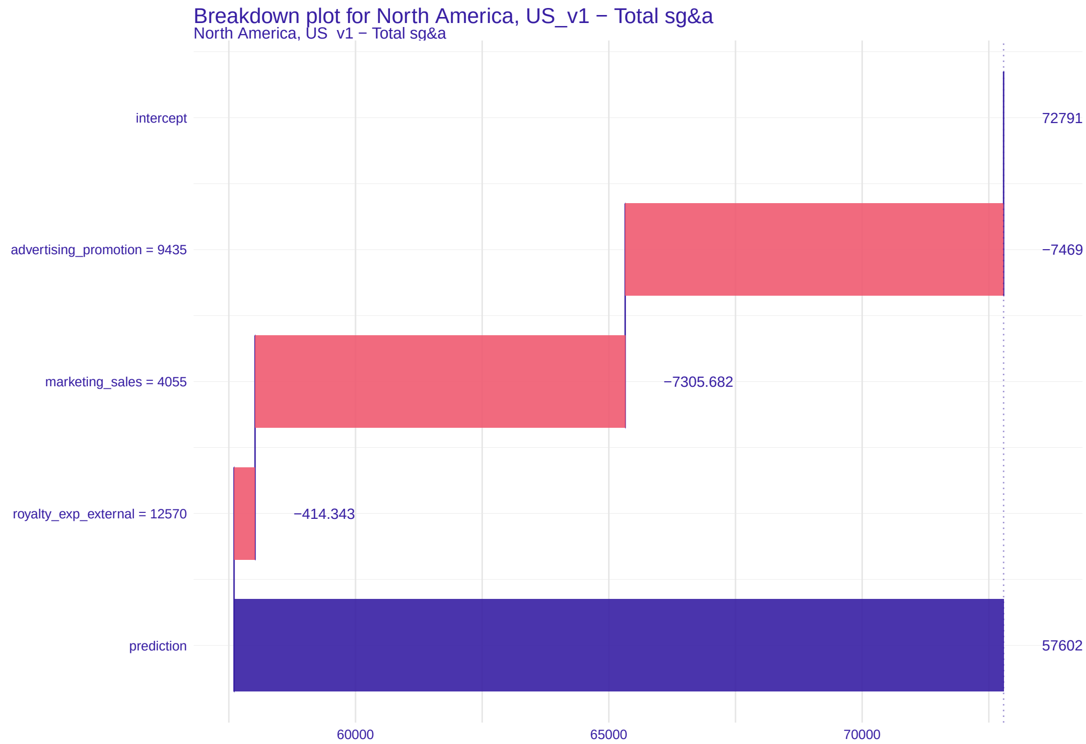


Figure 4.7: The Break-down plot for the predicted SG&A, using a non-seasonal ARIMAX model.

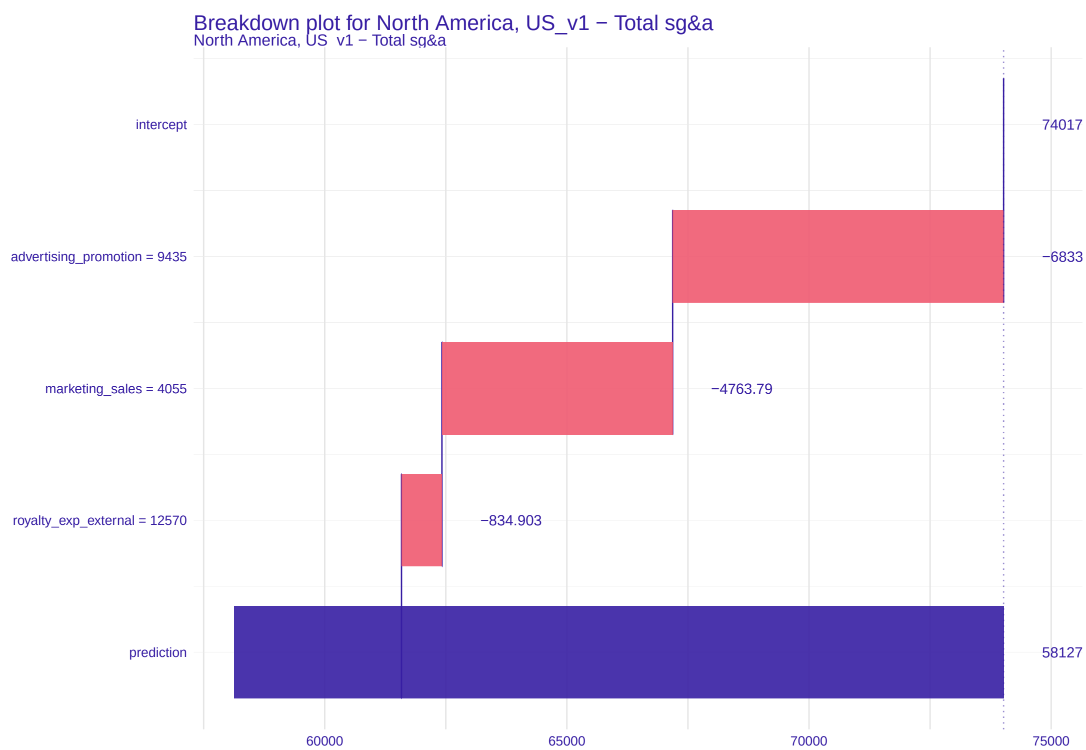


Figure 4.8: The Break-down plot for the predicted SG&A, using a seasonal ARIMAX model

4.3.2 Applying the XAI Generation Module

Here, we will elaborate on the implementation process of the XAI Generation Module, discussed in section 3.5.3, for PrecisionView™ specifically. Furthermore, we will discuss the XAI data that is generated by the application of the module to PrecisionView™, by illustrating the resulting explanation tables discussed in 3.5.3.3.

As we have seen in section 3.5.2, the application of *DALEX*'s XAI functions requires an explainer object containing the forecast model, to be passed. However, the analytics component of PrecisionView™'s prototype solution, consisting of R scripts, does not store the generated forecast models, but only the generated forecasts. Therefore, the XAI Generation Module and accompanying *DALEX* is applied from within these R scripts. The R scripts contain a function for each of the model types, multi-linear regression, ARIMA, ARIMAX, Prophet and Prophet with regressors. In each of these functions, it generates a forecast model for every financial line item to be forecasted, and computes the accompanying forecasts. After the generation of each of these models and accompanying forecasts the *applyXAI* function, given in listing 3.10, is called to obtain the XAI data. Due to the confidential nature of both the R scripts and the datasets they use, we cannot disclose their content. However, in listings 4.1, 4.2 and 4.3 we illustrate how these calls are implemented for the R scripts in PrecisionView™. Furthermore, the *getTidyNames* function used by *applyXAI*, line 7 in listing 3.10, is specific to the Financial Forecasting solution for which the XAI Generation Module is to be applied. The implementation of this function for PrecisionView™ can be found in Appendix A.2

```
applyXAI(inputpath, inputfile, f_revenue, "dyn_regr", revenue_driver,
   drTrainOutputs, predicted_revenue, myPredictionFunction = NULL, t, f,
   "total_net_revenues")
```

Listing 4.1: The call to *applyXAI* used to obtain the XAI data for a multi-linear regression model that predicts the net revenue.

```
applyXAI(inputpath, inputfile, fit_arimax_total_net_rev, "arimax",
   xreg_total_net_revenue, total_net_rev_arimax,
   fcast_arimax_total_net_rev$mean, myPredictionFunction, t, f,
   "total_net_revenues")
```

Listing 4.2: The call to *applyXAI* used to obtain the XAI data for an ARIMAX model that predicts the net revenue.

```
applyXAI(inputpath, inputfile, prophet, "prophet_reg",
   dplyr::select(p_total_net_revenue, -y), p_total_net_revenue_train$y,
   myforecast_total_net_revenue$yhat, myPredictionFunction, t, f,
   "total_net_revenue", originalVars)
```

Listing 4.3: The call to *applyXAI* used to obtain the XAI data for a Prophet with regressors model that predicts the net revenue.

After the application of our XAI module, we obtain a number of explanation table files. These files are used to recreate the explanation plots within the selected dashboarding tool, as explained in sections 3.5.2 and 3.5.3.3. An explanation table file is created for every combination of financial line item, prediction model and XAI plot. For PrecisionView™ specifically, this results in 27 table files, namely $\{$*net revenue, product costs, SG&A*$\}\times\{$*multi-linear regression, ARIMAX, Prophet with regressors*$\}\times$ $\{$*BD plot, VI plot, ALE plot*$\}$. The resulting files for the BD, VI and ALE plots for PrecisionView™'s Prophet model to predict procuct costs are shown in Figures 4.9, 4.10 and 4.11, respectively.

- **BD Explanation Table**
  All columns in the explanation table for BD plots, except for the *date* column, are created by *DALEX*'s *variable_attribution* function. The *date* column is added, because it is required to link the BD plot for every specific forecast instance to the corresponding forecasts in the dashboarding tool. Furthermore, the values of the *contribution* and *cumulative* columns are rounded to the nearest integer. Lastly, as shown in listing 4.3, Prophet passes the *originalVars* argument to the *applyXAI()* function. This array containing the original variables names, is used in the creation of the explanation tables to change the model's variable names in columns *variable* and *variable_name* back to their originals, as shown in listing 3.5 (lines 20:29 and 49:55). This renaming procedure is required due to the fact that Prophet models replace the original variable names with generic names, e.g. *x1, x2, .., xn*, and hence we do no longer know which variables are which, unless we change their names back to the originals.

variable	contribution	variable_name	variable_value	cumulative	sign	position	label	date
intercept	0	intercept	1	80868	1	5	North America, US_v1 - Total product costs	01/03/2018
external_sales = 142100	-18251	external_sales	142100	62617	-1	4	North America, US_v1 - Total product costs	01/03/2018
us_inflation = 2.36	-5631	us_inflation	2.36	56986	-1	3	North America, US_v1 - Total product costs	01/03/2018
Date = 2020-03-01	-165	Date	01/03/2020	56820	-1	2	North America, US_v1 - Total product costs	01/03/2018
prediction	-24047			56820	X	1	North America, US_v1 - Total product costs	01/03/2018
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
intercept	0	intercept	1	80868	1	5	North America, US_v1 - Total product costs	01/12/2022
external_sales = 223400	10575	external_sales	223400	91443	1	4	North America, US_v1 - Total product costs	01/12/2022
us_inflation = 7.244	5590	us_inflation	7.244	97033	1	3	North America, US_v1 - Total product costs	01/12/2022
Date = 2024-12-01	20	Date	01/12/2024	97053	1	2	North America, US_v1 - Total product costs	01/12/2022
prediction	16185			97053	X	1	North America, US_v1 - Total product costs	01/12/2022

Figure 4.9: The resulting explanation table for BD plots, generated for a Prophet with regressors model predicting the total product costs.

- **VI Explanation Table**
The explanation table created to reconstruct VI plots only contains data generated by *DALEX*'s *model_parts* function. As shown in Figure 4.10, for every i^{th} permutation, the file contains the increase in error rate, expressed as *dropout_loss*, for each of the model's independent variables, *us_inflation, Date, external_sales* in this case.

variable	permutation	dropout_loss	label
_full_model_	0	0	North America, US_v1 - Total product costs
us_inflation	0	1.2590509986852	North America, US_v1 - Total product costs
Date	0	77.1265493140234	North America, US_v1 - Total product costs
external_sales	0	77.1859390665145	North America, US_v1 - Total product costs
baseline	0	92.2597365916643	North America, US_v1 - Total product costs
. . .	. . .	. . .	. . .
_full_model_	10	0	North America, US_v1 - Total product costs
Date	10	68.9331791463078	North America, US_v1 - Total product costs
external_sales	10	60.8818534394819	North America, US_v1 - Total product costs
us_inflation	10	1.25972875491793	North America, US_v1 - Total product costs
baseline	10	98.6335742970406	North America, US_v1 - Total product costs

Figure 4.10: The resulting explanation table for VI plots, generated for a Prophet with regressors model predicting the total product costs.

- **ALE Explanation Table**
All the columns in the explanation table for ALE plots as well originate from *DALEX*'s *model_profile* function. The resulting explanation table, depicted in Figure 4.11, contains the predicted value (*yhat*) for each of the observed values (x) of the independent variables (*variable*).

variable	label	x	yhat	ids
external_sales	North America, US_v1 - Total product costs	34276	23712	0
external_sales	North America, US_v1 - Total product costs	39670.95	25624	0
. . .	. . .	. . .	. . .	. . .
external_sales	North America, US_v1 - Total product costs	487768.45	180006	0
external_sales	North America, US_v1 - Total product costs	511282	184447	0
us_inflation	North America, US_v1 - Total product costs	-0.2	68167	0
us_inflation	North America, US_v1 - Total product costs	-0.10565	68384	0
. . .	. . .	. . .	. . .	. . .
us_inflation	North America, US_v1 - Total product costs	3.78555	77062	0
us_inflation	North America, US_v1 - Total product costs	3.868	77324	0

Figure 4.11: The resulting explanation table for ALE plots, generated for a Prophet with regressors model predicting the total product costs.

4.3.3 Incorporating XAI in the Dashboard

Here, we will discuss the process of incorporating the XAI data, produced in the previous step, in order to obtain the XAI enabled Financial Forecasting solutions for PrecisionView™. This process involves selecting the right explanation table files, creating the different XAI plots in Tableau, and linking them to the corresponding individual forecasts or forecast models.

As discussed in the previous section, several explanation table files are created for every forecasting model, i.e. multi-linear regression, ARIMAX and Prophet with regressors. However, PrecisionView™ does not use all forecasting models from the analytics phase in the dashboarding phase. Instead, it determines which model achieves the highest accuracy at predicting each of the financial items, using the MAPE measurement. Therefore, before we can implement the XAI plots in the dashboarding tool, we need to select the right explanation table files based on the models selected by PrecisionView™. The selection of best performing models is done inside the driver based excel model for a specific segment, shown in figure 4.2. For the dataset used in the prototype solution, the Prophet with regressors model has the highest prediction accuracy for all three forecasted line items, *net revenue, product costs* and *SG&A*. Therefore, only the 9 explanation table files created for the Prophet model are used for the implementation of the XAI plots in Tableau.

After the correct XAI table files are determined, they are uploaded to Tableau in order to reconstruct the XAI plots. For each forecasted financial item and every XAI plot, the process starts with uploading one of the 9 corresponding explanation table files, discussed above. Next, a new worksheet is added in which the plot will be implemented. The remaining steps for the creation of the plots, depends on the plot type. Below we will discuss the process involved in reconstructing these plots for Tableau specifically. For reference purposes, the interface for the creation of new worksheets in Tableau is depicted in Figure 4.12.

Break-down Plots
The reconstruction of BD plots requires a more complex procedure compared to the VI and ALE plots. It consists of the following steps described below. The resulting BD plot is shown in Figure A.4 in Appendix A.2.

1. Drag the *variable* dimension onto the rows shelf.
2. Right click the *variable* row → 'Sort..', and set 'sort by' = field, 'sort order' = descending and 'Field name' = *position*. This ensures that the variables with the biggest contribution to the final prediction are shown at the top of the BD plot.
3. Drag the *cumulative* measure onto the columns shelf.
4. Select 'Gantt bar' in the Marks tab.
5. Drag the *contribution* measure onto the Size Marks Card. This ensures that the bar plots for every variables are sized based on the contribution of that variable.
6. To get the bars to line up properly, double-click the *contribution* measure that is assigned to the Size Marks Card and add a negative sign(-) to the formula.
7. Drag the *sign* dimension onto the Color Marks Card and change the colors of the dimensions shown in the legend on the right.
8. Right click on the x-axis → 'edit axis..' and uncheck the 'include zero' checkbox.
9. Clear the 'Title' field under 'Axis Titles'
10. Drag the *contribution* measure onto the Label Marks Card. Edit the formula in this field to field to the following: `IF [variable] != "intercept" THEN [contribution] ELSE [cumulative] END`.

Variable Importance Plots
The process for recreating the VI plots is done by executing the following steps. The created VI plot is given in Figure A.5 in Appendix A.2.

1. Drag the *variable* dimension onto the rows shelf.
2. Drag the *dropout_loss* measure onto the columns shelf.
3. Double click *dropout_loss* to change its function from SUM() to AVG(). This is required to show the average *dropout_loss* over all permutations.
4. Right click 'Rows' → 'Sort..'. Select 'sort by' = field, 'sort order' = descending, 'Field name' = *dropout_loss* and 'aggregation' = Average. This ensures that the variables in the VI plot are sorted based on the highest increase in prediction error after permutation.
5. Drag the *variable* dimension to the filters and uncheck the *_baseline_* and $\_full_model\_$ fields. This ensures that only the *dropout_loss* for the individual variable is shown.
6. Clear the 'Title' field under 'Axis Titles' and alter the ticks step size.

Accumulated Local Effect Plots
Reconstruction of the ALE plots is achieved by the following steps. The generated ALE plot is illustrated in Figure A.6 in Appendix A.2.

1. Drag the x measurement onto the columns shelf.
2. Right-click the *variable* measurements and select 'Dimension'. This is required in order for *variable* to be used on the x-axis of the ALE plot.
3. Drag the *variable* dimension onto the columns shelf.
4. Drag the *yhat* measurement onto the rows shelf.
5. Change the 'Title' field under 'Axis Titles' (y-axis), uncheck 'include zero' and format.
6. Change the 'Title' field under 'Axis Titles' (x-axis), select 'independent axis ranges for each row or column' and format. This ensures that for each variable in the ALE plot a separate plot is created, similar to the plot in Figure 3.11.

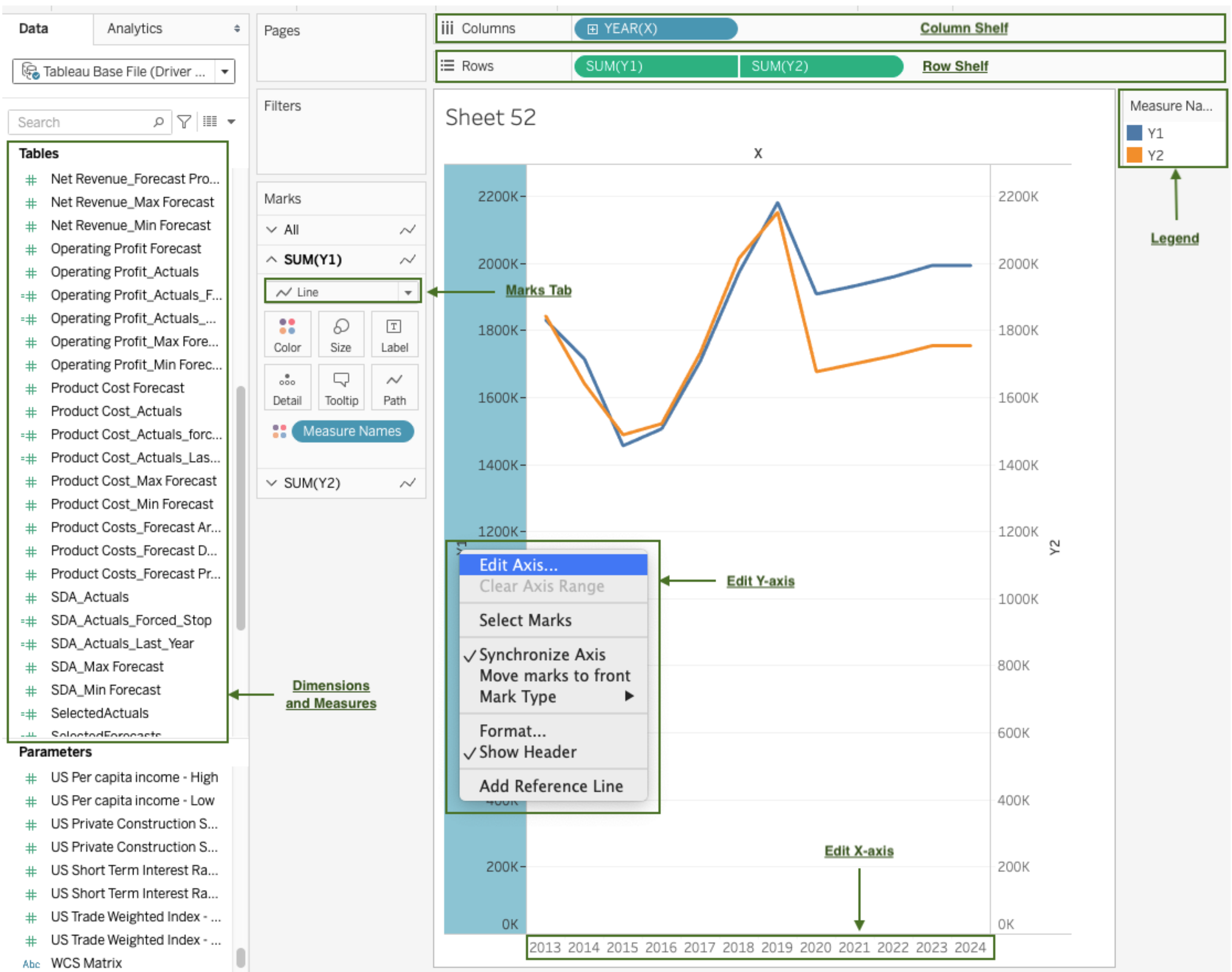


Figure 4.12: Adding a new worksheet in Tableau.

Once the XAI plots for every financial item are created, they have to be linked to the corresponding forecast instance or model. More specifically, BD plots are linked to forecast instances as they provide a local explanation. ALE and VI plots are linked to forecast models, since they provide a global explanation for the model as a whole. Below we describe these two different processes.

Linking BD Plots

The process of linking the recreated BD plots to the corresponding forecast instances consists of the two steps described below. These two steps have to be repeated for every forecasted financial item.

1. **Creating a link to the BD plot dashboard.**
    - Create a new filter action by navigating to 'Dashboard' → 'Actions ..' and select 'Add Action >' → 'Filter..'.
    - Under 'Source Sheets', select the dashboard that contains the forecasts and check the forecast plot to which the BD plot should be linked. Furthermore, select 'Run action on' = 'Menu'. The 'Menu' option ensures that the link to the BD plot will appear when a specific forecast instance within the forecast plot in question is clicked upon.
    - Next, under 'Target Sheets', select the dashboard containing the BD plots for the forecasts in question, and check the specific worksheet containing the BD plots. Furthermore, select 'Clearing the selection will' = 'Leave the filter'.
    - Under 'Target Filters' select 'All Fields'.
2. **Filtering the BD plot for a specific instance.** Providing a link from a certain forecast plot to the corresponding BD plot is not sufficient. In addition to the link, a filter is required that tells the dashboard containing the BD plots, for which specific forecast instance we want to obtain the BD plot. This filtering is performed based on the date of a forecast instance. The reason for this is that the *date* variable is the only variable of our forecast model of which we can assume uniqueness.
    - Navigate to the worksheet containing the forecast plot for the financial item in question, e.g. 'Forecast Product Cost'.
    - Click the *MONTH(Period)* dimension in the filters window and select 'Apply to Worksheets' → 'Selected Worksheets..'. In the pop-up window, select the worksheet that contains the BD plots for the forecasted financial item in question.
    - Apply the same procedure as described above for the *YEAR(Period)* dimension filter. These two filters ensure that when a specific forecast instance is clicked in the forecast plot, that the corresponding month and year of that instance are passed to the BD plot. This allows the BD plot to filter on the given month and date and obtain the BD plot for that specific instance.

Linking VI and ALE plots

The steps required to link VI or ALE plots to their corresponding forecast models is more straightforward, since it does not require filtering the XAI plot for a specific instance. Furthermore, the creation of links to both the VI and ALE plots as well is less complex. Lastly, similar to BD plots, the process described below has to be repeated for every forecasted financial item.

1. **Creating a link to the VI or ALE plot dashboard.**
    - Navigate to the dashboard sheet contain the forecast plots to which the VI and ALE plots should be linked.
    - Add a 'Navigation' object to this dashboard sheet, for both the VI and ALE plot.
    - Click on the added navigation object and select 'Edit Button..'.
    - Under 'Navigate to', select the dashboard sheet that contains the VI or ALE plot for the forecasted financial item in question.

4.4 Worked Example

In the previous section we illustrated how to incorporate the selected XAI techniques into PrecisionView™. In this section, we will showcase the resulting XAI enabled Financial Forecasting solution for PrecisionView™. This resulting XAI enabled dashboard was further improved upon in collaboration with professional within Deloitte. We will discuss these improvements and highlight their contributions to the final version of the dashboard.

At first sight, the main dashboard sheet, the *Planning Overview*, does not appear to differ greatly from the dashboard without XAI, displayed in 4.6. In fact, the only changes to this sheet include the links to the XAI plots that were added to the *Forecasts* component, indicated by the orange boxes in Figure 4.13. As explained in the previous section, these links navigate to separate dashboard sheets containing the BD, VI and ALE plots. These dashboards sheets, for the BD, VI and ALE plot, are presented in Figure A.4, A.5 and A.6, respectively.

Figure 4.13: The Tableau forecasting dashboard with XAI techniques.

After completing the preliminary version of XAI enabled dashboard shown in Figure 4.13, a number of sessions were held with a senior consultant and consultant at Deloitte's Finance & Performance team, experienced with Business Intelligence and Dashboarding tools. The aim of these sessions was to further improve the obtained XAI dashboard from a visual perspective. Drawing from their experience with the use of BI tools amongst financials, they provided several aspects to improve upon. This feedback included the following points:

- The confusion caused by offering users the ability to select a forecast model inside the *Model Error* component. This confusion is caused by the fact that changing the forecast model also changes the *Actual Values* and *Forecasted Values* plots inside the *Model Error* component, but does not have any effect on the forecast plots inside the *Forecasts* component. To address this confusion, they suggest informing the user that the best performing forecasting model that determines these forecast plots is chosen automatically and can therefore not be changed by the user.

- The ambiguity regarding the specific forecasting model that was used to derive the forecasts shown in the *Forecasts* component. More specifically, in the *Model Error* component, users can select several forecasting models. However, it is not clear which of these models were used to obtain the forecasts in the *Forecasts* component.

- They find the layout of the components inside the *Planning Overview* sheet to be suboptimal. The *Forecasts* component is the key component in this dashboard, not the *Model Error* component. Hence, it makes more sense to place the *Forecasts* component on the top of the screen and the *Model Error* component below it.

- They recommended displaying the XAI plots insides the *Planning Overview* dashboard, instead of forcing users to navigate to a different dashboard sheet. The motivation for this is that allowing for the XAI plots to be displayed inside the *Planning Overview*, enables users to view them in the context of the forecast plots without having to navigate back and forth. In addition to that, it enables users to view XAI plots for multiple financial forecasts at once, which supports the analysis of a single driver's impact on different financial forecasts. They suggested the following approaches for the local and the global explanations.

  1. For the BD plots they recommended displaying them inside the corresponding forecast plot, by minimizing the forecast plot by 50% when a specific forecast instance is clicked upon. This allows the BD plot to be displayed in the freed up space inside the forecast plot.
  2. For the VI and ALE plots, they suggested showing the plots when hovering over the "VI plot" and "ALE plot" displayed in Figure 4.13, respectively.

- Lastly, they suggested expanding the menu in the top of the dashboard to allow for the inclusion of an *Info Center* where users can find additional explanations on the XAI plots and specific drivers.

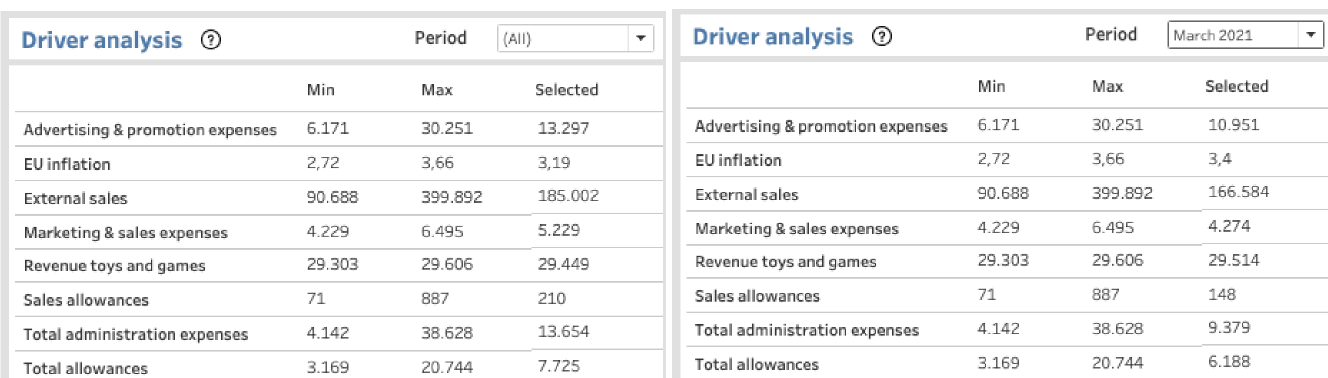
With the help of Blaisdell and Lahaije, we implemented the suggested changes listed above. The improved version of the dashboard was then tested amongst two informal connections of the researcher. Based on these test rounds, we identified a few more aspects to improve upon. These included:

- Allowing users to navigate to the VI and ALE plots located in a separate dashboard sheet while remaining the hover over functionality discussed above. The motivation for doing so is that during the test rounds it was found that users struggled to view the details of these XAI plots. Allowing for both a hover and click functionality enables users to quickly view these global plots inside the context of the *Planning Overview* dashboard, as well as viewing the plots in more detail if desired.

- The above improvement increases the navigation required by the users of the dashboard. However, as discussed above, Blaisdell and Lahaije recommended limiting the navigation required. Therefore, it was decided to combine the VI and ALE plot in a single dashboard sheet.

- Lastly, we added the ability to view the driver value for a specific month to the *Drivers' Analysis* component. As shown in Figure 4.5, the *Drivers' Analysis* component initially only contained the minimum, maximum and average values over the given forecast period. Through the addition of a select box where users can select a specific forecast month, they can also obtain the driver values for that specific month.

The final XAI enabled Financial Forecasting solution implemented for PrecisionView™ is shown in the figures below. The *Planning Overview* dashboard in Figure 4.20 now contains a larger top bar that includes the info center. Furthermore, the *Forecasts* and *Model Error* Component are swapped, and the *Drivers' Analysis* component is reduced to allow for the expansion of the BD plots, shown in Figure 4.16. The *Drivers' Analysis* component, depicted in Figure 4.14, now includes a question mark to provide users with additional information on how to interpret the values shown in the table. Furthermore, as shown in Figure 4.14b, it now contains a select box to select the desired period for which a user wants to view the driver variables. If no period is selected, the average driver value over the forecasted period is given, as illustrated in Figure 4.14a.



Driver analysis ⓘ Period (All)

	Min	Max	Selected
Advertising & promotion expenses	6.171	30.251	13.297
EU inflation	2,72	3,66	3,19
External sales	90.688	399.892	185.002
Marketing & sales expenses	4.229	6.495	5.229
Revenue toys and games	29.303	29.606	29.449
Sales allowances	71	887	210
Total administration expenses	4.142	38.628	13.654
Total allowances	3.169	20.744	7.725

(a) Showing the average driver values.

Driver analysis ⓘ Period March 2021

	Min	Max	Selected
Advertising & promotion expenses	6.171	30.251	10.951
EU inflation	2,72	3,66	3,4
External sales	90.688	399.892	166.584
Marketing & sales expenses	4.229	6.495	4.274
Revenue toys and games	29.303	29.606	29.514
Sales allowances	71	887	148
Total administration expenses	4.142	38.628	9.379
Total allowances	3.169	20.744	6.188

(b) Showing the driver values for a specific period.

Figure 4.14: The *Drivers' Analysis* component in the final version of the dashboard.

Figure 4.15 displays the *Model Error* component, now renamed to *Forecasts' Model Error*. As shown, it contains an additional text box stating which forecast model was used to derive the forecasts in the *Forecasts* component and to indicate that the 'Select a Model' select box does not influence these forecasts.

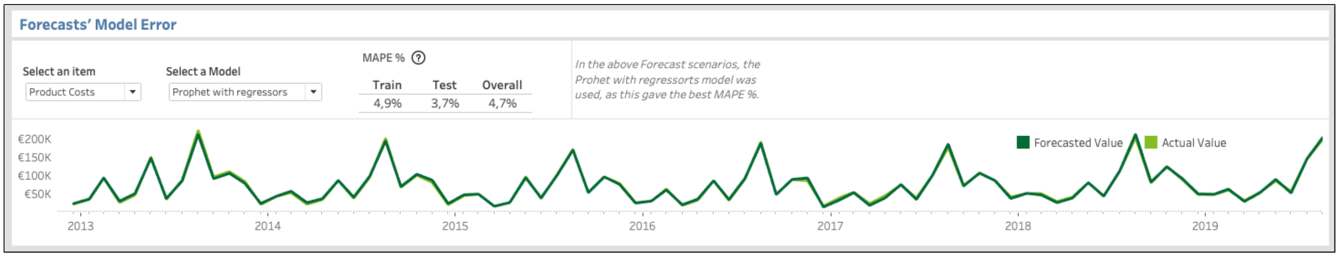


Figure 4.15: The *Forecasts' Model Error* component in the final version of the dashboard.

Figures 4.16, 4.17 and 4.18 illustrate the XAI plots shown from within the *Planning Overview itself*. Users are now enabled to view multiple XAI plots of the same or different financial forecasts, simultaneously. For example, users can compare the BD plots for the net revenue, product cost and SG&A at once, or view both the VI and BD plot for the net revenue at the same time.

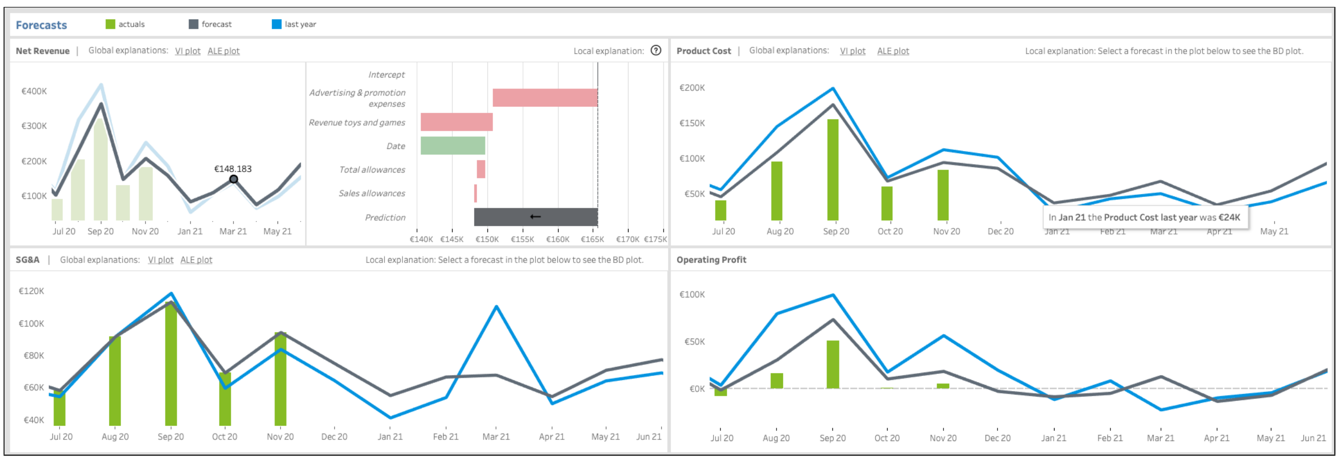


Figure 4.16: The BD plots in the final version of the dashboard, displayed by minimizing the corresponding forecast plot when clicking on a specific forecast instance.

Figure 4.17: The VI plot displayed when hovering over the corresponding 'VI plot' link.

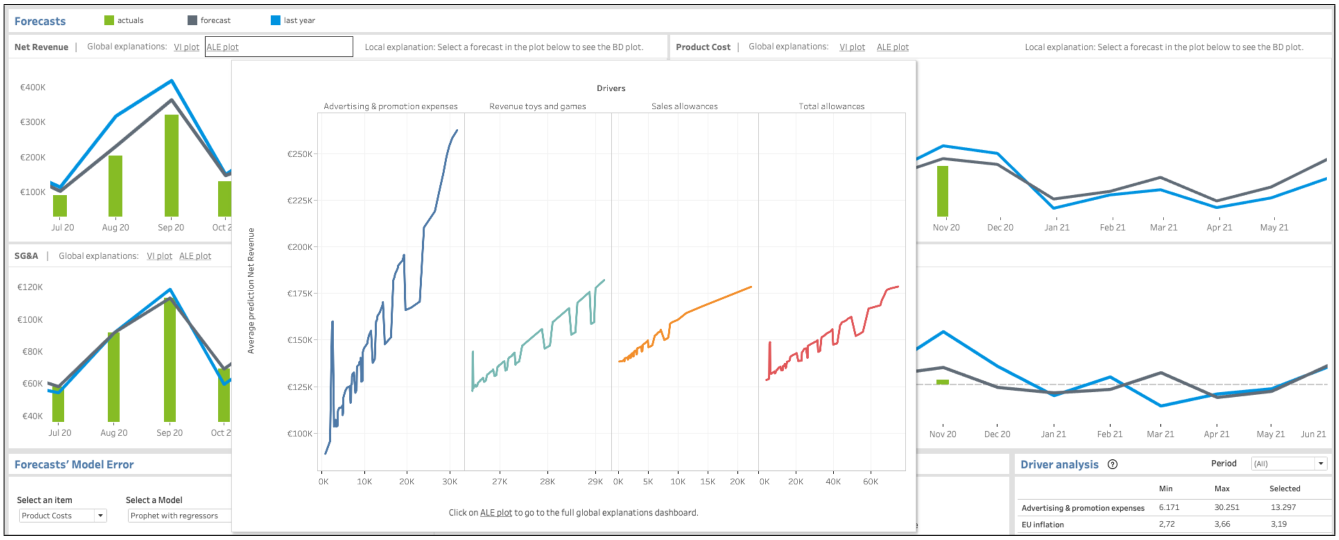


Figure 4.18: The ALE plot displayed when hovering over the corresponding 'VI plot' link.

Furthermore, users still have the ability to click on both the 'VI plot' and 'ALE plot' link to navigate to a separate dashboard sheet containing these two global explanations. This dashboard sheet is shown in Figure 4.19 and allows users to inspect the two plots in more detail. This includes obtain the precise increase in prediction error within the VI plot or viewing the exact effect of specific driver values on the average prediction within the ALE plot.

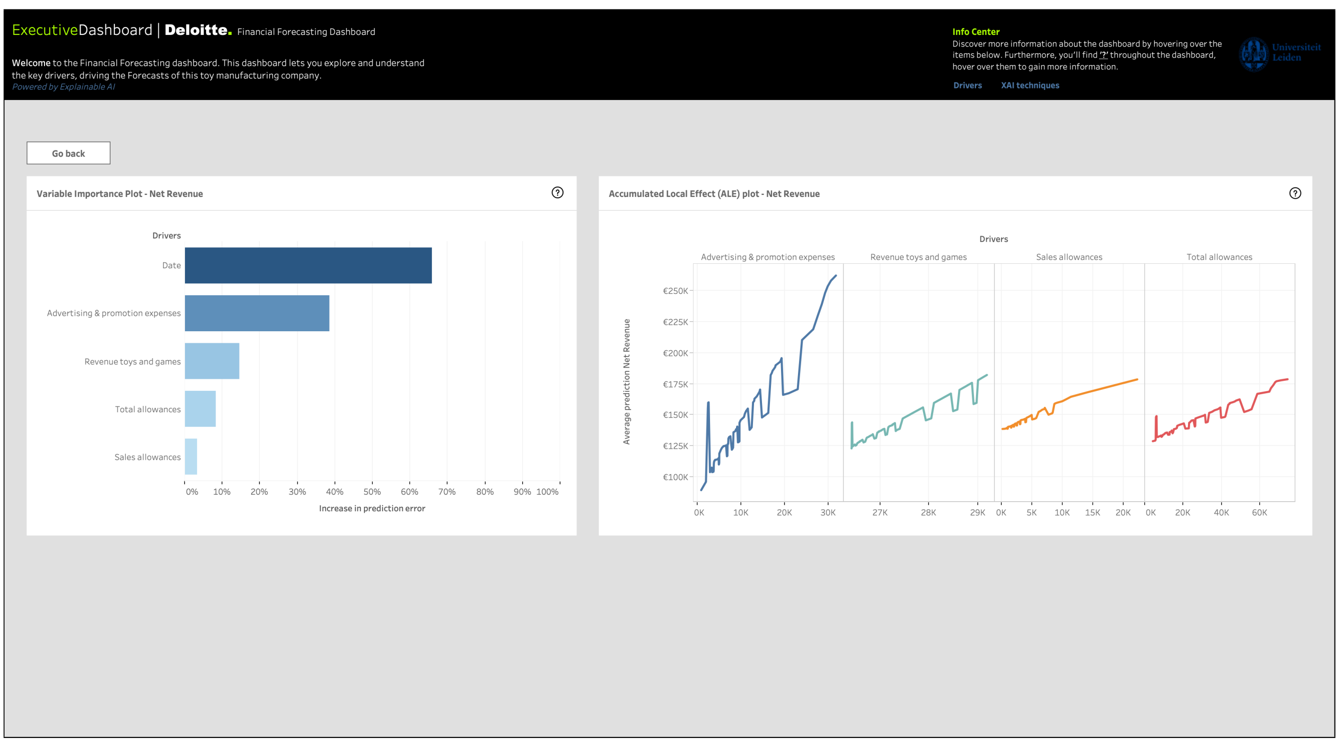


Figure 4.19: The dashboard containing the global VI and ALE explanations, accessed when clicking on the corresponding 'VI plot' and 'ALE plot' links in the *Planning Overview* dashboard.

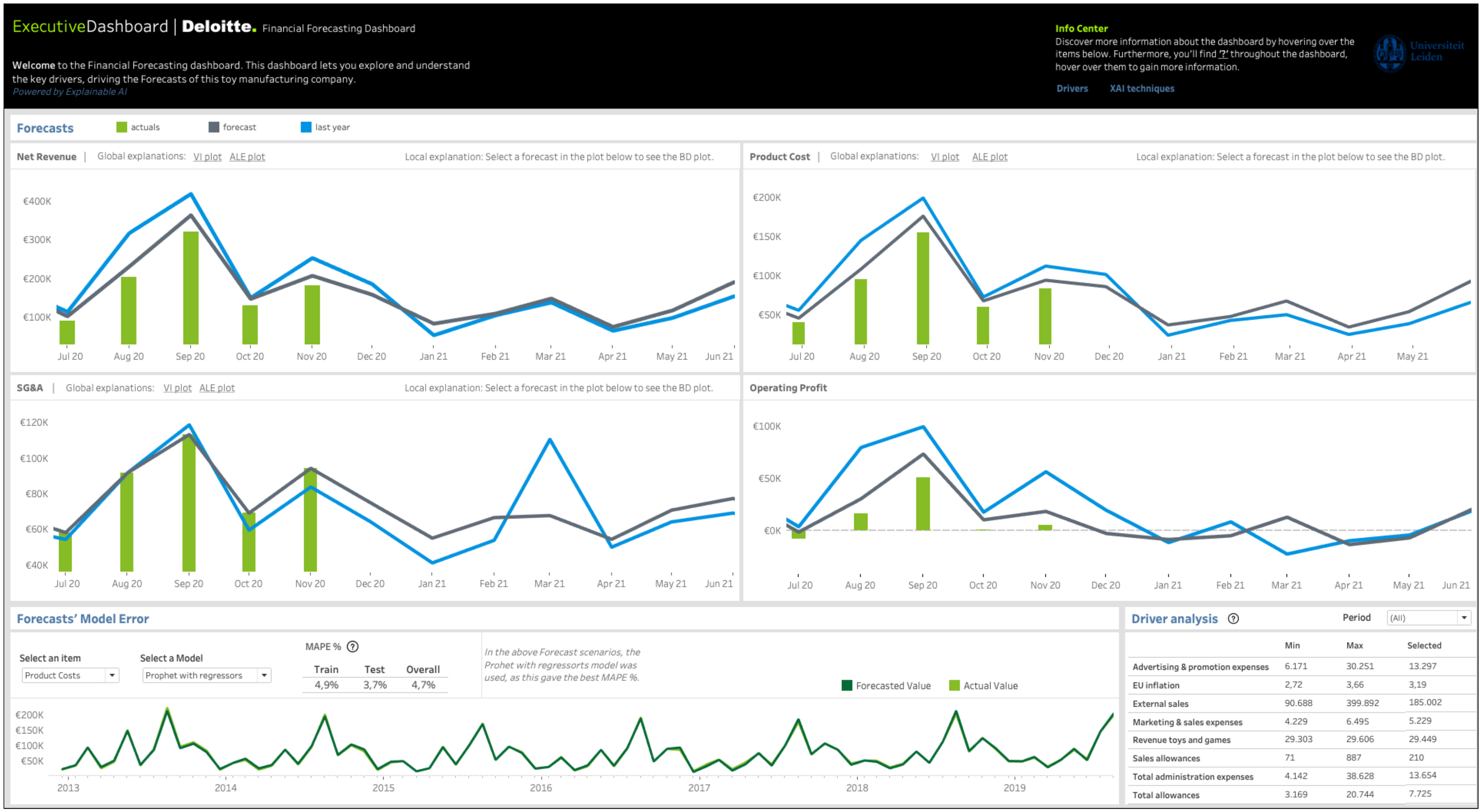


Figure 4.20: The *Planning Overview* sheet in the final version of the dashboard.

5 Prototype Validation

In this section, we will discuss the validation of our XAI prototype for Financial Forecasting solutions. In section 5.1, we will elaborate on the experiment design, where we discuss the overall approach for the prototype validation and the measurements used to test the hypotheses stated in section 1.3. Secondly, in section 5.2, we will report on the results obtained from this validation approach. Lastly, in section 5.3, we will discuss the meaning of the results and their implications with respect to the research questions addressed in this thesis.

5.1 Experiment Design

The experiment design consists of two main aspects, the research setup specifics and the measurements used to test the hypotheses. In the discussion of our research setup we will elaborate on the overall research approach, details concerning with whom and where the validation took place, the stimuli material used and the methods involved in data collection. Next, in section 5.1.2, we elaborate on the measurements used for the validation of the different research objectives. First, we discuss the setup of the case studies that were used to investigate the influence of XAI on the derivation of insights. Next, we provide the scales used to measure the effect of XAI on both trust and understandability. Lastly, we elaborate on the satisfaction scales, used to evaluate the explanation satisfaction of the individual XAI techniques.

5.1.1 Research Setup

As briefly discussed in section 1.4, the research approach used for the validation of the XAI prototype consists of a mix of both qualitative and quantitative methodologies. The quantitative methodology consists of the measurement scales for trust, understandability and explanation satisfaction, as well as the scores computed for the case studies. However, we also recorded the participants during the conducting of the case studies to obtain additional, qualitative data to assist in the interpretation of the case study results, if needed. Furthermore, the explanation satisfaction scales each were followed by an open-ended question to obtain additional feedback on each of the three XAI techniques. The order in which the case studies and measurements were taken is as follows. Prior to the actual start of the experiment, research participants were asked a number of demographic questions, relating to their education, professional background, and experience with Financial Forecasting solutions. Next, participants were asked to perform a case study based on a Financial Forecasting dashboard without XAI. The case study is followed by two measurement scales relating to their trust in and understandability of this dashboard without XAI. Upon completion, they start a second case study based on a Financial Forecasting dashboard that does contain XAI, namely the XAI prototype developed in this research. Again, this case study is followed by two scales relating to their level of trust in and understanding of the XAI enabled Financial Forecasting dashboard. After having completed the case studies and trust and understandability scales for both the dashboard without and with XAI, they are presented three more measurement scales. These scales relate to their level of satisfaction with the explanations provided by each of the three individual XAI techniques implemented in the XAI prototype.

The research was conducted through a series of one-on-one video meetings. Video meetings were chosen on the basis of three reasons. Firstly, the presence of the researcher during the experiment was required to share the Financial Forecasting dashboards with the participants. The files containing the dashboards could not be shared electronically. This is due to both the confidential nature of the data contained in the dashboard, as well as due to the payed software required to open these dashboard files. Secondly, presence was required in order to record the conduction of the experiment. Thirdly, due to Covid-19, this presence could not take place through in person experiments. Hence, it was decided to conduct online video meetings, through which the researcher could present and give control over the forecasting dashboard through screen-sharing, while simultaneously recording the experiments. The experiments where conducted amongst a small group of Finance professionals, working either as consultants of Finance Functions or as Finance or Business Controllers within the Finance Function.

The stimuli material involved in the validation include the Financial Forecasting dashboard without XAI, discussed in section 4.2, the developed XAI prototype, discussed in section 4.4 and a demonstration video showcasing both these dashboards. The purpose of the demo was to get acquainted with the Financial Forecasting dashboard and the implemented XAI techniques. The reason for doing so is, as we will discuss in more detail in section 5.1.2, measuring trust in and understandability of XAI systems, requires the users of the system to have some level of experience with it. Research participants were sent a link to this demonstration video, with the request to watch the demo prior to the conduction of the experiment. As explained above, the two versions of the Financial Forecasting dashboard where made available to the participants through the screen-sharing functionality within video conference software. Lastly, the data collection methods used to record the results are an online survey tool and video recordings. The online survey tool Qualtrics was used to record the responses for the case studies and the measurement scales. As discussed above, the video recordings were aimed at collecting additional, quantitative data on the conducting of the case studies.

5.1.2 Measurements

As explained above, the research approach for the validation of the XAI prototype involves a case study and three different measurements scales. In this section, we will discuss each of them and explain what their purpose is in relation to the research objective.

Understandability and Trust Scales
As stated in section 1.3, one of the main objectives of this research was to investigate whether XAI techniques have the intended effect, namely to solve the black-box problem, when applied within the financial domain. As we have seen in section 2.4.2, the main causes for the need of XAI are the lack of trust and understandability. Therefore, we defined the impact of XAI on the black-box problem found in Financial Forecasting solutions, in terms of the level of trust in and understanding of those Financial Forecasting solutions. In order to test this, the following hypothesis were derived in 1.3:

H1. The use of XAI helps overcome the black-box problem by ..

H1a. increasing the understandability of Financial Forecasting solutions.

H1b. increasing the trust in Financial Forecasting solutions.

As explained above, to measure understandability and test hypothesis **H1a.**, we asked participants to indicate their level of understandability prior to and after the implementation of our XAI prototype. The scale that was used to measure understandability is part of the measurements in the scale developed by Madsen and Gregor [MG00]. The Madsen-Gregor scale was designed to measure human-computer trust by measuring five factors, namely reliability, technical competence, understandability, faith and personal attachment. In this research, we adapted the items from the understandability factor. The reason for choosing this scale is two-fold. Firstly, the scale was developed to be used in a similar context as the context of our validation experiment, namely human-computer interaction. Secondly, the authors tested the validity of their scale and showed very high reliability ($\alpha = 0.94$) for the scale as a whole. Furthermore, the reliability coefficient for the understandability scale on its own has proven to be sufficiently high as well ($\alpha = 0.84$), and hence can be used stand-alone. The items making up the understandability scale are listed below. Participants were asked to rate each item on a 5-point Likert scale ranging from *strongly disagree* (1) to *strongly agree* (5).

1. I know what will happen the next time I use the system, because I understand how it behaves.
2. I understand how the system will assist me with decisions I have to make.
3. Although I may not know exactly how the system works, I know how to use it to make decisions about the problem.
4. It is easy to follow what the system does.
5. I recognize what I should do to get the advice I need from the system the next time I use it.

To measure the effect of XAI on the level of trust in Financial Forecasting solutions and test the corresponding hypothesis **H1b**, an adopted version of the Cahour-Forzy scale was used. Cahour and Forzy initially developed their scale to measure the trust users place in a cruise control system [CF09]. Their scale was selected based on the extensive review performed by Hoffman et al. In their paper, they review a number of metrics used to measure different aspects of XAI. Based on this review, they provide several recommendations with regards to the correct use of those metrics [HMKL18]. In their review of trust measurement scales, they find that several of these scales are similar to or duplicate items from the Cahour-Forzy scale. Based on this review, they propose a new scale in which several items of the scales that were reviewed, are combined. The majority of the items in this scale as well originate from the Cahour-Forzy scale. Due to practical time issues, we had to limit the size of the scales used in the validation experiment. Therefore, we decided not to use the full scale recommended by Hoffman et al., but only the items originating from the Cahour-Forzy scale. We did, however, use the modified version of the Cahour-Forzy items as suggested by Hoffman et al. The recommended modification only includes fitting the items on the 5-point Likert scale, in order to match the scaling used for the trust and explanation satisfaction scales. The resulting items that make up the trust scale are listed below. Similar to the understandability measurements, participants were asked to rate each item on a scale ranging from *strongly disagree* (1) to *strongly agree* (5).

1. I am confident in the forecasting dashboard. I feel that it works well.
2. The outputs of the forecasting dashboard are very predictable.
3. The forecasting dashboard is very reliable. I feel safe that I will get the right answers.
4. The forecasting dashboard is efficient in that it works very quickly.

Financial Case Studies
The second research objective stated in section 1.3 concerns the influence of XAI techniques on the effectiveness of Financial Forecasting solutions in terms of the insights gained. In section 2.1.2, we saw that improved predictive analytical capabilities in Finance, such as Financial Forecasting, support a more future focus, reduce the time spend on manual activities, improve the quality of data analytics, and promote the acquisition of insights. These benefits, in turn, are expected to support the successful implementation of FBP and improve the decision-making process. Therefore, we expect the use of Financial Forecasting solutions to improve the derivation of insights and the overall decision-making process for Finance Functions, whether FBP is the goal or not. To test this assumption, the following hypotheses were derived in 1.3:

H2. XAI increases the effectiveness of Financial Forecasting solutions by improving the derivation of insights by..

H2a. increasing the quality of the insights gained.

H2b. increasing the efficiency with which insights are gained.

In order to measure the effect of XAI techniques on the derivation of insights, two case studies were developed. The case studies involve a decision-making problem regarding which drivers to change in order to timely influence the forecast in the right direction. The first case study is to be made using the Financial Forecasting solution without XAI. During the second case study, participants are presented the Financial Forecasting solution with XAI techniques. This way we can measure whether the XAI techniques contribute to better insights on which drivers 'drive' a forecast and how to optimally influence the predicted outcome. These two case studies have to be very similar but not identical. If the case studies differ too much from each other, this could lead to biased results that obstruct a fair comparison. Case studies that are identical, on the other hand, could also bias the results. This is because participants would then already have had time to think about the case study during the first experiment round, and as a result of this provide better answers in the second round. To balance these two requirements, we developed two identical case studies that concern two different financial items.

- **Case Study 1:** *You signaled a significant deviation between the planned Sales, General & Administrative expenses (SG&A) for May 2021 of €50.000 and the forecasted SG&A for that same month. The Business has asked you to come up with a scenario to address* *this deviation in order to help them meet their plan. The SG&A is influenced by a number of drivers. You're asked to provide your scenario by specifying which drivers you would change, in which direction you would change them and what new value you would give them to meet the planned SG&A of €50.000.*

- **Case Study 2:** *You signaled a significant deviation between the planned Net Revenue for May 2021 of €205.000 and the forecasted Net Revenue for that same month. The Business has asked you to come up with a scenario to address this deviation in order to help them meet their plan. The Net Revenue is influenced by a number of drivers. You're asked to provide your scenario by specifying which drivers you would change, in which direction you would change them and what new value you would give them to meet the planned Net Revenue of €205.000.*

For both case studies, the participants were asked to answer the following questions:

1. Please indicate which drivers you would change in order to meet the planned *[concerned financial item]*.
2. Please indicate for each of the drivers you've selected whether you would increase or decrease their value in order to meet the planned *[concerned financial item]*.
3. Please provide for each of the drivers you've selected their new value (round number) that enables the business to meet the planned *[concerned financial item]*.

Explanation Satisfaction Scale

The third research objective investigated in the validation experiment is an exploratory one. Therefore, no hypotheses were derived to support this objective. As stated in section 1.4 and discussed in section 3.3, the XAI techniques implemented in our XAI prototype were selected on the basis of existing literature. Furthermore, they were tested from a technical perspective in terms of their suitability for Financial Forecasting models and data. However, we are also interested in their suitability according to the perspective of its target audience, the finance user. As discussed in section 2.4.1, the target audience of XAI is a key aspect in determining its effectiveness. This is in part investigated by the use of the trust and understandability measurements discussed above. However, these measurements concerns the XAI enabled Financial Forecasting solution as a whole and hence do not make a distinction between the individual XAI techniques selected. Therefore, we also incorporated a scale to measure the explanation satisfaction of the XAI techniques on a stand-alone basis. Hofmann et al. define explanation satisfaction as "a contextualized, a posteriori judgment of explanations, indicating the degree to which users feel that they understand the AI system or process being explained to them." [HMKL18]. Based on their literature review, they developed a scale to measure this explanation satisfaction. The scale was found to be valid based on a test amongst several XAI researchers during a DARPA-sponsored meeting. The resulting explanation satisfaction scale used in our validation are listed below. The same agreement scale as for understandability and trust was used with possible scores ranging from *strongly disagree* (1) to *strongly agree* (5).

1. From the [BD/VI/ALE] plot explanation, I understand how the forecasting dashboard works.
2. This [BD/VI/ALE] plot explanation of how the forecasting dashboard works has sufficient detail.
3. This [BD/VI/ALE] plot explanation of how the forecasting dashboard works tells me how to use it.
4. This [BD/VI/ALE] plot explanation of how the forecasting dashboard works is useful to my goals.
5. This [BD/VI/ALE] plot explanation of the forecasting dashboard shows me how accurate the forecasting dashboard is.
6. This [BD/VI/ALE] plot explanation lets me judge when I should trust and not trust the forecasting dashboard.

5.2 Experiment Results

In this section we will report the results obtained from the validation setup discussed above. We firstly discuss the characteristics of the sample group that partook in the research. Then, we elaborate on the results regarding the effect of XAI on understandability and trust. Next, we will look into the results of the case studies aimed at measuring the effect of XAI on the derivation of insights. Lastly, we will discuss the results of the explanation satisfaction measures for each of the individual techniques. The data discussed in this section was obtained using the online survey tool Qualtrics [Qua21]. The analysis was conducted in both Jasp and R. The statistically significant results for $\alpha = 0.05$ and $\alpha = 0.01$ are indicated with * and **, respectively.

5.2.1 Sample Group

A total number of 15 people participated in the research. 8 of the participants work at Deloitte and are consultants within the Finance domain. The other 7 participants are clients from Deloitte that work within the Finance Function of companies operating in the FSI. In order to get a clear picture of the level and type of experience within the Finance domain, we presented the participants a number of demographic questions. The majority of all participants has completed a Master's degree (80%). The remaining participants either have an associate degree (6.7%) or a doctorate degree (13.3%). The number of years experience working as a Finance professional differed greatly ($mean$=15.400, sd=10.370). The specific professions reported by these Finance professionals include Consultant (53.3%), (Business) Controller (26.7%), Business Developer (6.7%), Finance Business Partner (6.7%) and Risk & Finance Analyst (6.7%). Furthermore, participants were asked to indicate their level of experience with the financial planning and forecasting process, on a scale from 1 (*Not experienced at all*) to 5 (*Extremely experienced*). None of the participants indicated having no experience at all with the planning and forecasting process, 13.3% indicated being slightly experienced, 20% moderately experience, 53.3% very experienced and 13.3% identified as being extremely experienced. Lastly, we were interested in the experience participants have with Financial Forecasting solutions, both with and without Machine Learning. To this end, participants were asked to indicate their experience with both, on a scale from 1 (*Not experienced at all*) to 5 (*Extremely experienced*). It was found that participants had greater experience with Financial Forecasting solutions that do not incorporate ML (*mean*=3.200, *sd*=1.207), than with Financial Forecasting solutions that do (*mean*=2.467, *sd*=1.457). The exact distributions are given in Figures 5.1 and 5.2.

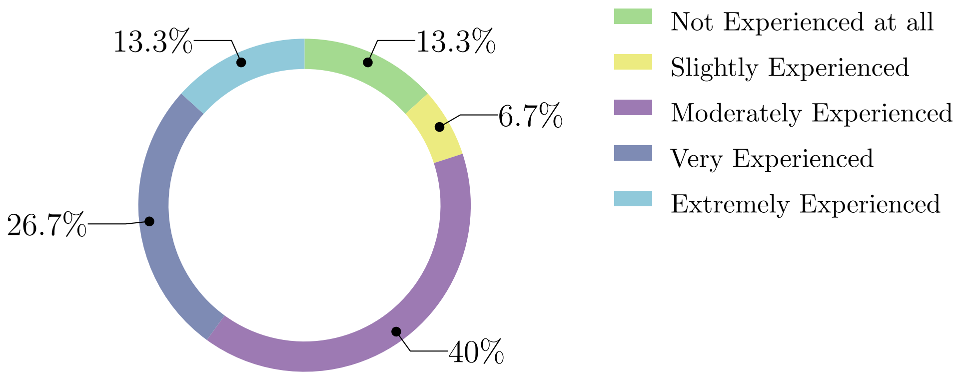


Figure 5.1: Participants' experience with Financial Forecasting solutions that do not use Machine Learning.

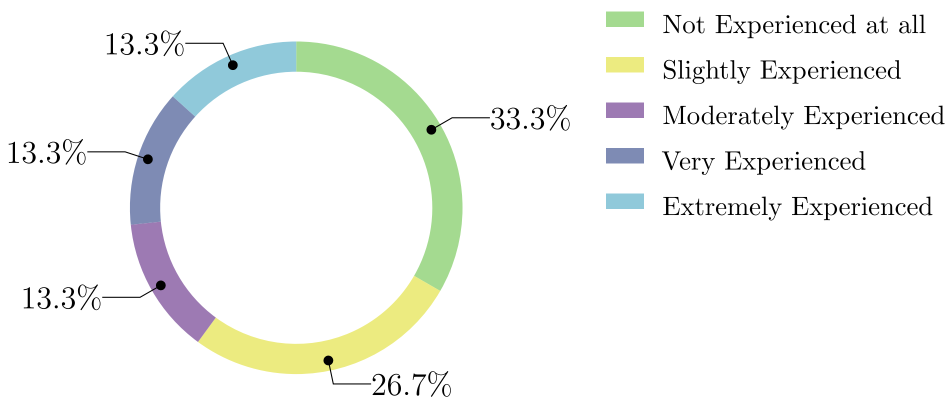
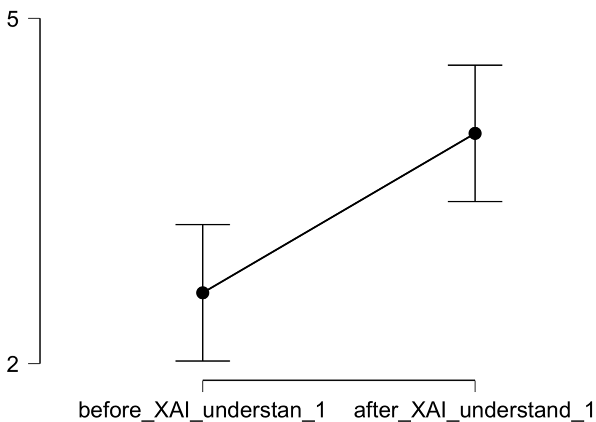


Figure 5.2: Participants' experience with Financial Forecasting solutions that do use Machine Learning.

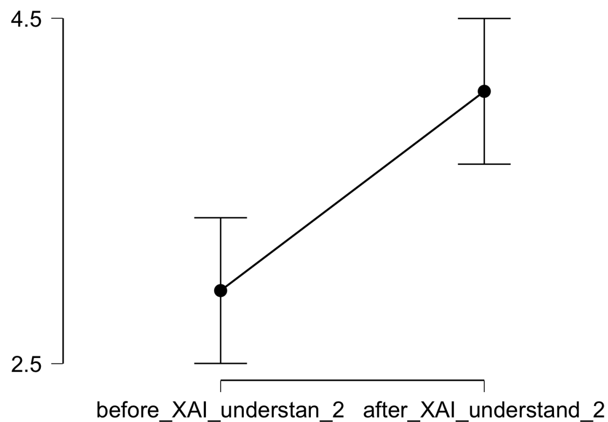
Before we can elaborate on the results of the validation experiment in the following sections, we have to briefly address concerns regarding outliers in the data. During data analysis of the case studies, two outliers were identified. One of these outliers skewed the results greatly, by providing a scenario for the case study with the use of XAI that deviated significantly from the target value compared to the other participants. Upon further examination of both the answers provided for this case study and the recording of the conduction of the case study, we found that the participant in question did not in any way make use of the available XAI techniques. Since the purpose of the case study was to measure the quality of the corresponding scenario analysis without and with the use of XAI techniques, the procedure applied by this particular participant obstructs this comparison. Therefore, it was decided to exclude this outlier from our analysis. Although the second outlier that not skew results as greatly as the previously discussed outlier, it was caused by the similar reason and hence as well obstructs a fair comparison. Therefore, this second outlier was also excluded from the analysis.

5.2.2 Effects on Understandability and Trust

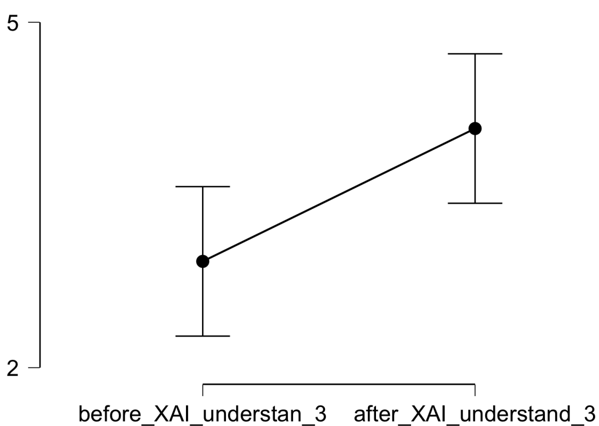
As explained in section 5.1.2, in order to test hypothesis **H1a**, we presented participants with five statements relating to their level of understandability of the Financial Forecasting solution without and with the use of XAI. For each of the understandability measurements we see an increase when participants had XAI techniques at their disposal, as illustrated in Figure 5.3. Namely, for each plot contained in Figure 5.3, we observe that the mean value for every understandability measurement is higher after the use of XAI than before the use of XAI. The error bars depicted in the plots, indicate the 95% confidence intervals of the means for the before and after measurements. It can be observed that only the confidence intervals for the third understandability measurement (Fig. 5.3c) display some overlap.



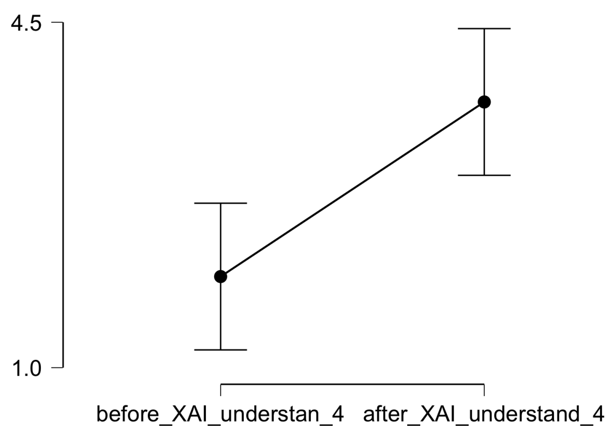
(a) *"I know what will happen the next time I use the system, because I understand how it behaves."*



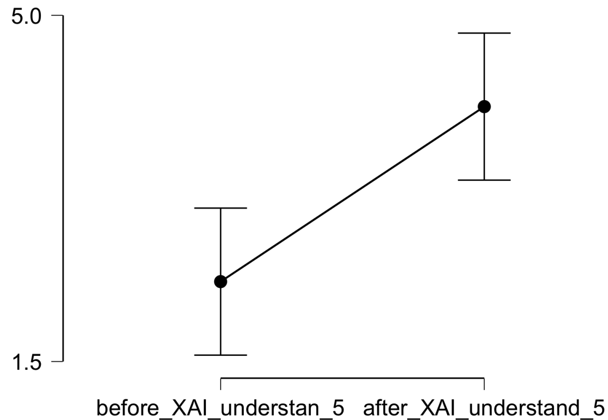
(b) *"I understand how the system will assist me with decisions I have to make."*



(c) *"Although I may not know exactly how the system works, I know how to use it to make decisions about the problem."*



(d) *"It is easy to follow what the system does."*



(e) *"I recognize what I should do to get the advice I need from the system the next time I use it."*

Figure 5.3: The means and error plots for the understandability measurements prior to and after the use of XAI.

With the use of XAI techniques, on average the understandability increases by 1.385 (se=0.385), 1.154 (se=0.274), 1.154 (se=0.421), 1.769 (se=0.482) and 1.769 (se=0.482), for the five measurements respectively. The overall increase in understandability is 1.446 (se=0.328). In order to investigate the significance of these increases, we performed a paired samples t-test. Prior to performing this test, we first verified whether the assumption of normality was not violated through means of the Shapiro-Wilk test. The results of this test, given in Figure 5.2, show that for all pairwise differences, except for the fifth measurement ($p = 0.002$), results are not significant and hence normally distributed. The paired samples t-test performed here, concerns an one-tailed t-test. The upper-tailed alternative hypothesis that was used is as follows:

$$H_1 = \mu_1 < \mu_2 \ (d > 0). \tag{5.1}$$

The results of the t-tests in Figure 5.1 indicate the increase in understandability to be significant for each measurement ($p = .002, < .001, 0.009, 0.002, 0.002$, respectively), which results in the acceptance of hypothesis $H1a$. However, as the fifth understandability measurement violated the normality of distribution assumption, a second non-parametric test was performed, namely the Wilcoxon's signed-rank test. From the Wilcoxon's test, it can be concluded that the increase in understandability is significant for the fifth measurement as well ($W = 69, p = 0.009$). Furthermore, Cohen's d suggests that the observed effect of XAI techniques on understandability is large for measurements 1, 2, 4 and 5 ($Cohen's\ d > 0.8$) and medium for the third measurement ($Cohen's\ d > 0.5$).

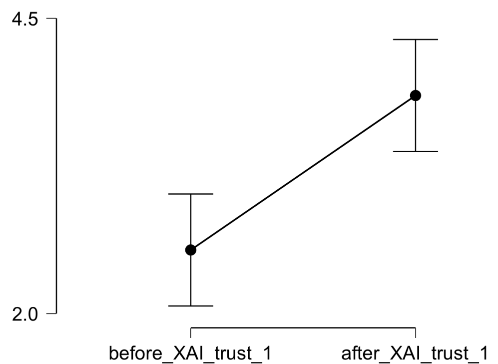
After XAI	Before XAI	t	df	p	Mean Difference	SE Difference	Cohen's d
understand_1	understand_1	3.600	12	.002**	1.385	0.385	0.998
understand_2	understand_2	4.215	12	< .001**	1.154	0.274	1.169
understand_3	understand_3	2.739	12	.009**	1.154	0.421	0.760
understand_4	understand_4	3.667	12	.002**	1.769	0.482	1.017
understand_5	understand_5	3.667	12	.002**	1.769	0.482	1.017

Table 5.1: Results of paired samples t-test on understandability.

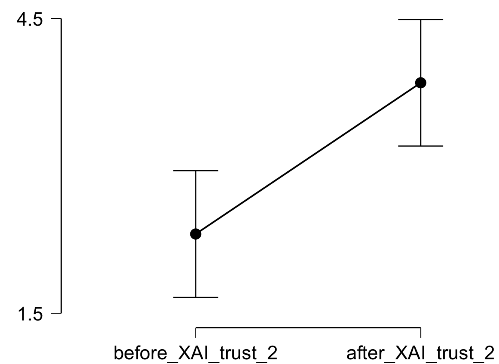
			W	p
after_XAI_understand_1	-	before_XAI_understan_1	.893	.109
after_XAI_understand_2	-	before_XAI_understan_2	.879	.070
after_XAI_understand_3	-	before_XAI_understan_3	.902	.142
after_XAI_understand_4	-	before_XAI_understan_4	.936	.409
after_XAI_understand_5	-	before_XAI_understan_5	.745	.002**

Table 5.2: Results for test of normality (Shapiro-Wilk) on understandability measurements.

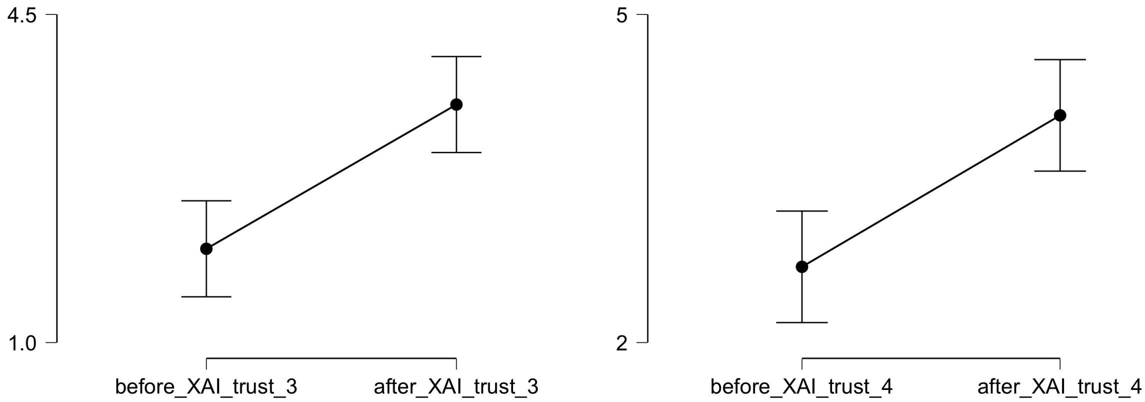
Hypothesis **H1b**, concerning the influence of XAI on the level of trust in Financial Forecasting solutions, was verified in a similar way as **H1a**. From Figure 5.4 it can be seen that, for all four trust measurements discussed in section 5.1.2, trust was higher in the presence of XAI techniques. Furthermore, no overlap is observed in the confidence intervals of the trust measurements prior to and after the use of XAI.



(a) *"I am confident in the forecasting dashboard. I feel that it works well."*



(b) *"The outputs of the forecasting dashboard are very predictable."*



(c) *"The forecasting dashboard is very reliable. I feel safe that I will get the right answers."* (d) *"The forecasting dashboard is efficient in that it works very quickly."*

Figure 5.4: The means and error plots for the trust measurements prior to and after the use of XAI.

With the use of XAI techniques, the level of trust on average increases by 1.308 (se=0.308), 1.538 (se=0.418), 1.538 (se=0.332) and 1.385 (se=0.331), for the four measurements respectively. The overall increase in understandability is 1.442 (se=0.247). Prior to performing the paired samples t-test, we again firstly checked whether the assumption concerning normality of distribution is satisfied. The results for the Shapiro-Wilk test, shown in Figure 5.4, indicate normal distributions for all four pairwise differences. The results for the one-tailed t-test, given in Figure 5.3, were derived based on the alternative hypothesis stated in 5.1. From the paired samples t-test it can be concluded that the increase in the level of trust is significant for all four trust measurements ($p =< .001, .002, < .001, < .001$, respectively). This leads us to accept hypothesis **H1b**, stating that the use of XAI techniques increase the level of trust in Financial Forecasting solutions. Furthermore, we conclude that the observed effect of XAI on the level of trust is large for all trust measurements, as indicated by Cohen's d.

After XAI	**Before XAI**	**t**	**df**	**p**	**Mean Difference**	**SE Difference**	**Cohen's d**
trust_1	trust_1	4.250	12	< .001**	1.308	0.308	1.179
trust_2	trust_2	3.682	12	.002**	1.538	0.418	1.021
trust_3	trust_3	4.629	12	< .001**	1.538	0.332	1.284
trust_4	trust_4	4.185	12	< .001**	1.385	0.331	1.161

Table 5.3: Results of paired samples t-test on trust.

			W	p
after_XAI_trust_1	-	before_XAI_trust_1	.908	.175
after_XAI_trust_2	-	before_XAI_trust_2	.946	.544
after_XAI_trust_3	-	before_XAI_trust_3	.909	.178
after_XAI_trust_4	-	before_XAI_trust_4	.892	.105

Table 5.4: Results for test of normality (Shapiro-Wilk) on trust measurements.

5.2.3 Effects on the Derivation of Insights

The second question addressed in this research, concerning the impact of XAI on the insights gained, was tested by means of the case studies discussed in section 5.1.2. For both case studies, we collected the new scenario participants developed in order to meet the planned values that were specified. Furthermore, we measured the time it took participants to finish the case study when using the Financial Forecasting solution without and with the use of XAI.

To test hypothesis **H2a**, the effect of XAI on the quality of insights, we measured the quality of the scenario analyzes that were provided for both case studies. In order to do so, computed the so-called closeness utility that indicates in percentages how close a scenario is to the planned value. To this end, we computed the forecasted value based on the selected drivers and corresponding driver values that participants provided for their scenarios. Hence, for both the SG&A (case study 1) and the net revenue (case study 2), we fed the participants' scenarios into the corresponding forecast models developed by PrecisionView™. Next, we computed the difference between the forecast obtained by their scenario and the specified planned value in percentages to obtain the closeness utility.

$$before\_XAI\_closeness\_utility = 100\% - \frac{abs(50.000 - forecast_{scenario\_1})}{50.000}) \times 100\% \tag{5.2}$$

$$after\_XAI\_closeness\_utility = 100\% - \frac{abs(205.000 - forecast_{scenario\_2})}{205.000} \times 100\% \tag{5.3}$$

Figure 5.5 shows the mean values and error bars for both the closeness utility prior to and after the use of XAI techniques. It can be observed that the closeness utility for the case study with XAI (*mean*=79.040, *sd*=19.107) is higher than the closeness utility for the case study without XAI (*mean*=68.959, *sd*=8.393). However, both the error plots in Figure 5.5, as well as the standard deviation of the closeness utility obtained with the use of XAI, show a fair amount of overlap. The overlap in utility closeness becomes more clear when looking at the box plots in Figure 5.6. The distribution range for utility closeness scores obtained with the use of XAI techniques is much greater (*min*=45.618, *max*=98.104) than the range of utility closeness scores without the use of XAI techniques (*min*=58.933, *max*=83.761). We observe from the box plots that nearly 25% of the utility closeness scores obtained with XAI techniques, is lower than all scores obtained without. On the other hand, over 50% of closeness utility scores with XAI are higher than all utility scores obtained without XAI. Hence, nearly 25% of the utility closeness for case studies performed with the use of XAI, overlap with the utility closeness scores of the case studies conducted without

XAI. When we analyse the change in utility closeness on an individual level, it appears that the greater distribution range for utility closeness with the use of XAI is caused by an underlying by-modal distribution. Although the majority of the participants experienced an increase in closeness utility with the use of XAI, four out of the thirteen participants experienced a decrease in utility closeness. The average change in utility closeness of participants for which an increase was observed is 22.703, whereas the average change for participants experiencing a decrease was −18.202. This explains the increase in the range of values of the utility closeness with the use of XAI.

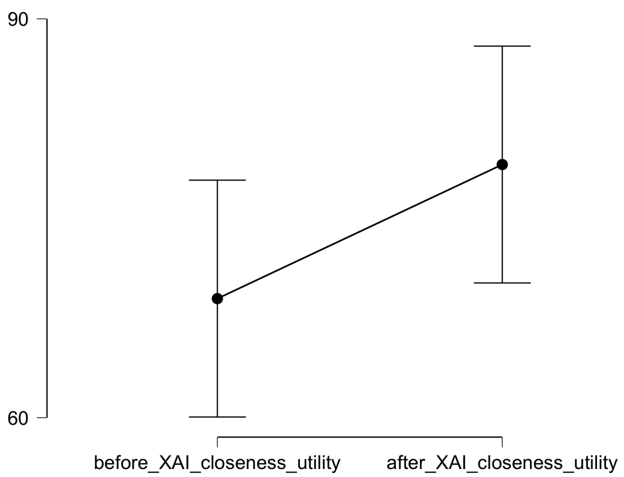


Figure 5.5: The means and error plots for the utility closeness prior to and after the use of XAI.

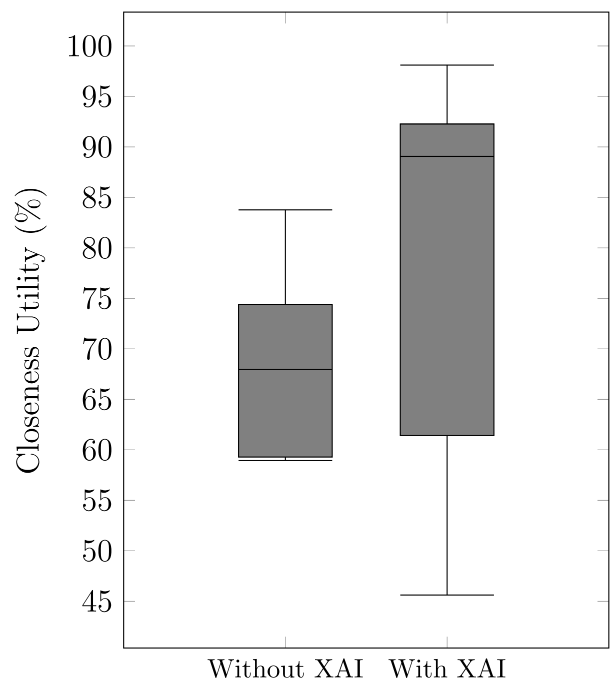


Figure 5.6: The box plots for the utility closeness without and with the use of XAI.

In Figures 5.5 and 5.6, we observed that despite the overlap, there is an increase in the closeness utility after the use of XAI techniques. In order to determine whether this increase is significant, we again perform a paired samples t-test. Similar to the t-tests compared for the trust and understandability measurements, we performed an upper-tailed t-test using the alternative hypothesis given in equation 5.1. However, the Shapiro-Wilk test shown in Table 5.5, indicated that the normal distribution assumption for the pairwise difference of the utility closeness measures was violated ($p = 0.029$). Therefore, a Wilcoxon signed-rank t-test was performed to check for significance. The results are presented in Table 5.6. The Wilcoxon test showed that the increase in closeness utility is non-significant ($p = .095$). This means that we cannot accept hypothesis **H2a**, stating that the use of XAI techniques increase the quality of insights gained for Financial Forecasting solutions. However, it has to be noted that with small sample sizes, the changes of finding statistically significant results are unlikely unless the effect size is large [Coe02]. Therefore, it is important that we also look at the effect size. Since the pairwise differences for the utility closeness are not normally distributed, another measure for the computation of effect size was used, namely the

Rank-Biserial Correlation (Table 5.6). From the effect size we observe that there is a small effect of the use of XAI techniques on utility closeness (*Rank-Biserial Correlation* > 0.2).

			W	p
after_XAI_closeness_utility	-	before_XAI_closeness_utility	.851	.029*

Table 5.5: Results for test of normality (Shapiro-Wilk) on utility closeness.

After XAI	Before XAI	W	p	Hodges-Lehmann Estimate	Rank-Biserial Correlation
closeness_utility	closeness_utility	65.000	.095	10.902	0.429

Table 5.6: Results of Wilcoxon signed-rank test on the closeness utility obtained from the case studies.

To test hypothesis **H2b**, the effect of XAI on the efficiency with which insights are gained, we analyzed the time it took participants to finish the case study without and with the use of XAI techniques. The expectation underlying **H2b** is that if XAI techniques increase the efficiency with which insights are gained, then we should observe a decrease in the amount of time it took participants to complete the case study when they have XAI techniques at their disposal. However, from figure 5.7 we observe that, on average, the completion time in minutes is greater with the use of XAI technique (*mean*=16.385, *sd*=6.838), than without (*mean*=14.769, *sd*=4.885). On average, it took participants roughly $14\frac{3}{4}$ minutes to complete the case study without the use of XAI, and $16\frac{1}{3}$ minutes with the use of XAI. Furthermore, the figure shows a fair amount of overlap in the completion times for both case studies. The box plots in Figure 5.8 provide more insights into this overlap. It can be seen that both the shortest completion time, as well as the longest, were measured during the conduction of the case study with the use of XAI. Interestingly, the participants for which we observed a decrease in utility closeness more often showed an increase in the measured completion time. More specifically, 75% of participants that obtained a lower utility closeness with the use of XAI, also required more time to complete the scenario analysis with the use of XAI. However, of the participants for which we observed an increase in utility closeness with the use of XAI, only 33.3% showed an increase in completion time.

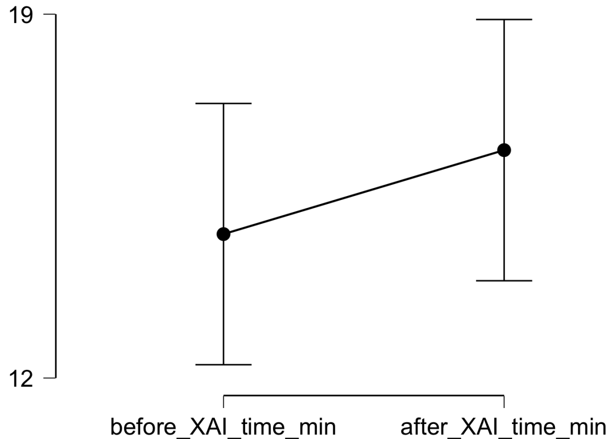


Figure 5.7: The means and error plots for the completion time prior without and with the use of XAI.

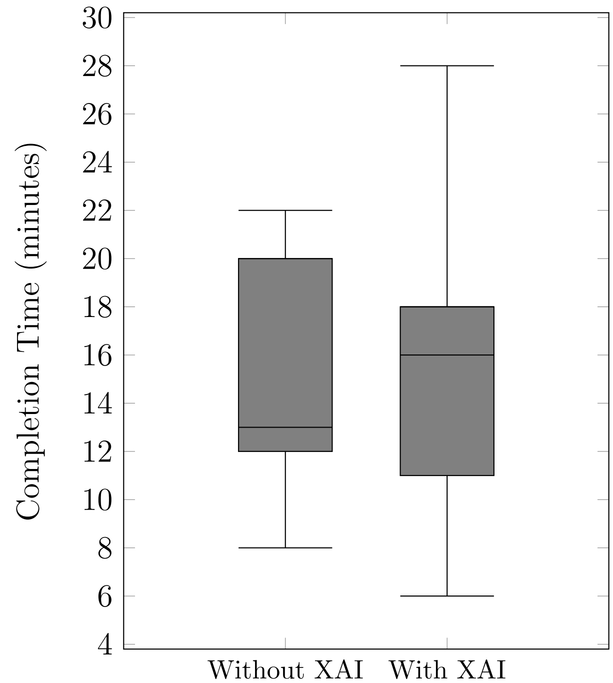


Figure 5.8: The box plots for the completion time without and with the use of XAI.

From the mean value and box plot for both completion times discussed above, we can already observe that there is no significant negative correlation between the use of XAI techniques and the time it took participants to perform complete a case study. Rather, we are seeing an increase in the completion time. Therefore, we did test for significance effects of the use of XAI techniques in the opposite direction. To this end we performed a upper-tailed t-test using the alternative hypothesis given in equation 5.1. The results are given in Table 5.7. From the results we observe the increase in completion time to also be non-significant ($p = .171$). Therefore, we conclude that we can not accept nor reject hypothesis **H2b**, stating that the use of XAI techniques increases the efficiency with which insights are gained.

After XAI	**Before XAI**	**t**	**df**	**p**	**Mean Difference**	**SE Difference**	**Cohen's d**
after_XAI_time_min	before_XAI_time_min	0.990	12	.171	1.615	1.631	0.275

Table 5.7: Results of paired samples t-test on completion time.

5.2.4 Evaluation of Individual XAI Techniques

The third question addressed in this research concerned the effectiveness of the individual XAI techniques from the perspective of the target audience, i.e. which XAI techniques did Finance users found to be most satisfying to their goals? As explained in section 5.1.2, this question is of exploratory nature and hence no hypothesis were derived. The explanation satisfaction measurements stated in 5.1.2 were measured on a Likert scale ranging from 1 (*strongly disagree*) to 5 (*strongly agree*). Figure 5.9 shows the box plots for the average explanation satisfaction of the three individual XAI techniques. From the box plots we observe that the average satisfaction scores for the different XAI techniques roughly fall within the same range. 75% of participants rated their satisfaction with each of the three XAI plots between the 3.5 and 5.0. Furthermore, none of the participants rated their average satisfaction with the ALE below 3.0. 50% of participants rated their average satisfaction with the ALE plot above 4.0.

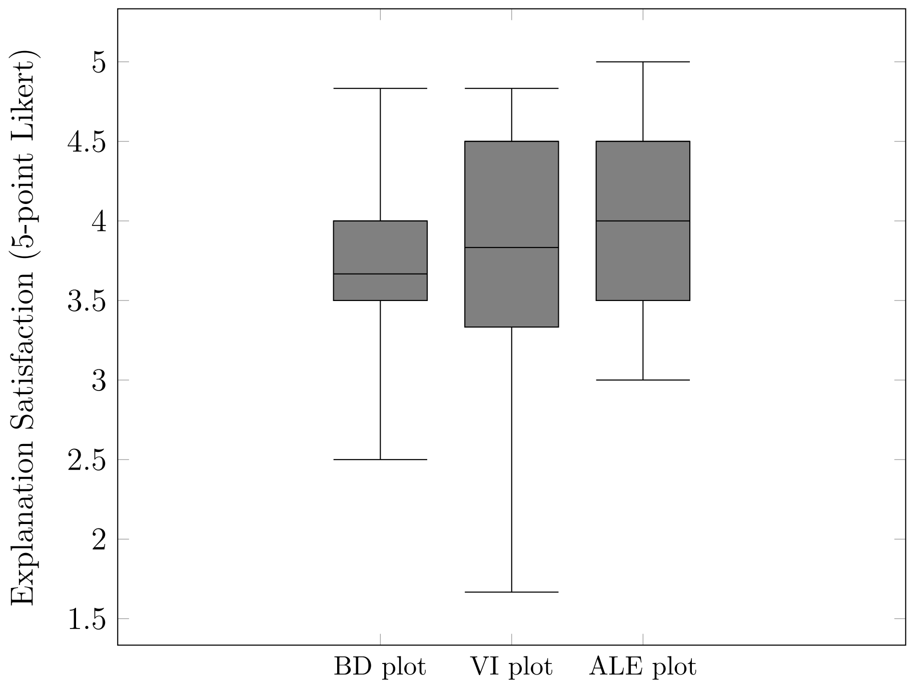


Figure 5.9: The box plots for the average explanation satisfaction of the BD, VI and ALE plot.

From the box plots we conclude that each individual XAI technique, on average, scores positive on explanation satisfaction. Furthermore, although the satisfaction scores for the individual XAI techniques are close to each other, the ALE plot appears to score slightly better. This becomes more clear when looking at mean values for the individual XAI plots in Table 5.8. The overall satisfaction score for the ALE plot is the highest (*mean*=3.974, *sd*=0.593), followed by the VI plot (*mean*=3.833, *sd*=0.915) and BD plot (*mean*=3.769, *sd*=0.618). The ALE plot scores highest on having sufficient detail (statement 2) and telling the user how to use the forecasting dashboard (statement 3). The VI plot obtained the highest scores when considering its ability to show how accurate the forecasting dashboard is (statement 5) and letting a user judge when he/she should trust and not trust the dashboard (statement 6). Lastly, the BD plot scores highest in supporting the user to understand how the forecasting dashboard works (statement 1) and in being useful to the user's goals (statement 4). Furthermore, the only score indicating dissatisfaction was given to the BD plot with regards to its ability to letting a user judge when he/she should trust and not trust the dashboard (statement 6)

	BD plot		VI plot		ALE plot	
	mean	**sd**	**mean**	**sd**	**mean**	**sd**
"From the ... plot explanation, I understand how the forecasting dashboard works."	4.385	0.650	4.154	0.987	4.308	0.630
"This ... plot explanation of how the forecasting dashboard works has sufficient detail."	4.000	0.913	3.846	1.214	4.231	0.832
"This ... plot explanation of how the forecasting dashboard works tells me how to use it."	4.154	0.555	4.000	1.155	4.154	0.689
"This ... plot explanation of how the forecasting dashboard works is useful to my goals."	4.538	0.519	4.077	1.038	4.462	0.660
"This ... plot explanation of the forecasting dashboard shows me how accurate the forecasting dashboard is."	3.000	1.528	3.462	1.266	3.385	1.193
"This ... plot explanation lets me judge when I should trust and not trust the forecasting dashboard."	2.538	1.391	3.462	1.266	3.308	1.109
	3.769	0.618	3.833	0.915	3.974	0.593

Table 5.8: The mean and standard deviation for all explanation satisfaction measurements, for each XAI plot.

5.3 Discussion

In the previous section we presented the results of the validation experiment conducted in this research. In this section, we will discuss their meaning and implications in relation to our research questions, but also what the results currently can not tell us.

Firstly, the results presented in section 5.2.2 showed an increase in the level of understandability of Financial Forecasting solutions in the presence of XAI techniques. The students t-test and Wilcoxon signed-rank test indicated this increase in understandability to be significant. Furthermore, the effect caused by the use of XAI techniques was found to be very strong (*Cohen's* $d = 1.222$). Similar results were found for the effect of XAI techniques on the level of trust in Financial Forecasting solutions. All measures of trust increased in the presence of XAI and all increases were found to be significant by the conducted student t-tests. Again, the size of the effect of XAI on the four measured levels of trust, as well as on the overall trust level (*Cohen's* $d = 1.619$) were found to be large. Furthermore, when we place the overall understandability and trust scores for the Financial Forecasting dashboards without and with XAI, into the context of the measurement scale that was used, another interesting observation arrises. As explained in section 5.1.2, participants were asked to rate each understandability and trust measurement on a scale from 1 (*strongly disagree*) to 5 (*strongly agree*). Therefore, the average increases of 1.446 and 1.442 in understandability and trust respectively, can not simply be interpreted as a 1.446 increase in understandability and 1.442 increase in trust. Rather, these average increases indicate a flip on both the understandability and trust scales, as depicted in Figures 5.10 and 5.11. When placed in context of the scales upon which understandability and trust were measured, we observe that the increases for both results going from a lack of understandability and trust to the presence of understandability and trust. In other words, not only do we observe a significant increase in understandability and trust caused by a strong effect of the use of XAI, but this effect also causes the users' distrust and non-understanding to convert to trust and understanding. These findings lead to the acceptance of hypothesis **H1a** (*XAI increases the understandability of Financial Forecasting solutions*) and **H1b** (*XAI increases the trust in Financial Forecasting solutions*). The acceptance of these two underlying hypothesis also results in the acceptance of its parent hypothesis **H1**, stating that the use of XAI helps overcome the black-box problem in the context of Financial Forecasting.

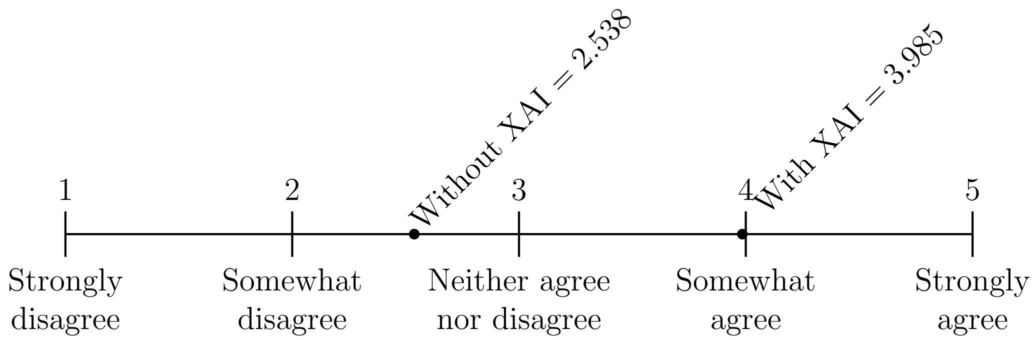


Figure 5.10: The overall understandability score without and with the use of XAI, in the context of the measurement scale.

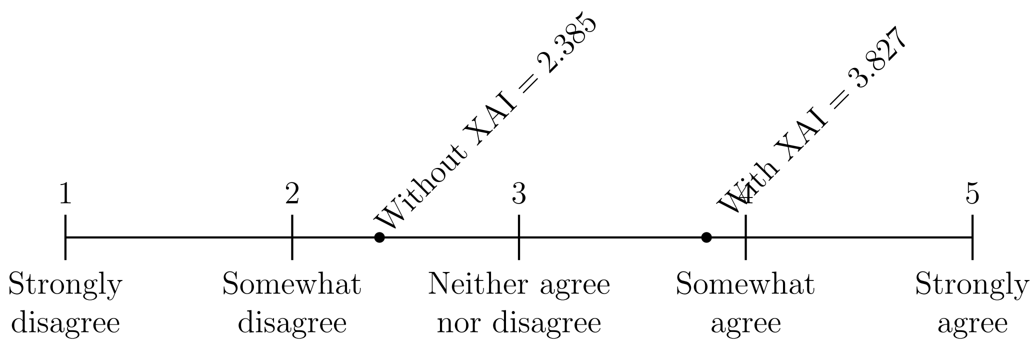


Figure 5.11: The overall trust score without and with the use of XAI, in the context of the measurement scale.

The results of the case studies presented in section 5.2.3 were aimed at testing hypothesis **H2**, stating that the use of XAI increases the effectiveness of Financial Forecasting solutions by improving the derivation of insights. This improvement in the derivation of insights was, in turn, tested by measuring the increase in quality of the insights gained (**H2a**) and the increase in efficiency with which the insights were gained (**H2b**). Although we found an increase in the quality of insights through means of the measured utility closeness, the increase was not found to be significant. Therefore, we failed to accept hypothesis **H2a**. Secondly, rather than finding a decrease in the completion time for the case studies with XAI, we found a slight increase instead. Therefore, no evidence was found to support our assumption that the use of XAI techniques increases the efficiency with which insights are gained, and hence the corresponding hypothesis **H2b** could neither be accepted. The failure to accept these to sub-hypotheses also means that no support was found for the parent hypothesis **H2**.

However, an important consideration should be discussed before completely disregarding the possible effects XAI can have on the improvement of insights. The literature describes a relation between the amount of time users have worked with XAI, and the positive benefits of XAI that can be observed [HMKL18]. It requires time for participants to become acquainted with the XAI techniques, learn how to use them, and generate enough trust to rely upon the information they provide. Our results support this view, as we are seeing an increase in the amount of time required to conduct a scenario analysis with the use of XAI, rather than a decrease. This indicates that it indeed requires more time from the user to understand and interpret the different XAI techniques. This finding is not only supported by the observed increase in completion time, but also by the manner in which the case study with XAI was conducted. In the recordings of case studies with XAI, several participants commented on this, stating: *"I quickly looked at the BD plot, but ignored it in my analysis. Now, afterwards, I realize that it is actually quite insightful.", "I need more time to interpret the BD plot.", "I do not really understand the BD plot yet, could you explain how it should be interpreted?", "How do I need to interpret the ALE plot?", "I do not understand the ALE plot, why are both expenses and revenues positively correlated with the prediction?"*.

This finding explains why we observed an increase in the completion time measured for case studies that were conducted with the use of XAI. However, more importantly it could have influenced the observed effect of XAI on the quality of the insights derived. Although a positive effect of XAI on the quality of insights was found, it was not large enough to be significant. This might be due to the users' lack of experience with the XAI techniques, and hence this effect could potentially increase further when users become more acquainted with the techniques. This can, for example, be done by replicating the experiment design used in this research and perform multiple cycles of the experiment. It is important that the group of participants partaking in these cycles remains the same, in order to measure the effect of XAI on the quality of insights derived for the same participant over multiple cycles. In addition to performing multiple cycles, one might consider replacing the video demonstration with a live demonstration. This enables participants to ask questions in case of uncertainties and allows the researcher to validate whether participants' interpretation of XAI techniques is indeed correct. We can only speculate on the impact of such research on the effect of the quality of insights derived. However, we expect that it will lead to a smaller increase in the amount of time required to conduct the scenario analysis, and eventually even a decrease. This would result in the acceptance of hypothesis **H2a**. More importantly, we expect such an experiment in which participants become more acquainted with the XAI techniques, to lead to an increase in the quality of insights gained that is significant and hence lead to the acceptance of hypothesis **H2b**.

The last set of results presented in section 5.2.4 was aimed at exploring the users' level of satisfaction with each of the individual XAI techniques. Firstly, the results tell us that all three XAI techniques implemented in our prototype were deemed satisfying in terms of the degree to which it enabled users to understand the system being explained to them, the Financial Forecasting solution. The ALE plot obtained the highest overall satisfaction score, followed by the VI and BD plot, respectively. The three techniques were al roughly ranked with a 4, indicating that participants on average *"somewhat agreed"* to each satisfaction measure. This indicates that there is still room for improvement. Based on the individual satisfaction scores for each of the XAI techniques and the open feedback provided by participants, we gathered several suggestions for improvement.

ALE plots

Amongst all six satisfaction measurements, the ALE plot obtained its lowest scores on the fifth and sixth measurement, indicating the ALE plot's ability to show how accurate the forecasting dashboard is and letting the user judge when he or she should trust/not trust the dashboard, receptively. There were two feedback comments in particular that indicate potential sources of this lack of accuracy information and distrust.

1. *"Yes, the ALE plot provides an indication of what to do, but it's a bit unclear if there are any drawback on the drivers. Can you just increase them to the max.? What will go wrong, or what not?"*

2. *"I like the ALE plot and the concept behind it. There is no accuracy information to build confidence here [ALE plot] that the curves/shapes are correct, or the relative confidence I should have in them, however."*

The comments of both participants concern the shape of the graphs depicted in the ALE plot. In line with this, the remaining feedback primarily addressed the layout of the graph. Participants state that *"The ALE plot is too small: it provides some challenges to hover over the right data points.", "The line could be extended for areas where details are not available, showing different colors and indicating it is a forecast on the forecast of data."* and question *"Why does the graph have a wiggly line? What does it mean?"*. Based on these comments, possible improvements could include:

- Either smoothing the lines of the ALE plot or provide more detailed information on how the lines are computed in order to explain why lines can become wobbly.
- Enlarging the view of the ALE plot and lower the step size for which the values of each graph in the ALE plot are obtained.
- Increase the x range of the graphs in the ALE plot to also include values not in the data set, in order to extend the graph lines beyond what is observed from the data. This could be done by extracting the formula behind the graphs in the ALE plotted computed on the original data set, and extrapolating them to include values beyond the original range.

Lastly, one interesting comment concerned the isolated view of the graphs in the ALE plot. The participant in question stated: *"I like this chart and the concept behind it. It would be a bit more real-world useful if I could look at the changes in the individual charts together as multiple drivers are modified (think: scenario modeling). Once I understood the charts, it was probably some of the most useful/actionable information, the only downside is that drivers are isolated."* In fact, during the conduction of the case studies other participants commented on this as well, indicating their uncertainty with viewing the effect of the drivers in isolation. Therefore, we expect improvements to the ALE plot that enable users to investigate the joint effect of drivers, rather than the isolated effect, on the average prediction to potentially have a significant impact on the overall satisfaction with ALE plot.

VI plots
The VI plot also obtained its lowest satisfaction scores for measures 5 and 6. However, compared to the other two plots it ranked highest on these two measurements. This is indeed what would be expected, as the VI plot is the only XAI plot that is computed using accuracy measurements, namely the MAPE in this case. On the remaining four satisfaction measurements, the VI plot scores lower that both the ALE and BD plot. The lowest score was obtained for measurement 2, stating whether the VI plot contained sufficient detail or not. The majority of the feedback given on the VI plot indeed addresses additional information participants would like to see in the plot. Participants suggested the following:

- Including information on the interaction amongst drivers: (*".. However, there are still some black box elements, especially for the 'date' driver I would really need to understand the relationship with the other drivers.", "It would be interesting to show the interaction between drivers, if it exists, and how they integrate with each other."*)
- Including background information on the period over which the importance was computed: *"I find this chart to be too abstract and singular in purpose to be helpful. It is an aggregate/average view without clearly defined parameters, and is not clear how this would help me make decisions, or what the range of variance is, or how the importance might change by period or be changing over time (trending). All which might be more helpful use of the space."*
- Separating drivers based on certain characteristics: (*"Separate drivers that can be influenced from drivers that can not be influenced."*).

BD plots
The BD plot also obtained the lowest scores for the fifth and sixth measurement, both in comparison with the scores for the other two plots and in comparison with its scores on the other satisfaction measures. From the gathered feedback it appears that the distrust primarily stems from the lack of information on the joint effect of the drivers. Two comments in particular addressed this concern, stating that:

1. *"It is a bit difficult to understand the full implications of parameters on the model. Based on advertising expenses you would always want to spend the maximum - as that seems to generate maximum revenues - but how about patterns and such - how do these behave?"*
2. *"I like the BD plot, if I already trust the model is accurate. It could be useful if additional breakdown detail or some indication of the interrelationship between multiple drivers were available. It does not give me any facts to assess accuracy and whether I should or should not trust the outcomes (on its own)."*

Furthermore, participants indicated that they would like the BD plot to have more information that assists them in their decision making. Two suggested were made to achieve this objective:

- Separating drivers based on certain characteristics: (*"I think the idea is very good. Ofcourse, the difficulty is always with AI and regression not to find regression, but to find drivers you can influence. Distinction in those would be desired, i.e. to have management have a clear focus on what to steer and those drivers that have value for your storyline.",* *"Sort graph differently by making a split between negative and positive and including the subtotal after all negative + subtotal after all positive."*)

- Including a functionality that provides recommended changes to the user: ("*A way to request recommendations on how to adjust the forecast, according to what the system knows of itself. A chatbot interacting with the user would be amazing."*)

6 Conclusion

Artificial Intelligence and its subfield Machine Learning have proven to be powerful technologies that can assist in cost and time reduction, improve safety and accuracy, and make part of the jobs still performed by humans more valuable and satisfying. However, the successful implementation of ML is often obstructed due to ML models being perceived as black boxes. Research into the field of Explainable Artificial Intelligence is aimed at overcoming this black-box issue by increasing the understandability of, and trust in these models. However, until now the use of XAI in the specific context of Financial Forecasting has been left uninvestigated. Yet, investigating XAI in this specific context is important for a number of reasons. Firstly, predictive analytics capabilities, such as Financial Forecasting solutions, are said to be key for the future of the Finance Function. They are thought to reduce manual activities, support a more forward-looking approach, improve the quality of data analytics, promote the acquisition of insights and thereby ultimately improve the overall decision-making process. Hence, overcoming trust and understandability issues to successfully implement such Financial Forecasting solutions can have great benefits for the Finance Function. Secondly, the validation of XAI techniques is dependent on the context in which it is used, both from a technical perspective, as well as from the perspective of the target audience of XAI. From a technical perspective, the context in which XAI is implemented, determines the type of ML models used and hence also the XAI techniques potentially suitable for those models. From the perspective of the target audience, the context in which XAI is implemented largely determines the desiderata for the explanations generated by XAI, as well as whether they are deemed successful at explaining an AI system. Therefore, in this research we investigated the effectiveness of XAI for Financial Forecasting solutions. To this end, we developed an XAI prototype that is implemented in an existing Financial Forecasting solution, called PrecisionView™. The reason for the development of this prototype was three-fold. Firstly, it allowed for the verification of the applicability of different XAI techniques for models commonly used in Financial Forecasting, from a technical perspective. Secondly, the proposed prototype was aimed at providing the reader with a generic approach for the implementation of XAI techniques on any forecasting solution within the Finance domain. Lastly, and most importantly, the prototype formed the basis for the conducted experiment in which we tested the impact of XAI on both the black-box problem and the derivation of insights.

For the investigation of the applicability of XAI techniques from a technical perspective, we found that the selected XAI techniques work for all three types of ML models tested, namely multi-linear regression, ARIMAX and Prophet with regressors. However, we did discover peculiarities when applying the Break-down method to SARIMAX models. This is remarkable, since the BD method is said to be model-agnostic, meaning it should be applicable for any type of model. We will discuss this in more detail in section 6.2.

The results of the conducted experiment showed that XAI successfully improves financials' trust in and understandability of Machine Learning driven forecasts. More specifically, it was found that the overall level of trust increased, on average, from 2.385 to 3.827 on a 5-point Likert scale. The level of of understandability increased, on average, from 2.538 to 3.985. This significant increase in both trust and understandability contributes to overcoming the black-box issue and supports the successful implementation of Financial Forecasting solutions. Furthermore, we found an effect between the use of XAI and the acquisition of insights. The closeness utility score obtained for case studies performed with the use of XAI, on average, is 10.08% higher than those obtained for case studies performed without the use of XAI. Although the positive effect observed in our experiment was not strong enough to indicate significance, our expectation is that with more research and further improvements, this positive effect observed has the potential to increase further. More specifically, as discussed in section 5.3, we expect that the conduction of a similar experiment involving multiple experiment rounds and more extensive training, will lead to a stronger increase in the quality of the insights gained, as well as a decrease in the amount of time required to conduct the scenario analysis described. If this increase in the quality of insights gained and decrease in the amount of time required to conduct a scenario analysis reach significant levels, this would result in the acceptance of both hypothesis **H2a** and **H2b**, making it a promising area for further research. Especially, considering that this the potential effect of XAI on the acquisition of insights can maximize the value of predictive analytics and enable Finance to inform, support and challenge the business, thereby improving the overall decision-making process. Lastly, the results on the conducted experiment showed that all three selected XAI techniques were deemed satisfactory based on the 5-point Likert scale for explanation satisfaction. Overall, the ALE plot scored highest, with an average explanation satisfaction score of 3.974, followed by the VI plot with 3.833, and the BD plot with 3.769.

6.1 Limitations

While both the development of the XAI prototype and the experiment conducted on the basis of this prototype provided promising results, there are a few things to keep in mind. One aspect to keep in mind concerns the impact of both the characteristics of the sample group, as well as the chosen Financial Forecasting solution on the generalisability of our findings. Firstly, nearly half of the participants that partook in the experiment originate from a single industry, namely the Financial Services Industry. It is important to note that this does not mean that these participants fulfil typical FSI jobs, but rather that they work within the Finance Function of a company that operates within the FSI. Although the other half of the sample group consists of consultants who arguably represent a wide range of industries as they provide consultation to and operate in all industries, overall results are still skewed in the direction of the FSI. Consequently, this means that we can not state with 100% certainty whether our findings are generalizable to Finance professionals operating in other industries than the FSI. Secondly, the experiment conducted in this research was based on a particular Financial Forecasting solution, namely PrecisionView™. Steps were taken to ensure that the verified XAI prototype was as generic as possible by only incorporating model-agnostic techniques that can easily be implemented to other Financial Forecasting solutions. Furthermore, to ensure that any observed effects on trust, understandability and insights are truly and only attributable to the use of XAI, we kept the Financial Forecasting dashboards without XAI completely identical to our XAI prototype, only excluding the XAI techniques. Despite these measures taken, we can not rule out the possibility that findings might differ for other Financial Forecasting solutions. Different results could be caused by either the use of different forecasting models or the use of other dashboarding tools that contain different visualisation capabilities.

A second aspect to take into consideration concerns the generalisability of the proposed XAI prototype. The logical architecture, as well as the selected set of XAI techniques are generic in that they can easily be extended to Financial Forecasting solutions that use a different software stack or incorporate other types of ML models. As explained in section 3.1, the chosen set of XAI techniques consists of model-agnostic, post-hoc methods, which enables others looking to replicate our approach to incorporate the same set of techniques, irrespective of the specific type of ML models used. Furthermore, explanation data produced by our XAI Generation Model can be used to visualise explanations in any dashboarding tool and hence is not restricted to Financial Forecasting solutions whose software stack uses Tableau as its visualisation component. However, the technical implementation details inside the XAI Generation Module are `DALEX` specific. The `DALEX` library was chosen with the generic characteristic of our prototype in mind, namely because it offers several model-agnostic, post-hoc on both a local and global level, and supports both R and Python implementations. Nevertheless, it does restrict the use of our XAI Generation Module to others wanting to replicate our approach, in that it is dependent on the use the `DALEX` library.

A third aspect to take into consideration concerns the size of the sample group that partook in the validation experiment. As mentioned previously, when sample sizes are relatively small, chances at finding statistically significant results becomes less likely. This also means that replication of the validation experiment presented in this research amongst a greater sample group could change the significance of some of our findings.

Lastly, and perhaps most importantly, is the relatively limited level of the sample group's experience with XAI based upon which the results were derived. As discussed in section 5.3, there is a relation between the amount of experience with and exposure to XAI techniques and the positive benefits of XAI that can be measured. Through means of the case studies, as well as the demonstration video participants were asked to watch prior to the conducting the experiment, we tried to mitigate any negative effects due to a lack of experience with the XAI technique. However, due to time constraints we were unable to perform multiple experiment rounds with participants to ensure a sufficient level of acquaintance with the XAI techniques. We saw this reflected in increased completion time for the case study with XAI, as well as comments participants made in this regard. This means that findings might differ when multiple repetitions of the experiment with the XAI prototype are conducted.

6.2 Future Work

Observations made during the development of our XAI prototype, the feedback on the individual XAI techniques collected during the experiment, and the limitations stated above, provide several areas for further research. These suggestions can be roughly divided into improvements on the experimental design and improvements on the XAI prototype and incorporated XAI techniques.

Firstly, a promising direction for future research would be to measure the effects of XAI on trust, understandability and insights, based on the conduction of repetitive or more extensive experiment rounds aimed at increasing participants' level of experience with the XAI techniques. As explained above, the limited level of participants' experience caused an increase rather than the expected decrease in the completion time for case studies conducted with the availability of XAI. Future research in this direction could verify whether this also caused the observed effect of XAI on the quality of insights to be too small to find statistically significant results. In addition to that, it could help answer the question whether both trust in and understandability of Financial Forecasting solutions increases further when participants are more acquainted with the XAI. Secondly, as discussed in section 6.1, the experiment could be expanded to include Finance professionals from other industries, as well as expand the size of the sample group as a whole. The results must show whether such expansions have an influence on the observed effect of XAI on trust, understandability and insights. Lastly, it would be interesting to source more extensive, real-world case studies from actual Finance departments, potentially accross different industries, and measure the change in effect caused by XAI. The case studies used in this research were relatively simple. Using more extensive and complex case studies, would requires participants to conduct a more in-depth scenario analysis which could also impact the observed effect of XAI. Secondly, financial reports and the pairing analysis can differ amongst industries. For example, seasonal influences or inflation might be of greater importance on the financial results of one company, but insignificant for the results of others. Different industries work with different drivers that impact their financial results and forecasts. Therefore, using case studies that are more representative of the real world scenarios observed amongst different industries might also lead to different findings.

A first, interesting area for future research relating to the improvement of the proposed XAI prototype is referred to as feature grouping. Feature grouping is an idea that was developed in the preliminary phase of this research, but ultimately has been left unexplored due to time constraints. The current set of feature influence techniques within the field of XAI, discussed in section 2.4.5, focusses on explaining the relationship between the independent variables and the dependent variable. However, limited research has been done into techniques that provide deeper insights into the independent variables itself. The idea behind feature grouping is to provide such insights, by classifying the independent variables based on certain properties they hold. Firstly, classifying or partitioning the independent variables can offer a solution to the input space problem, which refers to the negative correlation between the size of the number of independent variables, and the quality of an explanation [ADRDS+20, KM19]. Although partitioning does not reduce the input space, it does provide a more structured overview based on the property for which it is partitioned. Grouping of variables can take place along various dimensions, depending on the model under investigation. Interesting dimensions that we have explored and are potentially interesting to consider in future work include controllability, collinearity and interaction.

- **Controllability.** Partitions the set of variables based on whether the user has control over them or not. This supports the user in scenarios where he or she wants to explore possibilities to influence the foreseen prediction, either positively or negatively, by identifying which variables he or she has control over. Consider, for example, an input variable *temperature* that might have a large effect on the prediction, but is not controllable by the user.

- **Collinearity.** Partitions the set of variables based on correlation amongst them, if present. This can include correlation amongst 2 variables as well as multicollinearity, i.e. correlation amongst more than 2 variables. Information on collinearity helps users understand what happens with other variables correlated to a certain variable they want to alter to influence the forecast. Suppose, for example, that a user wants to increase the forecast. By analysing the feature attribution explanation of the forecast model in question, he or she might decide to do so by increasing a certain variable A that attributes greatly to the increase of the forecast. If this variable A is, however, positively correlated with some other variable B that negatively influences the prediction, then the user's efforts to increase the forecast might be partially or completely cancelled out by this negative effect of B. Hence, it is essential for the user to not only understand the relations between the independent variables and dependent variable, but also those amongst the independent variables itself.

- **Interaction.** Although closely related to collinearity, interaction is not the same as collinearity. Two variables are said to interact when their combined effect on the prediction does not equal the sum of their individual effects on the prediction [Mol20]. It is important for the user of a model to know which variables interact when tweaking variable values to alter the forecast. The motivation for this is the same as for collinearity, namely that when lacking knowledge on the potential interaction effects, users are not fully aware of the impact on the prediction when altering certain variables. Some research has been done into providing insights into these interactions. Henelius et al. proposed an iterative algorithm, called `GoldenEye`, to group interacting variables of classification models [HPB+14].

What makes the suggestion for feature grouping even more interesting and promising is that it was found to be a desire expressed by several participants that partook in this research. As discussed in section 5.3 several feedback comments on the individual XAI techniques stated the difficulty participants experienced due to the lack of information on the joint effect of drivers. Furthermore, two feedback comments explicitly suggested a further division or break-down of drivers based on the interrelationships, i.e. collinearity and interaction, and the influenceability, i.e. controllability, of the drivers.

Two more areas for future research aimed at improving the proposed XAI prototype were identified during its development. Firstly, we found that in the case of PrecisionView™, certain financial line items are forecasted on a segment or subsegment basis and added together to obtain the overall forecast for that financial line item, as illustrated in Figure 4.2. The forecasts generated for different (sub-)segments often incorporate different drivers or independent variables. As a consequence of this, the Break-down plots generated for these lower level forecasts differ greatly. Currently, there does not exist an approach to combine BD plots of lower level forecasts into a single BD plot explaining the overall forecast, obtained by adding lower level forecasts. However, the computation of forecasts that are derived by taking the sum of lower level forecasts computed on a segment, subsegment or geographical basis appears to be a common practice in the Financial Forecasting process of larger companies. Therefore, an interesting area for future work would be to develop a solution that enables combining the explanations of several lower level forecasts into a single explanation for the overall forecast. The second area for improvement identified during the development of the proposed XAI prototype was already briefly discussed in section 4.3.2. It concerns the peculiarities that arise when generating Break-down plots for SARIMAX models, as illustrated in Figures 4.7 and 4.8. It appears that the issue arises because the predicted value averaged over all instances in the dataset does not equal the predicted value for the specific forecast instance for which the BD plot is obtained. As a consequence of this, the prediction depicted in the second last contribution bar $royalty\_exp\_external = 12570$ in Figure 4.8, does not align with the last contribution bar *prediction*. As explained in section 3.2.1, the computation of the BD plot involves changing the driver values of all data instances one by one, to equal the driver values of the instance of interest. For example, to obtain the BD plot depicted in Figure 4.8, first for all data instances the value of *advertising_promotion* is set to equal the *advertising_promotion* value of the instance of interest, namely 9435. Next, we change the values for the driver *marketing_sales*, etcetera. Hence, in the last step, the values of all the drivers equal the driver values of the specific instance of interest. Therefore, at this point we expect the forecasted values for all instances in the dataset to equal the forecast obtained for the instance of interest. However, certain factors incorporated in the formula for SARIMAX models seem to cause the average prediction for all data instances to differ from the prediction for the instance of interest. The additional factors incorporated in a SARIMAX model differ greatly depending on the values of the parameters p, d, q, P, D, Q and the seasonal period m. Therefore, more research needs to be done in order to understand the cause of peculiarities found in BD plots generated for SARIMAX models.

References

[AB18] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*, 6:52138–52160, 2018.

[ABC+20] Vijay Arya, Rachel KE Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C Hoffman, Stephanie Houde, Q Vera Liao, Ronny Luss, Aleksandra Mojsilovic, et al. Ai explainability 360: An extensible toolkit for understanding data and machine learning models. *Journal of Machine Learning Research*, 21(130):1–6, 2020.

[ABKN20] Hubert Anysz, Łukasz Brzozowski, Wojciech Kretowicz, and Piotr Narloch. Feature importance of stabilised rammed earth components affecting the compressive strength calculated with explainable artificial intelligence tools. *Materials*, 13(10):2317, 2020.

[Abr19] Soheila Abrishami. Time series analysis and forecasting for Business Intelligence applications. 2019.

[ADRDS+20] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115, 2020.

[Alp14] Ethem Alpaydin. *Introduction to machine learning*. Adaptive computation and machine learning. Third edition. edition, 2014.

[AMI16] Mohiuddin Ahmed, Abdun Naser Mahmood, and Md Rafiqul Islam. A survey of anomaly detection techniques in financial domain. *Future Generation Computer Systems*, 55:278–288, 2016.

[Ana20a] Anaplan. *Anaplan Platform overview*. Anaplan, jul 2020. https://www.anaplan.com/wp-content/uploads/2020/08/platformOverviewBrief_072020.pdf, [Accessed: February 2021].

[Ana20b] Anaplan. Intelligent forecasting and agile scenario planning. Technical report, Anaplan, sep 2020. https://www.anaplan.com/wp-content/uploads/2020/09/whitepaper_intelligentForecasting_webReady.pdf, [Accessed: February 2021].

[ana20c] anaplan. Statistical forecasting methods. Technical report, Anaplan, sep 2020. https://community.anaplan.com/pjakv59666/attachments/pjakv59666/communitykb/255/2/Stat%20Forecasting%20Methods%20High%20Level_v6.pdf, [Accessed: February 2021].

[AZ16] Daniel W Apley and Jingyu Zhu. Visualizing the effects of predictor variables in black box supervised learning models. *arXiv preprint arXiv:1612.08468*, 2016.

[BB20] P Biecek and T Burzykowski. Explanatory Model Analysis-Explore, Explain and Examine Predictive Models, 2020.

[BCG+21] Przemysław Biecek, Marcin Chlebus, Janusz Gajda, Alicja Gosiewska, Anna Kozak, Dominik Ogonowski, Jakub Sztachelski, and Piotr Wojewnik. Enabling Machine Learning Algorithms for Credit Scoring - Explainable Artificial Intelligence (XAI) methods for clear understanding complex predictive models. *arXiv preprint arXiv:2104.06735*, 2021.

[BDJS19] Philippe Bracke, Anupam Datta, Carsten Jung, and Shayak Sen. Machine learning explainability in finance: an application to default risk analysis. 2019.

[BGG+18] Jocelyn Barker, Amita Gajewar, Konstantin Golyaev, Gagan Bansal, and Matt Conners. Secure and automated enterprise revenue forecasting. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[Bie18] Przemysław Biecek. DALEX: explainers for complex predictive models in R. *The Journal of Machine Learning Research*, 19(1):3245–3249, 2018.

[Bie21] Przemysław Biecek. Model ingredients. https://github.com/ModelOriented/ingredients, mar 2021. 688f317ca59a9eeb162a7ef4938bb58928ee9e63.

[BM17] Erik Brynjolfsson and ANDREW Mcafee. The business of artificial intelligence. *Harvard Business Review*, pages 1–20, 2017.

[Bru] Thierry Brunet. Time Series Forecasting in SAP Analytics Cloud Smart Predict in Detail.

[Byr19] Ruth MJ Byrne. Counterfactuals in Explainable Artificial Intelligence (XAI): Evidence from Human Reasoning. In *IJCAI*, pages 6276–6282, 2019.

[CD13] P Cooper and Eleanor Dart. Business partnering as a complement to the accountant's other roles: International survey evidence. 2013.

[CD17] Robert Culkin and Sanjiv R Das. Machine learning in finance: The case of deep learning for option pricing. *Journal of Investment Management*, 15(4):92–100, 2017.

[CF09] Béatrice Cahour and Jean-François Forzy. Does projection into use improve trust and exploration? an example with a cruise control system. *Safety science*, 47(9):1260–1270, 2009.

[CGM15] CGMA. Finance business partnering, the conversations that count. https://www.cgma.org/Resources/Reports/Documents/CGMA-Business-partnering-report.pdf, oct 2015. [Accessed: May 2020].

[CIM09] CIMA. Improving decision making in organisations: The opportunity to reinvent finance business partners. https://www.cimaglobal.com/Documents/ImportedDocuments/cid_execrep_finance_business_partners_Jul09.pdf, jul 2009. [Accessed: April 2020].

[CLL19] Jian Cao, Zhi Li, and Jian Li. Financial time series forecasting model based on CEEMDAN and LSTM. *Physica A: Statistical Mechanics and its Applications*, 519:127–139, 2019.

[Coe02] Robert Coe. It's the effect size, stupid: What effect size is and why it is important. 2002.

[CRC16] Ning Chen, Bernardete Ribeiro, and An Chen. Financial credit risk assessment: a recent review. *Artificial Intelligence Review*, 45(1):1–23, 2016.

[CT01] Lijuan Cao and Francis EH Tay. Financial forecasting using support vector machines. *Neural Computing & Applications*, 10(2):184–192, 2001.

[DeB18] Chris DeBrusk. The risk of machine-learning bias (and how to prevent it). *MIT Sloan Management Review*, 2018.

[Del12] Deloitte. Changing the focus: Finance business partnering. https://www2.deloitte.com/content/dam/Deloitte/uk/Documents/finance-transformation/deloitte-uk-finance-business-partnering.pdf, aug 2012. [Accessed: April 2020].

[Del13] Deloitte. Finance business partnering: Making the right move. https://www2.deloitte.com/content/dam/Deloitte/ie/Documents/Finance%20Transformation/2013_finance_business_partnering_survey_deloitte_ireland_finance_transformation_finance_survey_partnering.pdf, feb 2013. [Accessed: May 2020].

[Del14] Deloitte. Finance business partnering: Less than the sum of the parts. https://www2.deloitte.com/uk/en/pages/finance/articles/finance-business-partnering.html, sep 2014. [Accessed: May 2020].

[Del18] Deloitte. Precisionview™ training. Technical report, Deloitte, aug 2018.

[Del19] Deloitte. Driving the future of finance: Leveraging digital opportunities to support strategic finance business partners focus on decision-making. https://www2.deloitte.com/content/dam/Deloitte/ch/Documents/finance/deloitte-ch-finance-transformation-driving-future-of-finance.pdf, jan 2019. [Accessed: May 2020].

[DP10] RK Dase and DD Pawar. Application of Artificial Neural Network for stock market predictions: A review of literature. *International Journal of Machine Intelligence*, 2(2):14–17, 2010.

[DR18] Thomas H Davenport and Rajeev Ronanki. Artificial intelligence for the real world. *Harvard business review*, 96(1):108–116, 2018.

[Gar16] Megan Garcia. Racist in the machine: The disturbing implications of algorithmic bias. *World Policy Journal*, 33(4):111–117, 2016.

[GB16] Amita Gajewar and Gagan Bansal. Revenue forecasting for enterprise products. *arXiv preprint arXiv:1701.06624*, 2016.

[GCR19] Hamed Ghoddusi, Germán G Creamer, and Nima Rafizadeh. Machine learning in energy economics and finance: A review. *Energy Economics*, 81:709–727, 2019.

[Gin11] Edward J Giniat. Using business intelligence for competitive advantage: the use of data analytics is emerging as a key discipline for healthcare finance. *Healthcare Financial Management*, 65(9):142–145, 2011.

[GL20] John van Decker Greg Leiter, Robert Anderson. Magic Quadrant for Cloud Financial Planning and Analysis Solutions. *Gartner*, oct 2020.

[Glo] Gartner Glossary. Enterprise Performance Management (EPM).

[GMR+18] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018.

[HA19] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, Melbourne, Australia, third edition, 2019. [Accessed: June 2020].

[HGP11] Balázs Hidasi and Csaba Gáspár-Papanek. ShiftTree: an interpretable model-based approach for time series classification. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 48–64. Springer, 2011.

[HMKL18] Robert R Hoffman, Shane T Mueller, Gary Klein, and Jordan Litman. Metrics for explainable AI: Challenges and prospects. *arXiv preprint arXiv:1812.04608*, 2018.

[HMYC18] Dongxu Huang, Dejun Mu, Libin Yang, and Xiaoyan Cai. CoDetect: financial fraud detection with anomaly feature detection. *IEEE Access*, 6:19161–19174, 2018.

[HPB+14] Andreas Henelius, Kai Puolamäki, Henrik Boström, Lars Asker, and Panagiotis Papapetrou. A peek into the black box: exploring classifiers by randomization. *Data mining and knowledge discovery*, 28(5-6):1503–1529, 2014.

[ICA18] ICAEW. Finance business partnering, a guide. https://www.icaew.com/-/media/corporate/files/technical/business-and-financial-management/finance-direction/business-partnering.ashx?la=en, jan 2018. [Accessed: May 2020].

[icwC11] KPMG in collaboration with CIMA. Mastering finance business partnering: The missing link to building finance's influence. https://www.cimaglobal.com/Documents/Thought_leadership_docs/KPMG%20CIMA%20business%20partnering%20white%20paper%20280111.pdf, feb 2011. [Accessed: May 2020].

[IGCB20] Igor Ilic, Berk Gorgulu, Mucahit Cevik, and Mustafa Gokce Baydogan. Explainable boosted linear regression for time series forecasting. *arXiv preprint arXiv:2009.09110*, 2020.

[Jan19] Hoon Jang. A decision support framework for robust R&D budget allocation using machine learning and optimization. *Decision Support Systems*, 121:1–12, 2019.

[JBL19] David Jonker, Richard Brath, and Scott Langevin. Industry-Driven Visual Analytics for Understanding Financial Timeseries Models. In *2019 23rd International Conference Information Visualisation (IV)*, pages 210–215. IEEE, 2019.

[JMK96] George H John, Peter Miller, and Randy Kerber. Stock selection using rule induction. *IEEE Expert*, 11(5):52–58, 1996.

[JT01] Y JingTao and Chew Lim Tan. Guidelines for financial forecasting with neural networks. In *Int. Conf. Neural Information Processing, Shanghai, China*. Citeseer, 2001.

[KM19] Jon Kleinberg and Sendhil Mullainathan. Simplicity creates inequity: implications for fairness, stereotypes, and interpretability. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pages 807–808, 2019.

[KRPG18] Isak Karlsson, Jonathan Rebane, Panagiotis Papapetrou, and Aristides Gionis. Explainable time series tweaking via irreversible and reversible temporal transformations. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 207–216. IEEE, 2018.

[KSAZ13] Jochen Kruppa, Alexandra Schwarz, Gerhard Arminger, and Andreas Ziegler. Consumer credit risk: Individual probability estimates using machine learning. *Expert Systems with Applications*, 40(13):5125–5131, 2013.

[LB19] Salim Lahmiri and Stelios Bekiros. Can machine learning approaches predict corporate bankruptcy? evidence from a qualitative experimental design. *Quantitative Finance*, 19(9):1569–1577, 2019.

[LL17] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774, 2017.

[LLC09] Chi-Jie Lu, Tian-Shyug Lee, and Chih-Chou Chiu. Financial time series forecasting using independent component analysis and support vector regression. *Decision support systems*, 47(2):115–125, 2009.

[MA21] Eric Merrill and Tay Adrian. PrecisionView™ Advanced Forecasting and Modeling, 2021.

[May20] Melina Mayer. Your opportunity to Influence SAP – Customer Engagement Initiative Projects for SAP Intelligent Technologies (Machine Learning, Artificial Intelligence, Blockchain, IoT), feb 2020.

[MG00] Maria Madsen and Shirley Gregor. Measuring human-computer trust. In *11th australasian conference on information systems*, volume 53, pages 6–8. Citeseer, 2000.

[Mol20] Christoph Molnar. *Interpretable Machine Learning*. Lulu. com, 2020.

[Mor19] Laurence Moroney. Machine learning zero to hero (google I/O ’19), 2019.

[Nie15] Michael A Nielsen. *Neural networks and deep learning*, volume 25. Determination press San Francisco, CA, 2015.

[ÖA98] Dilek Önkal-Atay. Financial forecasting with judgement. *Forecasting with judgment*, pages 139–167, 1998.

[Oli91] Lianabel Oliver. Accountants as business partners. *Strategic Finance*, 72(12):40, 1991.

[Ora21] Oracle. *Oracle® Cloud, Working with Planning*, 2021. https://docs.oracle.com/en/cloud/saas/planning-budgeting-cloud/pfusu/EPM-INFORMATION-DEVELOPMENT-TEAM-E94218-6693400D.pdf, [Accessed: February 2021].

[PE14] Dilek Penpece and Orhan Emre Elma. Predicting sales revenue by using artificial neural network in grocery retailing industry: a case study in turkey. *International Journal of Trade, Economics and Finance*, 5(5):435, 2014.

[Pis18] Joseph Pistrui. The future of human work is imagination, creativity, and strategy. *Harvard Business Review*, 1(18):2018, 2018.

[PJC19] Frank Pasquale, Kristin Johnson, and Jennifer Elisa Chapman. Artificial Intelligence, Machine Learning, and Bias in Finance: Toward Responsible Innovation. 2019.

[Pro19] The CFO Program. Central europe CFO survey. Technical Report 10, Deloitte, 2019.

[PWC17] PWC. Finance as a business partner: Adding up or adding value. https://www.pwc.nl/nl/assets/documents/pwc-finance-as-business-partner-adding-up-or-adding-value-2017.pdf, jan 2017. [Accessed: May 2020].

[Qua21] Qualtrics. https://www.qualtrics.com, 2021.

[Qui14] Martin Quinn. The elusive business partner controller. *Controlling & Management Review*, 58(2):22–27, 2014.

[QWD+16] Junfei Qiu, Qihui Wu, Guoru Ding, Yuhua Xu, and Shuo Feng. A survey of machine learning for big data processing. *EURASIP Journal on Advances in Signal Processing*, 2016(1):67, 2016.

[RSG16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[SAEA+19] Udo Schlegel, Hiba Arnout, Mennatallah El-Assady, Daniela Oelke, and Daniel A Keim. Towards a rigorous evaluation of XAI Methods on Time Series. *arXiv preprint arXiv:1909.07082*, 2019.

[Sam15] Michael Samonas. *Financial forecasting, analysis, and modelling: a framework for long-term forecasting*. John Wiley & Sons, 2015.

[SAP21] SAP. SAP Analytics Cloud Help, feb 2021.

[Sch19] Utz Schäffer. *The Future of Controlling*. PhD thesis, WHU–Otto Beisheim School of Management, 2019.

[SGY+15] Damian Dion Salam, Irwan Gunardi, Amega Yasutra, et al. Production optimization strategy using hybrid genetic algorithm. In *Abu Dhabi International Petroleum Exhibition and Conference*. Society of Petroleum Engineers, 2015.

[Smi15] Sean Stein Smith. Accounting, integrated financial reporting, and the future of finance. *Journal of Accounting and Finance*, 15(2):11, 2015.

[SSBD14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[Tha12] Paul Thambar. The transforming finance function: Implications for the education and training of accountants. *Emerging Pathways for the Next Generation of Accountants*, page 65, 2012.

[THSK21] Di Tian, Xiaogang He, Puneet Srivastava, and Latif Kalin. A hybrid framework for forecasting monthly reservoir inflow based on machine learning techniques with dynamic climate forecasts, satellite-based data, and climate phenomenon information. *Stochastic Environmental Research and Risk Assessment*, pages 1–23, 2021.

[TK17] Eric R Teoh and David G Kidd. Rage against the machine? google's self-driving cars versus human drivers. *Journal of safety research*, 63:57–60, 2017.

[TL18] Sean J Taylor and Benjamin Letham. Forecasting at scale. *The American Statistician*, 72(1):37–45, 2018.

[Tyn16] Jannika Tynkkynen. Finance as a business partner at marimekko: Searching for finance business partnering opportunities at a finnish design company. 2016.

[vdBK20] Martin van den Berg and Ouren Kuiper. XAI in the Financial Sector. 2020.

[vdV14] Arco van de Ven. Controller en business partner. *Maandblad Voor Accountancy en Bedrijfseconomie*, 88:84, 2014.

[Ven15] Sailesh Venkatraman. Business partnering. *Strategic Finance*, 97(2):47–53, 2015.

[VSO+13] Chang Sim Vui, Gan Kim Soon, Chin Kim On, Rayner Alfred, and Patricia Anthony. A review of stock market prediction with Artificial Neural Network (ANN). In *2013 IEEE international conference on control system, computing and engineering*, pages 477–482. IEEE, 2013.

[Wat17] Heather Watson. The future of the finance function. 2017.

[WS19] Weiyu Wang and Keng Siau. Artificial intelligence, machine learning, automation, robotics, future of work and future of humanity: a review and research agenda. *Journal of Database Management (JDM)*, 30(1):61–79, 2019.

[You15] Ernst & Young. Partnering for performance, part 5: the CFO and the chief executive officer. https://codfiscal.net/media/2015/12/REPORT-CFO-CEO-Partnering-for-performance.pdf, jan 2015. [Accessed: May 2020].

[You16] Ernst & Young. The DNA of the CFO: Is the future of finance new technology or new people? https://www.ey.com/en_gl/advisory/is-the-future-of-finance-new-technology-or-new-people, feb 2016. [Accessed: May 2020].

[ZLR+18] Jichen Zhu, Antonios Liapis, Sebastian Risi, Rafael Bidarra, and G Michael Youngblood. Explainable AI for designers: A human-centered perspective on mixed-initiative co-creation. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–8. IEEE, 2018.

[ZNGL20] Yi Fei Zhang, Mohammad Namazi, Yong Qing Guo, and Xuan Li. Finance business partnering and manufacturing firms' performance: a mediating role of non-financial performance. *Journal of Business Economics and Management*, 21(2):473–496, 2020.

A Appendix

A.1 System Design

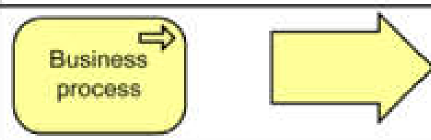
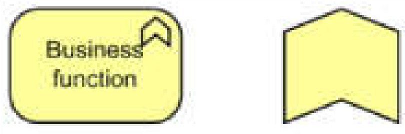
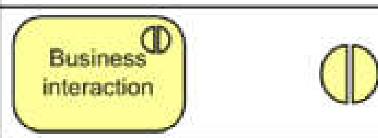
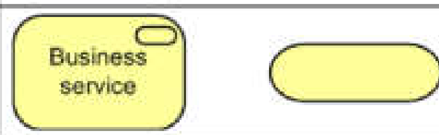
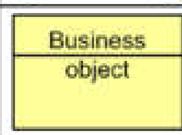
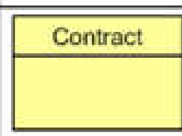
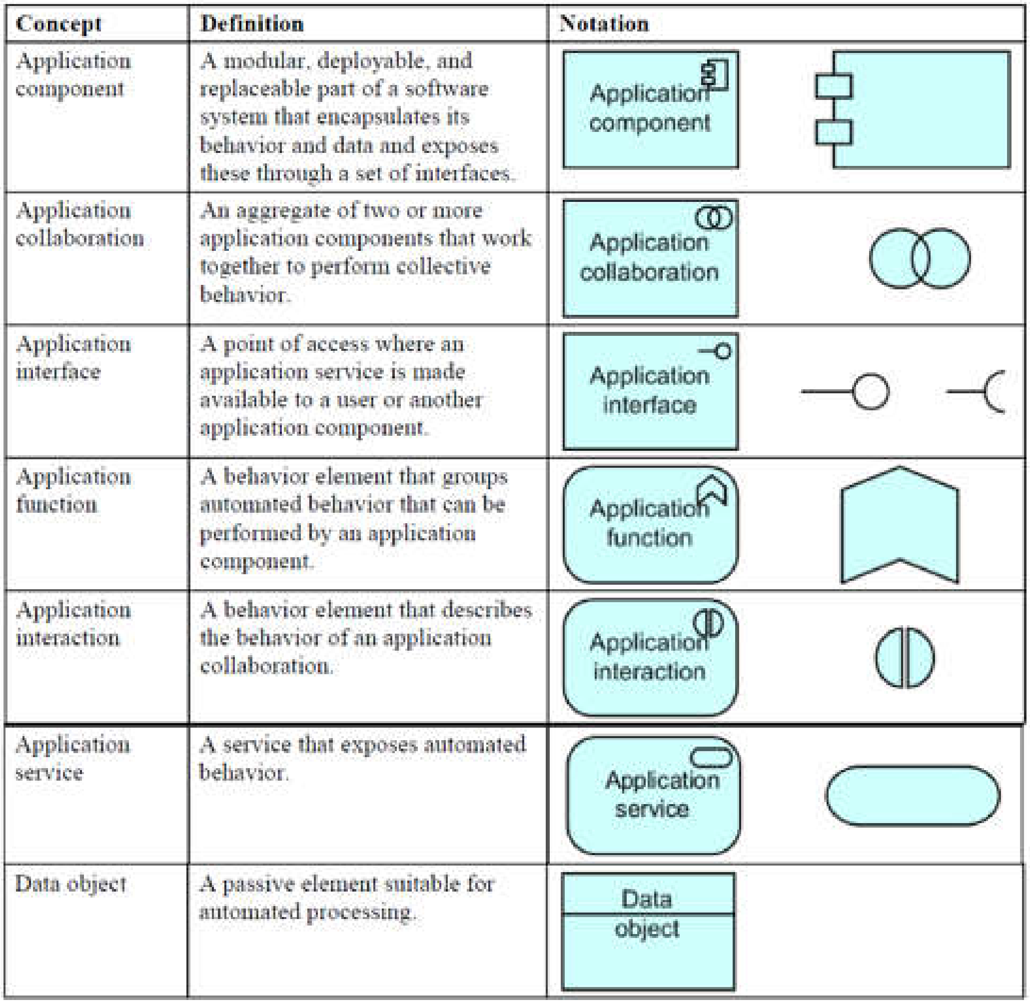
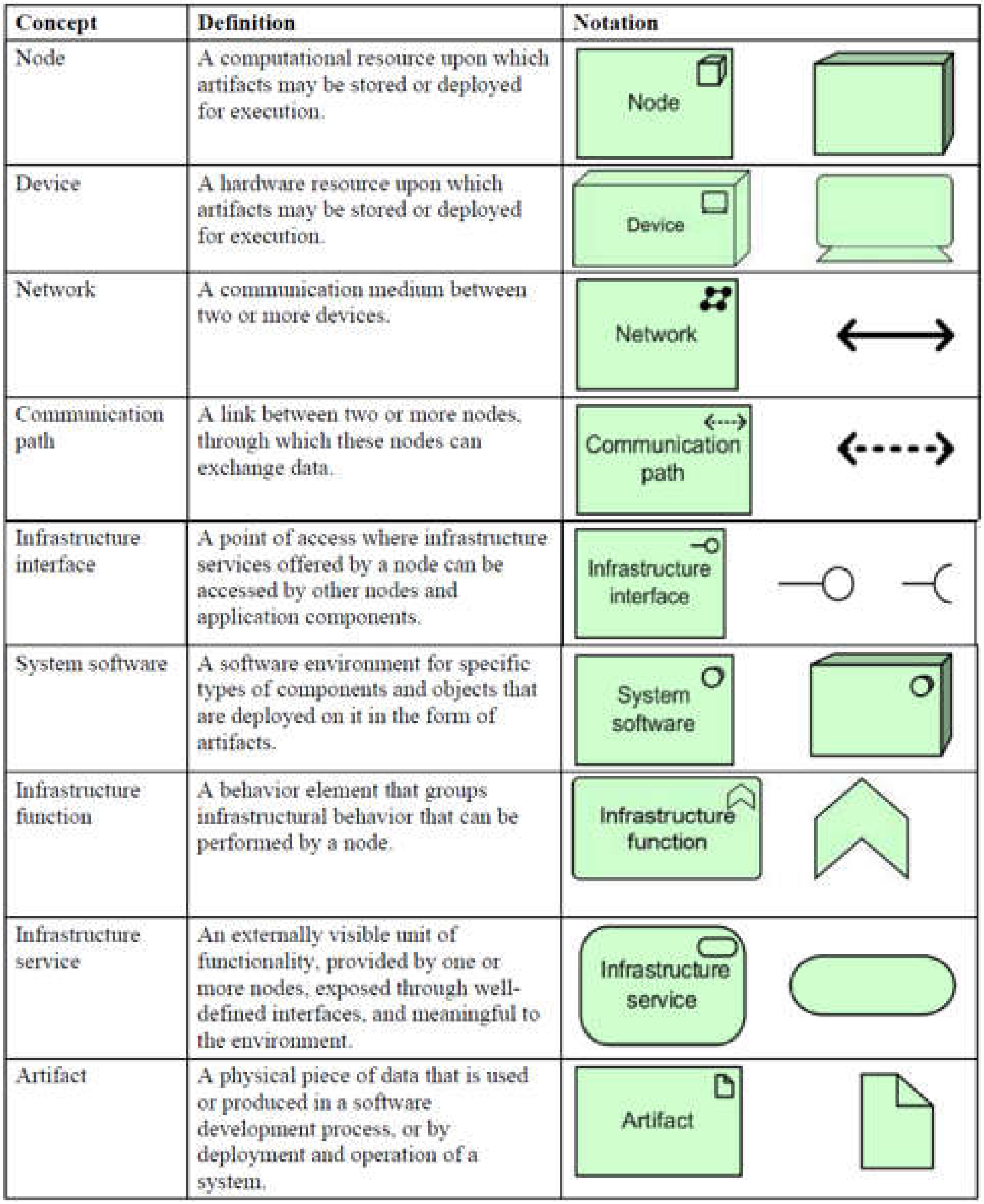
Concept	Description	Notation
Business process	A behavior element that groups behavior based on an ordering of activities. It is intended to produce a defined set of products or business services.	
Business function	A behavior element that groups behavior based on a chosen set of criteria (typically required business resources and/or competences).	
Business interaction	A behavior element that describes the behavior of a business collaboration.	
Business event	Something that happens (internally or externally) and influences behavior.	
Business service	A service that fulfills a business need for a customer (internal or external to the organization).	
Business object	A passive element that has relevance from a business perspective.	
Representation	A perceptible form of the information carried by a business object.	
Meaning	The knowledge or expertise present in a business object or its representation, given a particular context.	
Value	The relative worth, utility, or importance of a business service or product.	
Product	A coherent collection of services, accompanied by a contract/set of agreements, which is offered as a whole to (internal or external) customers.	
Contract	A formal or informal specification of agreement that specifies the rights and obligations associated with a product.	

Figure A.1: ArchiMate symbols that model the behavioral structure in an architecture.



Concept	Definition	Notation
Application component	A modular, deployable, and replaceable part of a software system that encapsulates its behavior and data and exposes these through a set of interfaces.	Application component
Application collaboration	An aggregate of two or more application components that work together to perform collective behavior.	Application collaboration
Application interface	A point of access where an application service is made available to a user or another application component.	Application interface
Application function	A behavior element that groups automated behavior that can be performed by an application component.	Application function
Application interaction	A behavior element that describes the behavior of an application collaboration.	Application interaction
Application service	A service that exposes automated behavior.	Application service
Data object	A passive element suitable for automated processing.	Data object

Figure A.2: ArchiMate symbols that model the active structure in an architecture.



Concept	Definition	Notation
Node	A computational resource upon which artifacts may be stored or deployed for execution.	Node
Device	A hardware resource upon which artifacts may be stored or deployed for execution.	Device
Network	A communication medium between two or more devices.	Network
Communication path	A link between two or more nodes, through which these nodes can exchange data.	Communication path
Infrastructure interface	A point of access where infrastructure services offered by a node can be accessed by other nodes and application components.	Infrastructure interface
System software	A software environment for specific types of components and objects that are deployed on it in the form of artifacts.	System software
Infrastructure function	A behavior element that groups infrastructural behavior that can be performed by a node.	Infrastructure function
Infrastructure service	An externally visible unit of functionality, provided by one or more nodes, exposed through well-defined interfaces, and meaningful to the environment.	Infrastructure service
Artifact	A physical piece of data that is used or produced in a software development process, or by deployment and operation of a system.	Artifact

Figure A.3: ArchiMate symbols that model the passive structure in an architecture.

A.2 System Implementation

```
getTidyNames <- function(inputpath, inputfile, myModel, modelType,
    independentVar){
     segment <- substr(inputpath, stringi::stri_locate_last_fixed(inputpath,
    '/')+1, nchar(inputpath))

     inputfileName <- substr(inputfile,
    stringi::stri_locate_last_fixed(inputfile, '/')+1, nchar(inputfile))
```

```
    segmentCode <- substr(inputfileName,
   stringi::stri_locate_first_fixed(inputfileName, '_')+1,
   stri_locate_last_fixed(inputfileName, '_')-1)
    subsegment <- substr(inputfile, stringi::stri_locate_last_fixed(inputfile,
   segmentCode)+3, nchar(inputfile)-4)

    myFilename <- paste(subsegment, independentVar, sep = "_");
    myFilepath <- paste(paste("05", segmentCode, sep = "_"), "XAI", sep = " ")

    dirs <- c(myFilepath, "XAI", modelType, subsegment, independentVar)
    checkFilepath(dirs, inputpath)

    myFilepath <- paste(myFilepath, "XAI", modelType, subsegment,
   paste0(independentVar, '/'), sep = '/')

    independentVarTidy <-
   stringr::str_to_sentence(stringr::str_replace_all(independentVar, "_", " "))
    myLabel <- paste(paste(segment, subsegment, sep = ", "),
   independentVarTidy, sep = " - ")

    return (list(
        "myLabel" = myLabel,
        "myFilename" = myFilename,
        "myFilepath" = myFilepath)
    )
}
```

Listing A.1: The implementation of the *getTidyNames* function, specifically for PrecisionView™.

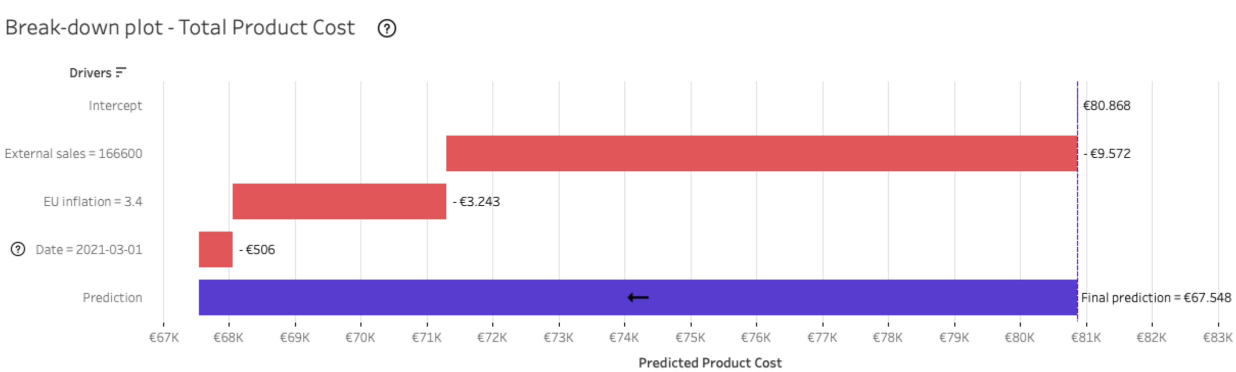


Figure A.4: The BD plot for the product cost prediction model in Tableau.

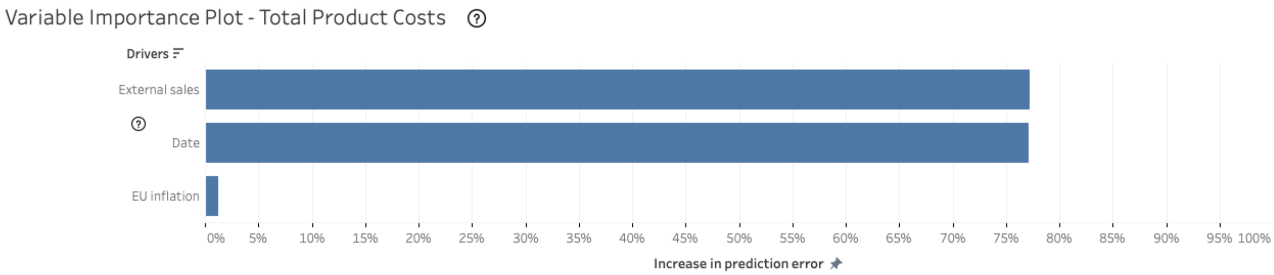


Figure A.5: The VI plot for the product cost prediction model in Tableau.

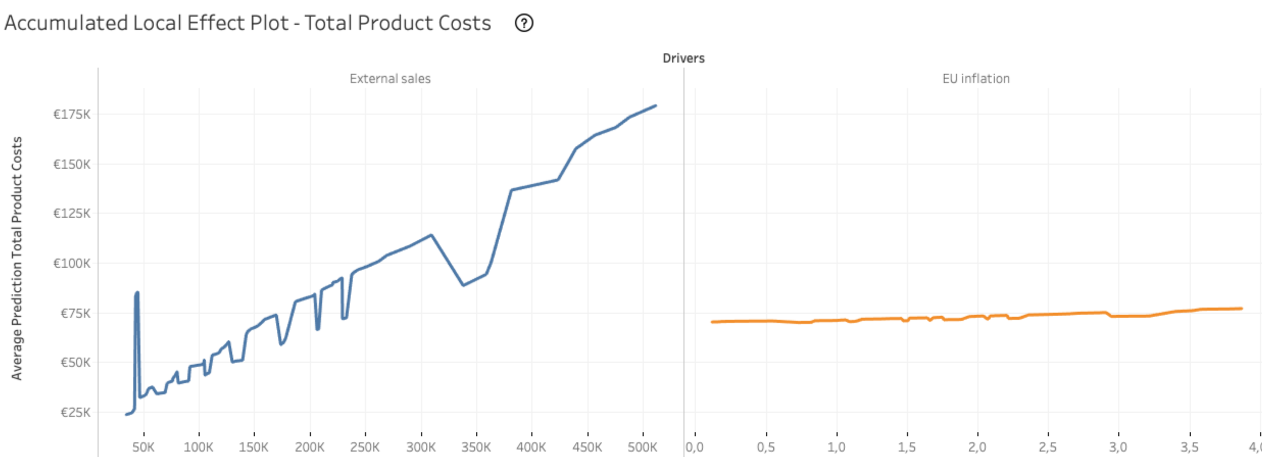


Figure A.6: The ALE plot for the product cost prediction model in Tableau.