Universiteit Leiden
The Netherlands

Opleiding Informatica & Economie

From Financial Statements to Business Sector Prediction: a Data Mining Approach

Mitch Angenent

Supervisors:
Frank Takes & António Barata

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

August 15, 2019

Abstract

In business, financial statements are a valuable source of information. Since they are abundantly available in open data sets, data mining techniques can be applied to obtain interesting patterns and predictions. Previous work mainly focused on predicting fraud and bankruptcy. In this study we assess the ability to perform sector prediction by applying data mining techniques on financial statements. Sector predictions have valuable applications such as filling up missing sectors to perform analysis on a larger proportion of a sector, or check company statements for errors and potential fraud. We used a data set from the Dutch Chamber of Commerce, containing 1,517,400 financial statements from Dutch companies from all sectors. A baseline was established by determining the performance of random forest classifiers, in terms of a ROC curve for each class and mean of their AUCs. Five approaches to deal with class imbalance were applied, among one custom approach. To determine the impact of financial ratios on sector prediction, we computed five financial ratios from the data set and applied all approaches to it. The most important features, and the optimal number of features, were retrieved by analysing feature importance. With these experiments, we concluded that data mining techniques by means of supervised learning on financial statements can be used for reasonable prediction of the sector in which a company is active. Financial ratios did not contribute to a better results. Furthermore, we concluded that the use of 26 to 28 of the most important features is optimal for classification. By means of feature importance, a small number of features, including missing-indicators, were considered most important.

Acknowledgements

This thesis is a partial fulfillment of the requirements of the bachelor degree Computer Science & Economics (”Informatie & Economie”), taught at the Leiden Institute of Advanced Computer Science (LIACS), Leiden University. I would like to thank my supervisors, Frank Takes and António Barata, for their intensive support during this thesis project.

Contents

Chapter 1

Introduction

Financial statements form the backbone of accounting. They play an important role in business, providing relevant financial information to stakeholders. Financial statements can be used to analyse a company over time and enables comparisons between companies. This information is internally used by controllers, managers, and the board, and externally by potential investors, analysts, and shareholders. Since the majority of established companies generate annual reports, higher-level applications such as sector analysis and anomaly detection are possible. The introduction of a digital standard for financial data, eXtensible Business Reporting Language (XBRL) [1], enforces the use of financial data at such a level. However, analysing this type of data is a hard and time consuming task for regular analysts. The field of data mining offers a solution. Data mining techniques are deployed for analysing large amounts of structured data, and has the potential to retrieve relevant relations between attributes of financial statements. Data mining techniques differ from regular analysts by obtaining new insights how to analyse this data, and the ability to perform these analysis automatically once a model has been established. We are particularly interested in sector prediction by applying data mining techniques on financial statements. Our interest for sector predictions is motivated by two possible application. First, by predicting the sector of companies without a sector label it is possible to perform analysis on a larger proportion of a sector or market. Examples of such analysis are, among others, determining the indebtedness for a sector, and comparing the performance of a company to the sectors average. Second, a predictive model can aid government institutions in checking filed statements on their correctness, automatically detecting potential errors and fraud.

Previous studies mainly focused on predicting whether or not a filed statement might be a Fraudulent Financial Statement (FFS). In 2012, Sharma et al. [2] categorized 35 papers from the period between 1995 and 2012 that researched FFSs. Amongst other algorithms, the authors applied neural networks, decision trees, support vector machines, and logistic regression on relative small data sets of 100 - 1,000 instances. In addition, other studies focused on predicting bankruptcy. For example, Zieba et al. [3] assessed the problem of predicting bankruptcy of Polish companies from the manufacturing sector. After extensively comparing available algorithms they concluded than an ensemble of trees had the best predictive ability in this classification problem. Ten of the eleven studies researching fraud and bankruptcy included financial ratios in their list of features.

Five of those studies concluded that the use of financial ratios is useful for financial statement analysis.

This study contributes to related work by assessing the ability to perform sector prediction by applying data mining techniques on financial statements. To the best of our knowledge, this has not yet been investigated. Moreover, this study further improves on past literature by using a data set of over 50-fold number of annual reports. The data set used origins from the Netherlands Chamber of Commerce (KvK) [4]. Among their tasks is keeping the Dutch Commercial Register and the collection of annual reports from registered companies that are mandatory to deposit them. Because of governmental transparency, anonymous versions of annual reports are combined into an open source data set [5]. It is continuously updated and contains the established, in XBRL deposited, annual reports from the last three closed financial years and is supplemented with the current year. The version we use contains 1,517,400 anonymous financial statements, distributed over the years 2015 – 2018, from all Dutch sectors.

In summary, the research question is as follows: *How can data mining techniques be used to predict the business sector of a company based on their anonymised financial statement?*

To get a more complete understanding of the topic, we propose to answer the following underlying questions:

1. Can financial ratios be used to improve sector prediction using financial statements?
2. What are the most important attributes of financial statements useful for sector prediction?
3. What is the optimal number of attributes of financial statements for sector prediction?

To answer the aforementioned questions, we follow a *data driven* approach. Therefore, no hypothesis is formulated. The core of this *data driven approach* is a classification model. We roughly follow the following steps. First, the data set is preprocessed, including the computation of financial ratios, adding missing-indicators, and reducing the number of classes. Then, classifiers will be trained on the instances of the data that contain a sector code. Several approaches are applied and compared. At last, the best performing approach is used to answer the sub questions.

In addition to the aforementioned applications and contributions to related work, we contribute to the computer science field by providing further insights on how to perform this kind of analysis. Best practises can be used to predict a different range of attributes or improve research on the current attribute.

The structure of this thesis is as follows. Chapter 2 provides background information about financial statements, sector categorization, and data mining. Chapter 3 discusses previous work related to ours. Chapter 4 describes the data set and methodology. Chapter 5 presents the setup and results of our experiments. Chapter 6 concludes this work and contains recommendations for future work.

Chapter 2

Definitions and Context

This chapter provides background information about the topics that are combined in this interdisciplinary research. The first section introduces the concept of financial statements, whereas the second section dives into details of the categorization of sectors in the Netherlands. The last section relates to the data mining field, including common challenges and techniques.

2.1 Financial statements

A financial statement must contain at least three elements: an income statement, a statement of cash flows, and a balance sheet. An income statement holds information about revenues and expenses of the company over a specific period, usually a calendar year. All single expenses from that period are summed up on ledger cards; these ledgers are based on accounting standards and are usually tailored for a specific company. The statement of cash flows shows the incoming and outgoing amounts of cash over the same specific period. Where the income statement represent the profitability of a company, the statement of case flows represents its liquidity. High profits without collecting revenue can result in a net outflow of cash, which leads to a higher risk of money shortage. Lasty, the balance sheet indicates the state of a organization at specific time, usually the end of a financial year (FY). This state is represented with the balance of standardized ledgers, such as: 'Accounts Receivables', 'Accounts Payable', 'Inventory' and 'Property', to name a few. These detailed ledgers are aggregated into the overarching categories 'Assets', 'Liabilities' and 'Equity'.

XBRL was created to enforce the standardization of financial statements and is currently implemented in more than 50 countries. XBRL is based on the XML language and allows reporting terms to be authoritatively defined. This enables more practical, more accurate, and faster comparisons between organizations. Central authorities, such as tax authorities and chambers of commerce, leverage the standardization for easier comparison [1]. An example is the KvK, which obliges certain companies to deposit their financial statement in the XBRL standard. This led to faster depositions, smaller files, easier financial comparisons and higher quality of financial statements for medium-sized Dutch companies in 2017 [6]. Imposing new standards takes

time, especially for larger corporations. Therefore, the smallest companies are first in the transition to the XBRL standard. Three criteria are used to categorize Dutch companies into four sizes, listed in Table 2.1.

Table 2.1: **Categorization of company sizes**. The financial year (FY) from when depositing in XBRL is mandatory depends on the company size [7]. * Listed companies and companies that must deposit under foreign law are excepted.

Requirement	**Micro**	**Small**	**Medium**	**Large**
Assets	$<$ €350.000	€350.000 - €6 m	€6 m - €20 m	$>$ €20 m
Net revenue	$<$ €700.000	€700.000 - €12 m	€12 m - €40 m	€40 m
Number of employees	$<$ 10	10 - 50	50 - 250	$>$ 250
Deposit in XBRL *	From FY 2016	From FY 2016	From FY 2017	From FY 2019

Based on the aforementioned information, there are three essential points to note regarding the represented companies in our data set. First, not all Dutch companies are obliged to deposit their annual reports. This depends on their legal form: companies with limited liability legal form, or companies that have public shares are obliged to deposit [7]. Second, not all companies that are obliged to deposit their annual reports are obligated to deposit them in XBRL and therefore not present in the data set. Third, the sector code is not a mandatory field to fill in. Therefore, it is likely that the data set mainly contains micro to medium sized companies with a limited liability legal form. Furthermore, we would expect more attributes from balance sheets than attributes from income statements in the data set. This is based on the fact that less companies are obliged to deposit their income statement as part of the annual report [7].

2.2 Sector categorization and coding

Since this research focuses on sector prediction of Dutch companies, it is relevant to understand the Dutch sector coding. The Dutch organization for statistics, Statistics Netherlands (CBS), defined the standard industrial classification (SBI) for the Netherlands. The SBI is a hierarchical mapping based on economic activities to classify a business in terms of their primary business activity [8]. It distinguishes five levels with references to other classification systems [9]:

1. The section, represented by a character.
2. The department, represented by two digits. These digits match the notation of the both the international (ISIC) and European (NACE) categorizations.
3. The activity, represented by three digits.
4. The activity, represented by four digits. This matches the NACE notation.
5. The activity, represented by five digits. It is based on the four digit NACE notation, with small adjustments for the Netherlands.

The longer the code, represented by the number of characters, the more information it details about the companies' activity. Moreover, the number of distinct codes per hierarchical level increases as the hierarchical standing decreases. This is illustrated in Table 2.2.

Table 2.2: Number of distinct categories for the most commonly used levels of SBI coding. SBI 2008, version 2019. [10]

Coding	Type	Number of distinct
One character SBI code	Sections	21
Two digit SBI code	Departments	95
Three to five digit SBI code	Activities	1348

2.3 Data mining

On an everyday basis, we apply knowledge to a situation before we can make a reasonable decision. Despite that tremendous amounts of data available, it not directly applicable to those situations. As illustrated by the DIWK pyramid in Figure 2.1, data lacks meaning and context. Therefore, data should be processed to obtain knowledge for it. This process is commonly refered to as Knowledge Discovery in Databases (KDD), a process that takes place within the data mining field. Successful decision making requires as much information as possible. In this sense, data mining techniques can help us in obtaining information from large data sets for better decision making [11].

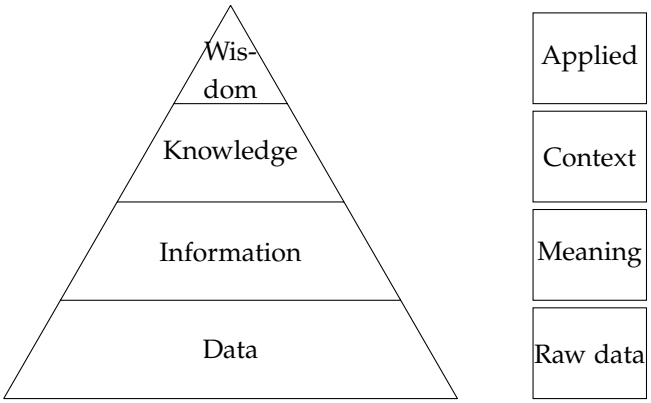


Figure 2.1: The DIKW pyramid.

Data mining techniques can be applied to real data from different domains, such as social sciences [12], medicine [13], economics [14], and business [15]. The precondition is that there must be a sufficiently large amount of structured and reliable information. This information may be sensitive, such as medical data from a patient or, in this research, financial information of a company. For that reason, it is important to consider potential privacy concerns. One practise used to deal with this issue is, for example, anonymizing data. Moreover, privacy concerns can be mitigated by gathering as less information as possible. The data used in this research was already gathered by the KvK.

2.3.1 Machine learning

Data mining provides insight about data. Applying these insights to learn something about new data, we speak of predictive analysis. Machine learning is a commonly used data mining technique for predictive analysis. An algorithm is trained to find patterns in given data and then applied to unseen data. There are

different techniques and domains in machine learning. Two categories of techniques, supervised learning and unsupervised learning, are explained hereafter [11].

The major difference between supervised learning and unsupervised learning is the presence of a labeled data set. In supervised learning problems, we use a data set where the target attribute (to be predicted attribute) is labeled (interpreted, e.g. 'fraud' or 'no fraud'). Machine learning algorithms search for patterns between attributes and the label of the target. When these patterns are deployed onto new data, a prediction of the target attribute is made. Classification and regression are subcategories of supervised learning. Information can still be obtained from non-interpreted data sets using unsupervised learning. A technique of unsupervised learning is 'clustering', where data points that have similar attribute values are forming clusters. In the context of the previous example: fraudulent firms would be forming their own cluster and the instances of the data that represent non fraudulent firms would not be in this cluster.

2.3.2 Class imbalance

Class imbalance refers to the problem of an unequal distribution (imbalance) of the values of the target attribute (class), caused by at least one class representing a minority of the data. Class imbalance is a common problem in real world classification problems such as disease prediction and fraud prediction, where the abnormal and thus more interesting class is less present in the data set. These are harder to predict since commonly used classification algorithms are likely to incline towards the most occurring class. This has to do with the fact that these algorithms focus on minimizing the total error rate instead of paying extra focus on the (interesting) minority class. This can be clarified by the following example.

We want to predict whether a person has a rare disease that does not manifest an illness. The data set contains 100,000 medical records of people without the disease and 10 medical records of people with the disease. Obviously, we want to classify these persons as carrier of the disease. Healthy people that are marked as carrier will still be declared healthy after examination in the hospital. We have two predictive models. Model A predicts 3 out of 10 individual with the disease and 50 of the 100,000 healthy people as carrier. Model B predicts 8 out of 10 and 100 out of 100,000, respectively. Commonly used algorithms will prefer model A over model B since it has a lower total error rate (57 faulty predictions instead of 102 faulty predictions). In this particular case we would actually prefer model B since it was able to classify more persons with the disease and false positives can be picked out after examination in the hospital. This leads us to choose a other measurement than accuracy, as discussed in Section 2.3.3.

2.3.3 Performance measurements

Several metrics can be computed for determining the performance of a model, such as the models in the aforementioned example. Naturally, these metrics should be representative for the accuracy of the predictive model in practice, and thereby the performance on an unknown data set. Cross validation is a model validation technique that assesses the predictive ability of a model on data that is not used in constructing the

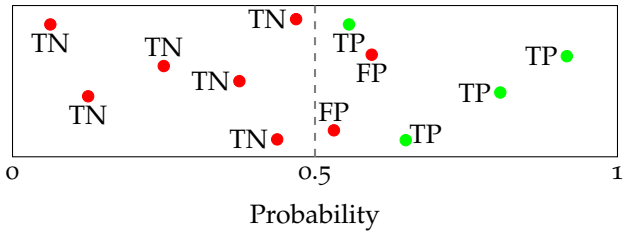
model. This is realized by using a distinct part of the data set during training, and using the remaining data to test the model by making predictions on this data and compare them with the actual data. A commonly used application is k-fold cross validation, partitioning the data set in k parts: $k-1$ part are used for training and one part is used for testing. This process is repeated k times, with different partitions for each iteration. Predictions based on the test set enables the computation and representation of results in a confusion matrix as shown in Figure 2.2.

		Predicted	
		True	False
Actual	True	TP	FN
	False	FP	TN

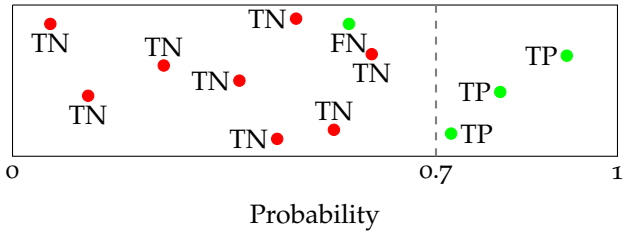
Figure 2.2: **Confusion matrix.** Each cell represents the number of instances with that classification result: true positives (TP) are correctly classified as 'True', false negatives (FN) are wrongly classified as 'False', false positives (FP) are wrongly classified as 'True', and true negatives (TN) are correctly classified as 'False'.

Several performance metrics are derived from these confusion matrices. The commonly known accuracy is computed by taking the good classifications (TP and TN) divided by all instances (TP, FN, FP, TN). Dividing the true positives by all positive instances (FP and FN) results in the true positive rate (TPR), and the false positive rate (FPR) is the results of the division of false positives by all negative instances (FN and TN).

Models do not output absolute values such as 'True' or 'False'. Instead, a probability is given that should be interpreted as "instance i has probability 0.8 that it is 'True'". A threshold value divides the cases in the absolute values. For the given example, a threshold of 0.5 would mark i as 'True', while a threshold of 0.9 would mark i as 'False'. Changing the threshold value can yield a better result, as illustrated with Figure 2.3.



(a) **Threshold value 0.5 results in two misclassifications.**



(b) **Threshold value 0.7 results in one misclassification**

Figure 2.3: **Adjusting threshold values**. Dots represent instances and are negative (red) or positive (green). A predictive model returns probabilities for each instance between 0 and 1. The labels represent the result of classification after taking the threshold value into account: instances on the left of the threshold are marked negative and instances on the right of the threshold are marked as positive. The second example shows that increasing the threshold value can yield a lower rate of misclassification.

As discussed in Section 2.3.2 with the example of the rare disease, using the total error rate or the average accuracy is not a desired performance metric for data sets with a high class imbalance. An alternative is a performance metric that is computed for each class. This is done with a receiver operating characteristic (ROC) curve and the Area Under the Curve (AUC). ROC curves have proven to be more useful than accuracy in this setting [16]. A ROC curve is the graphical representation of the TPR against the FPR for threshold values between 0 and 1. It is represented in Figure 2.4. The diagonal line the ROC curve that would be obtained by randomly guessing the class. Good classifiers are placed closer to the upper left corner. In Figure 2.4, in terms of quality of classifiers we can state that $C > B > A$.

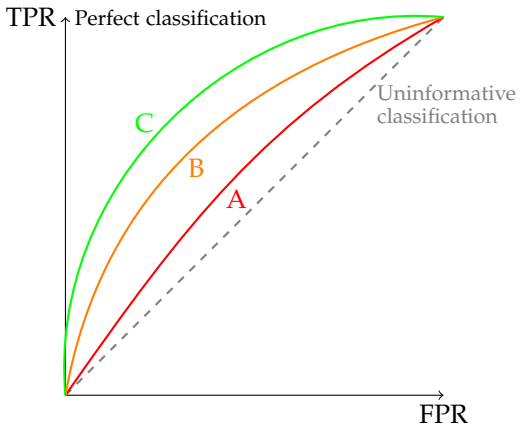


Figure 2.4: **ROC curve.** A receiver operating characteristic curve is a plot of the true positive rate (TPR) against the false positive rate (FPR) for particular thresholds. Random guessing will give the dotted ROC curve (uninformative classifier). ROC curve *C* represents the best classifier in this comparison, since it is the closest to the desired classification point in the upper left corner.

Although multi-class ROC curves do exist [17], the graphs are mostly used for representing a two-class problem. In the case of a classification with multiple classes, a ROC curve represents the ability of a classifier to predict one class versus all other classes. Therefore, binarization of the target attribute is a supplementary preprocessing step. All classes are transformed into binary columns. A '1' on row m of binary column n encodes that row m corresponds with the n^{th} class. For each class n to be trained, binary column n is selected. For this column, a '1' corresponds with class n and a '0' corresponds with any other class than n.

For evaluating the predictive ability of classifiers without graphical representation, AUC can be used. It is computed by taking the integral of the curve, giving a value between 0 and 1 (AUC $\in [0, 1]$) denoting the area under the curve. A perfect classifier has an AUC of 1.0. An uninformative classifier such as represented by the diagonal line in Figure 2.4 will give a AUC of 0.5.

Chapter 3

Related Work

An abundance of research has been done in regard to the application of data mining techniques over financial statements, generally focused on fraud prediction. For example, [18], [19], [20], [21], [22], [23] successfully modeled the problem whether or not a filed financial financial statement is a Fraudulent Financial Statement (FFS) as a supervised learning task. An overview of the field is provided by Sharma et al. [2]. They put effort in the categorization of 35 papers that researched FFS and other types of accounting fraud from the period between 1995 and 2012. The conclusion of the study is that neural networks have a good perform on classification problems, although they lack interpretability. Secondly, the researchers contributed with a framework for financial accounting fraud detection, emphasizing the use of sources other than financial statements alone. Another regularly recurring problem is predicting future revenue and ultimately bankruptcy of a company [3], [24], [25], [26]. Among others, Zieba et al. [3] implemented a successful approach. Whether a company would go bankrupt in the following five years was modeled as a classification problem. The problem is best assessed by using an ensemble of boosted trees (Extreme Gradient Boosting) and the addition of new features, by performing arithmetic operations on all possible combinations of features.

A broad range of algorithms and methods have been applied and thoroughly compared by the previously mentioned studies. One of the recurring classifiers are decision trees [3], [19], [23], [24], [26], [27], mostly as part of an ensemble [3], [23], [24], [27]. In the three studies that compared neural networks with other classifiers [18], [19], [26], the neural networks were the best in two studies [18], [26]. The main disadvantage of neural networks is that they act as a black box and therefore lack explainability. This in contrast to decision trees that are easier to interpret, without compromising much on performance [19], [26].

The aforementioned researches mainly used small data sets, shown here in four categories. Kirkos et al. [19] used the smallest data set (< 100). Between 100 and 1,000 instances were used in five studies [18], [20], [21], [23], [24] and 1,000 - 10,000 instances were used by four [22], [25], [26], [27]. One research stood out, Sharma et al. [2] used more than 10,000 records. The number of features used differ from 18 [26] to 65 [25].

Ten of the eleven studies mentioned earlier used financial ratios as part of their features. In some cases the financial ratios formed the majority of the variables [3], [19]. Five studies confirmed the use of ratios for

financial statement analysis as useful [19], [20], [21], [23], [25]. Kirkos et al. [19] and Kotsiantis et al. [23] even emphasized that "a relatively small list of financial ratios largely determines the classification results".

Based on the analysis in this chapter, the following can be established. First, data mining techniques have proven to be successful in several classification problems, such as fraud and bankruptcy. Second, the performance of available algorithms has been widely investigated. Third, overall small datasets were used for training the classification models. At last, financial ratios have proven to be valuable attributes in classification problems in previous research.

In comparison to these conclusions, the contribution of this research is twofold. First, from an economics point of view, investigating a new target attribute contributes to the field and enables application such as data completion, and error and fraud detection. To the best of our knowledge, no research has been done in predicting the sector in which a company is active. Since this target attribute has not yet been investigated, the impact of the attributes on the quality of prediction must be researched. From a computer science point of view, this study contributes by expanding the application of data mining techniques. A significant amount of effort is put in comparing various methods and benchmarking algorithms for classification problems. The emphasis of this work will therefore not be on the algorithmic part of machine learning, but rather focus on assessing the ability to perform sector prediction by applying data mining techniques on financial statements.

Chapter 4

Data and Methods

The data, materials and methods needed for the supervised classification problem that we are dealing with are described in this chapter. The structure is as follows: Section 4.1 describes characteristics of the data set. In Section 4.2 the steps undertaken during preprocessing are shown. Different approaches to tackle class imbalance are discussed in Section 4.3. Section 4.4 consists of a consideration of the classification algorithm to apply. Finally, Section 4.5 reports the cross validation methods and performance metric used.

4.1 Data set

The version of the open data set we use [5] contains 1,517,400 anonymous financial statements, distributed over the years 2015-2018, displayed in Figure 4.1. All statements are in XBRL format and are stored in separate XML files. In total there are 158 different attributes and, on average, a statement contains 15 attributes. The attributes that occur the most are attributes from balance sheets, see Table 4.1 where the column 'Frequency' denotes the relative frequency of an attribute in the complete data set.

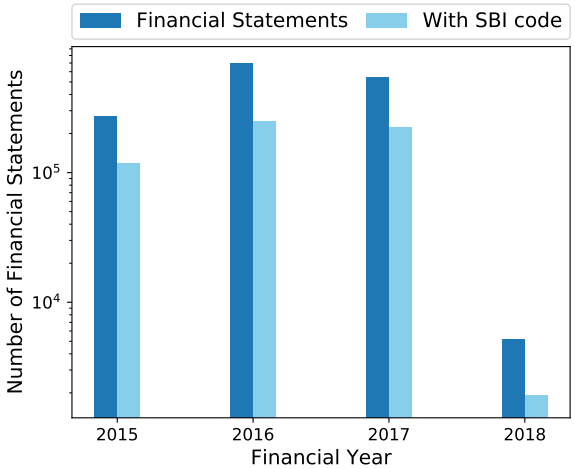


Figure 4.1: Total number of financial statements and the number of financial statements with SBI code, per financial year.

Table 4.1: **Top 35 most frequent features** and their relative frequency of the 158 features in the complete data set. Features in bold are used to compute financial ratios. A list with all features can be found in Appendix A.

x_i	Feature name	Frequency
x_1	FinancialYear	1.000
x_2	DocumentAdoptionDate	1.000
x_3	BalanceSheetBeforeAfterAppropriationResults	0.998
x_4	**Assets**	0.995
x_5	**Equity**	0.993
x_6	EquityAndLiabilities	0.993
x_7	**AssetsCurrent**	0.965
x_8	AssetsNoncurrent	0.829
x_9	ShareCapital	0.643
x_{10}	**LiabilitiesCurrent**	0.622
x_{11}	**Receivables**	0.595
x_{12}	**CashAndCashEquivalents**	0.579
x_{13}	ReservesOther	0.541
x_{14}	SbiBusinessCode	0.391
x_{15}	Provisions	0.387
x_{16}	FinancialAssets	0.369
x_{17}	PropertyPlantEquipment	0.369
x_{18}	AssetsCurrentOther	0.334
x_{19}	**Liabilities**	0.330
x_{20}	AssetsNoncurrentOther	0.286
x_{21}	LiabilitiesNoncurrent	0.214
x_{22}	Inventories	0.132
x_{23}	SharePremium	0.125
x_{24}	RetainedEarnings	0.117
x_{25}	IntangibleAssets	0.079
x_{26}	ResultForTheYear	0.059
x_{27}	LegalReserves	0.045
x_{28}	SecuritiesCurrent	0.031
x_{29}	ConstructionContractsAssets	0.023
x_{30}	RevaluationReserve	0.022
x_{31}	LegalStatutoryReserves	0.018
x_{32}	CostsIncorporationShareIssue	0.018
x_{33}	CalledUpShareCapital	0.016
x_{34}	InvestmentProperties	0.012
x_{35}	**Securities**	0.010

39% of the 1,517,400 financial statements were deposited with a SBI code. There are a total of 923 unique SBI codes in the data set. The distribution of the codes is not uniform. Table 4.2 shows that number 1 has a higher frequency than numbers 2 to 9 combined. This imbalance, in combination with practical problems of reporting a performance for each SBI code, motivates to use a different level of the SBI hierarchy. Instead of using the activity (three to five level SBI code) a higher level such as the department or section will be used. Table 2.2 in Section 2.2 shows that this leads to a significant reduction of the number of distinct codes, at the cost of detail of activity. We refer to this process as class reduction.

Table 4.2: Top 10 most occurring unique SBI codes and their relative frequency.

#	SBI code	Frequency
1	6420	0.367
2	70102	0.053
3	64303	0.040
4	70221	0.039
5	7112	0.015
6	6832	0.012
7	6201	0.012
8	6810	0.012
9	4120	0.011
10	68204	0.010

4.2 Data preprocessing

Preprocessing is described in four steps. The data set contains the financial statements in separate XML files. The first step is to copy the data from the separate files into one tabular file. This results in tabular file with 158 columns and 1,517,400 rows, where each row represent a financial statement with its corresponding attributes.

During the second step, the number of classes is reduced. As described in Section 4.1, for practical reasons we choose to use a SBI coding that is of a higher hierarchical level. The three to five digit codes (activity) are transformed into two digit codes (departments). Then, the two digit codes are transformed into one character code (sections) except for the section with the highest frequency. This section makes distinctions between the departments and activities within it, by adding a postfix. The split of this section into four subgroups further reduces class imbalance. This process is illustrated in Figure 4.2. The final classes we use as target for our classification are listed in Table 4.3

x_i	Sbi...	x_i
...	18129	...
...	49393	...
...	64191	...
...	66192	...
...	16239	...

$\Longrightarrow$

x_i	Sbi...	x_i
...	**18**	...
...	49	...
...	64	...
...	66	...
...	**16**	...

$\Longrightarrow$

x_i	SbiBusinessCode	x_i
...	**C**	...
...	H	...
...	K1 *	...
...	K4 *	...
...	**C**	...

Figure 4.2: **Class reduction**. The activities (left) are aggregated into their departments (middle) and then into sections (right). Note that different departments can be in the same section (marked bold). Only the majority makes a distinction between departments and activities by adding a postfix (market with $*$).

Table 4.3: Classes and their relative frequency.

Class	Description	Frequency
A	Agriculture, forestry and fishing	0.01474
B	Mining and quarrying	0.00026
C	Industry	0.03661
D	Production, distribution and trade of electricity, natural gas, steam and cooled air	0.00204
E	Water extraction and distribution; waste and waste water management and remediation	0.00190
F	Construction industry	0.04202
G	Wholesale and retail trade; repair of cars	0.11077
H	Transport and storage	0.01987
I	Accommodation, meals and drinks	0.01592
J	Information and communication	0.03192
K1	Financial institutions - Other financial institutions	0.03051
K2	Financial institutions - Financial holdings	0.36828
K3	Financial institutions - Investment institutions	0.05301
K4	Financial institutions - Insurance companies	0.00081
L	Rental of and trade in real estate	0.04390
M	Advice, research and other specialist business services	0.18743
O	Public administration, government services and compulsory social insurance	0.00004
P	Education	0.00593
Q	Health and welfare care	0.02002
R	Culture, sport and recreation	0.00899
S	Other services	0.00490
T	Households as an employer; goods and services by households for their own use	0.00001
U	Extraterritorial organizations and bodies	0.00002

During the third step, financial ratios are computed and added to the set of features. Since mostly attributes from balance sheets are present in the data set, only five of all available financial ratios could be computed. They indicate the leverage and liquidity of an organization. The following ratios are computed, with their relative frequency in the complete data set between brackets.

$$Current\ Ratio = \frac{Current\ Assets}{Current\ Liabilities} \Longrightarrow x_{159} = \frac{x_7}{x_{10}} \quad (0.6049)$$

$$Debt\ Ratio = \frac{Total\ Liabilities}{Total\ Assets} \Longrightarrow x_{160} = \frac{x_{19}}{x_4} \quad (0.3282)$$

$$Debt\ to\ Equity\ Ratio = \frac{Total\ Liabilities}{Equity} \Longrightarrow x_{161} = \frac{x_{19}}{x_5} \quad (0.3288)$$

$$Cash\ Ratio = \frac{Cash + Marketable\ Securities}{Current\ Liabilities} \Longrightarrow x_{162} = \frac{x_{12} + x_{35}}{x_{10}} \quad (0.0096)$$

$$Quick\ Ratio = \frac{Cash + Marketable\ Securities + Accounts\ Receivables}{Current\ Liabilities} \Longrightarrow x_{163} = \frac{x_{12} + x_{35} + x_{11}}{x_{10}} \quad (0.0093)$$

After feature creation, 2 features with a descriptive function of a financial statement (x_1 and x_2) are dropped and the target (x_{39}) is extracted. This leaves 155 features. The data set without FR consists of 155 features and the data set with FR consists of 160 features.

An average statement contains 15 attributes and therefore for each row in the tabular file, on average 143 columns are empty. The last step of preprocessing deals with missing data. Since the presence of an attribute (whether a company uses this attribute in financial statements) is characteristic for a company and its legal form, missing data can not just be removed or replaced. A missing-indicator will be added to encode the missingness, as commonly used approach, for example in data science community Kaggle [28]. First, the columns are duplicated. In the first set of columns, the missing data is filled with zero values. In the second set of columns, 0's and 1's represent whether that an attribute was present (1) or not (0) in the corresponding record. This process is illustrated in Figure 4.2.

x_1	x_2	x_3	x_4
...	NaN	NaN	...
NaN	...	NaN	...
NaN	NaN	...	NaN
...	...	...	NaN

$\Longrightarrow$

x_1	x_2	x_3	x_4	M_x_1	M_x_2	M_x_3	M_x_4
...	0	0	...	1	0	0	1
0	...	0	...	0	1	0	1
0	0	...	0	0	0	1	0
...	...	...	0	1	1	1	0

Figure 4.3: **Encoding missingness.** Transformation from original data to new table without missing data ($x_1, x_2, \ldots, x_{160}$) and missing-indicator ($M\_x_1, M\_x_2, \ldots, M\_x_{160}$). 'NaN' indicate cells with missing values, where cells with ... indicate cells with values.

The missing-indicator is used to encode missingness so that patterns in missing data may be used in conjunction with patterns in observed data. The downside of this technique is the increase of dimensionality and thus the additional computing time.

4.3 Handling class imbalance

Table 4.3 shows that there is an imbalance in the classes in our data set. The approaches to tackle the class imbalance problem come in two variations: 'cost sensitive learning' and 'sampling approach'. The first adds weights to instances, with a higher weight for the instances of the minority class so that they contribute more into the total error. The sampling approach removes or adds samples to the data set to obtain a more equal distribution of the classes. This study applies five class imbalance approaches:

1. **Undersampling**: a sampling approach that randomly removes instances of the majority class.
2. **Oversampling**: a sampling approach that randomly adds extra instances of the minority class.
3. **SMOTE**: stands for Synthetic Minority Over-sampling Technique. Instead of duplicating instance i, a new instance is synthesized by computing its features as slight variations of the features of the instances that are similar to i, based on sharing a cluster i. This leads to better generalization by decision trees. [29]
4. **Weighted classes**: a cost sensitive learning approach where the weight of a class is inversely proportional to their frequency.
5. **Custom approach**: this work will present a custom approach, named Performer vs Rest (PvsR). A class is removed after training to automatically balance remaining data. The order of training is based on the performance on the class as trained without a class imbalance approach. This technique is based on the idea that more prevalent classes tend to be more easily classifiable and training worst performing classes (classes with a lower frequency) on a more balanced data set with fewer classes yield better performance. The implementation of the PvsR approach is described in Section 5.1.

4.4 Classification algorithm

The primary criterion for the algorithm to choose is decent explainability in order to answer our sub questions. Although neural networks can have good performance, they are generally known, and emphasized in previous work, for their poor explainability. On the other hand, decision trees are valued at this point. As shown in previous work [19], [26], decision trees compete well with neural networks in settings similar to ours. There are algorithms that produce their output based on one decision tree (for example the C4.5 algorithm) or based on multiple trees, an ensemble method). Ensemble methods have a higher performance [3]. Two techniques implementing the ensemble method for decision trees are bagging and boosting. With bagging, each tree is generated from a different subset of the data. With boosting, each tree is generated sequentially and the goal of each tree is to reduce the error from the previous generated tree. Bagging generally outperforms boosting on data sets with outliers and class imbalance [30]. The latter occurs in this setting. Therefore, we will use a classifier based on the ensemble bagging method.

Random forest [31] is a bagging ensemble method by using a random subset of the data to generate a tree. After training a large number of trees, each tree produces an output. In classification problems, the most

occurring class from the outputs of the individual trees will be the output of the random forest. In comparison to other bagging ensemble methods it less sensitive to overfitting on outliers [32].

In addition to good generalization, there are several other advantages to mention. First, random forest is easily applied due to automatic scaling and selection of features. Moreover, there is no need for data transformation, or extensive hyperparameter tuning. Secondly, random forest works well with high dimensional data. Third, generated forests can be saved for future deployment on new data. At last, random forest has decent runtime by default, and options for parallelization to improve this. These advantages combined lead to the decision of using a random forest classifier in this study.

4.5 Cross validation

Insights such as feature importance and the minimal descriptive length are only relevant if these insights are produced by proper classifiers. The performance of the aforementioned approaches are measured and compared with ROC curves and AUC as performance metric, as described hereafter.

Each aforementioned approach is trained and tested separately. 10-fold cross validation as explained in Section 2.3.3 is used to average out randomness. A validation set will not be used since tuning of hyperparameters is out of the scope of this research. For each iteration of the 10-fold cross validation, a ROC curve with its corresponding AUC is computed. The combination of all folds will give the AUC for that particular class, including a standard deviation. Combining the AUCs of all classes by computing the mean will give the performance metric for the applied approach. All classes weigh equally during this computation.

The train sets and test sets identical for all approaches, except the custom approach. This approach starts with the same train and test sets but are adjusted after each class. After a class is trained, the positive instances of that class are removed from the train set. All positive predicted instances are removed from the test set after testing. Positive predictions are removed instead of deterministic data to create a realistic testing scenario.

Chapter 5

Experiments

In this chapter, the experiments are outlined. Section 5.1 describes the experimental setup. The results are shown in Section 5.2 and discussed in Section 5.3.

5.1 Experimental setup

The complete implementation of this experiment is performed in *Python*. Additional packages were used for specific tasks. During preprocessing, the transformation of XML files to CSV files is realized with XMLtree. Data visualisations were created with Seaborn and Matplotlib. Machine learning algorithms and measures were supplied by Scikit-learn [33]. The class imbalance approaches 1 to 3 (see Section 4.3) used the imbalance-learn module [34]. Numpy and Pandas provided datastructures for the data set.

The implementation of the random forest algorithm is realized by the RandomForestClassifier (RFC), wrapped in an OneVsRestClassifier for compatibility with two-class ROC curves. Both objects were initialized with default parameters. This resulted in 100 trees used by RFC for one model. For reproducibility, the random seed value was set to 42. These conditions apply to all objects initialized during this experiment.

The train sets (X_{train}) and test sets (X_{test}) are stratified using *StratifiedKFold*, whereby the class frequency in each of the train/test sets are a reflection of the whole population. Ten train sets and ten test sets are determined once and used over the complete course of the experiment. Roughly 534,000 instances are part of each train set and 59,000 instances part of each test set. For each fold of each class, the classifier is trained (using X_{train}) and tested (using X_{test}). The test results are used to compute performance for each fold. The means of the 10 folds are used to compute the performance of the class. Then, the mean of all classes is used to compute the performance of the complete method. Wilcoxon signed-rank tests [35] are applied to determine whether the differences between the methods are statistical significant. Hereby, we do not directly assume that an approach is better is there is a slight improvement in performance. Arrays with the performance per fold for each class are compared to assure that the arrays are not too short for Wilcoxon signed-rank tests.

The class imbalance approaches as described in Section 4.3 are implemented as follows:

1. **RUS**: an undersampling approach. Implemented by preprocessing X_{train} and X_{test} for each fold using RandomUnderSampler. The default sampling strategy implies that all classes except the minority class are undersampled.
2. **ROS**: an oversampling approach. Same implementation as RUS, but with RandomOverSampler. The default sampling strategy implies that all classes are oversampled, except the majority class.
3. **SMOTE**: a synthetic oversampling approach. Same implementation as ROS, but with synthetic oversampling (SMOTE) instead of normal oversampling (RandomOverSampler).
4. **CSL**: a cost sensitive learning approach where the weight of a class is inversely proportional to their frequency. Implemented by setting parameter 'class_weight' of RandomForestClassifier to 'balanced'.
5. **PvsR**: a custom approach that is not part of a *Python* package. The performance per class is determined by normal training/testing iteration. Hereafter, the class with the highest AUC is trained and tested. After training, the instances from this class are removed from the training sets. All positive predictions (both true positives and false positives) are removed from the test sets. After the first class is trained and tested, the second best performing class is trained on remaining data and then tested on all data except the positive predictions of previous class(es). This process repeats until all classes except one are processed. For each class and fold, the optimal threshold value is determined w ith the Youden Index and applied to arbitrate the cut-off point of positive predictions and therewith limit the impact of previous classifier's decisions in the current classifiers quality. An overview is shown in Listing 5.1.

```
# Regular classification to determine the ranking
for each class
    for each fold
        train
        test
        compute performance of fold
    determine performance of class
sort performance of classes

# Performer vs Rest
for range(1, classes)
    pick best performing of remaining classes
    for each fold
        load train index, remove indices from previous classes
        load test index, remove indices from previous  positive predictions
        train
        test
        determine train indices to remove next iterations
        determine optimal threshold value
        determine positive predictions with optimal threshold
        determine test indices to remove next iterations
        compute performance of fold
    compute performance of class
compute performance of all classes
```

Listing 5.1: The outline of the 'Performer vs Rest' approach.

The experiments are conducted in the following manner. First, a baseline is set. Then, the class imbalance approaches are applied and compared with the baseline. Third, the impact of financial ratios (FR) is deter-

mined by rerunning the baseline and all class imbalance approaches with a data set containing the financial ratios. The feature importance (FI) is retrieved from the classifiers train during the best performing approach. Finally, plotting FI against the number of features visualizes the optimal number of features to use.

5.2 Results

We generated the ROC curves and compute the performance for each approach. Figure 5.1 shows the performance of the baseline. Figure 5.2 illustrates the performance of the class imbalance approaches and Figure 5.3 shows the performance of all approaches with the financial ratios in the data set. Table 5.1 summarizes.

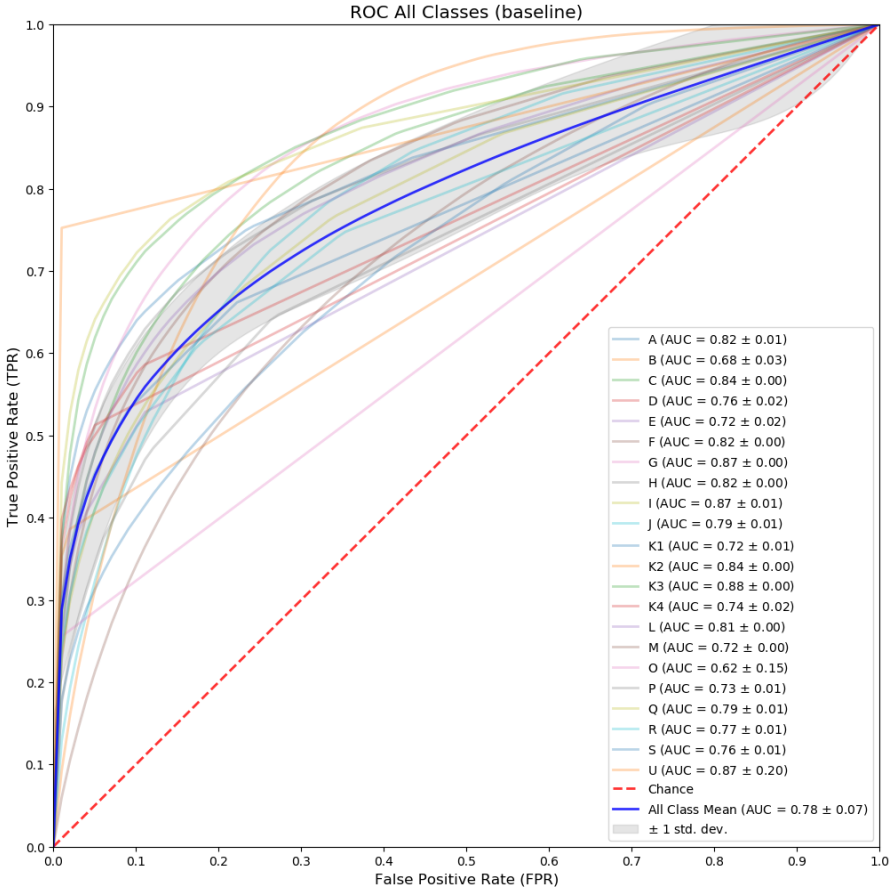
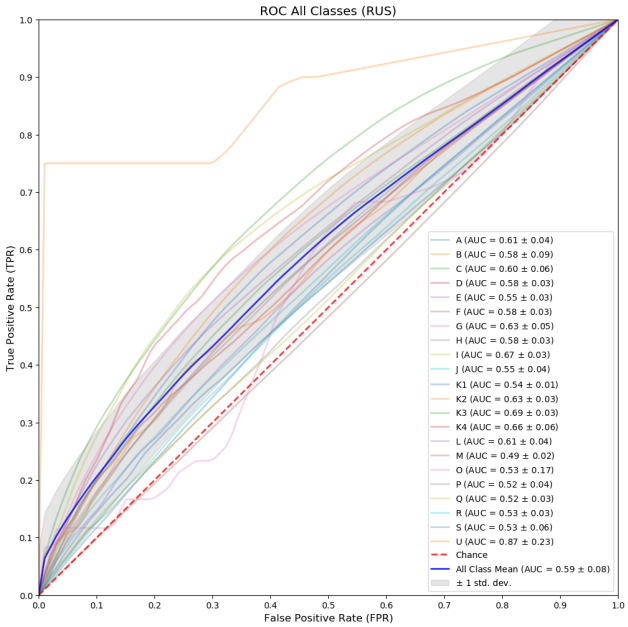


Figure 5.1: **Base performance**. The performance of each class is drawn with a ROC curve. All class ROC is represented by the blue line, with the standard deviation in gray. The performance is an AUC of 0.78 with a standard deviation of 0.07.

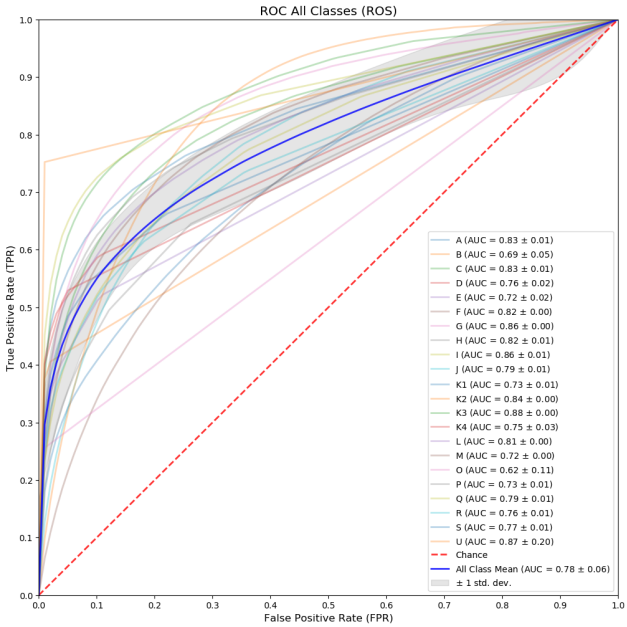
Table 5.1: **Performance per approach.** Performance is the AUC ± standard deviation. Wilcoxon signed-ranked tests were applied to compute the p-values: class imbalance approaches are compared to the baseline, FR approaches are compared to approaches without FR. p-values below 0.05 are considered to be indicate a significant change in the distribution of performance.

Class imbalance approach	Without FR		With FR	
	Performance	p-value	Performance	p-value
none (baseline)	0.78 ± 0.08	-	0.78 ± 0.07	0.558
RUS	0.59 ± 0.08	0.000	0.59 ± 0.08	0.000
ROS	0.78 ± 0.06	0.011	0.78 ± 0.06	0.267
SMOTE	0.79 ± 0.05	0.014	0.79 ± 0.06	0.274
CSL	0.78 ± 0.07	0.216	0.78 ± 0.07	0.904
PvsR	0.78 ± 0.08	0.026	0.78 ± 0.08	0.170

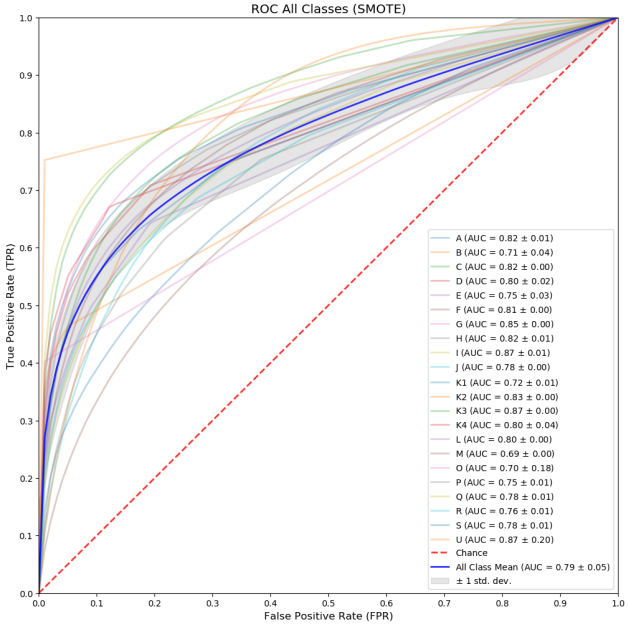
(a) **RUS** (AUC = 0.59 ±0.08)



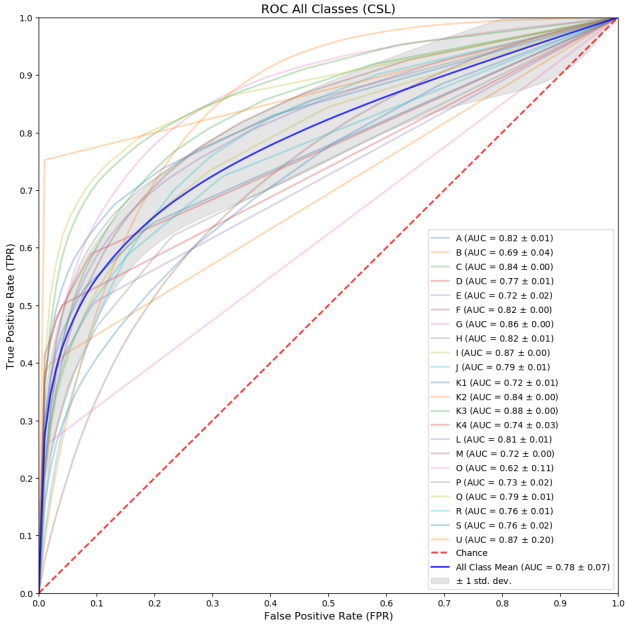
(b) **ROS** (AUC = 0.78 ±0.06)



(c) **SMOTE** (AUC = 0.79 ±0.05)



(d) **CSL** (AUC = 0.78 ±0.07)



(e) **PvsR** (AUC = 0.78 ±0.08)

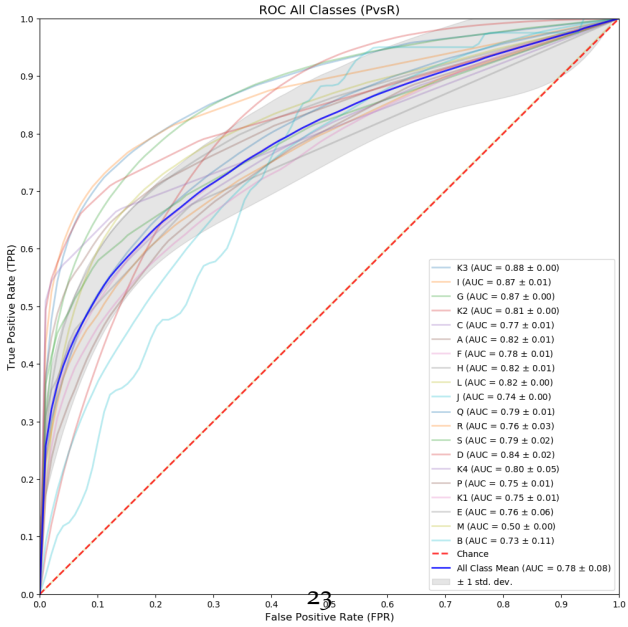
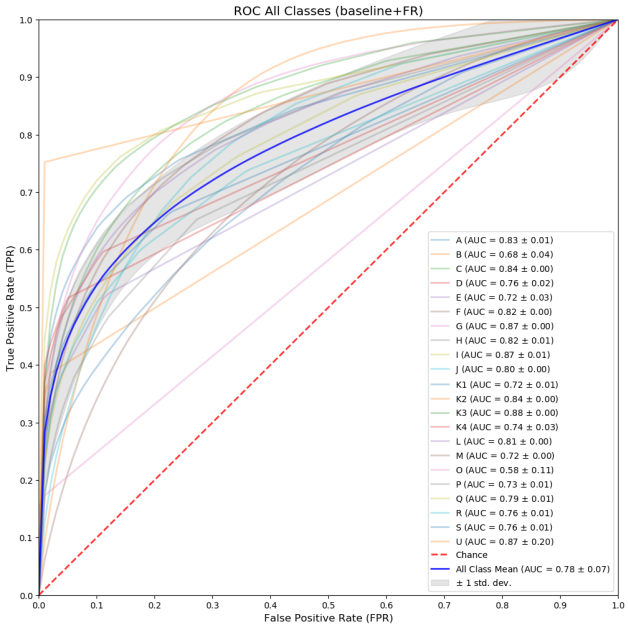
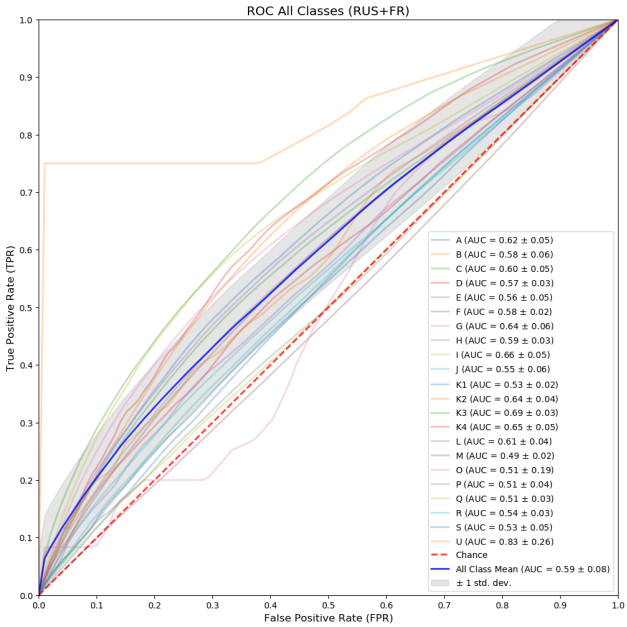


Figure 5.2: ROC curves of the five class imbalance approaches.

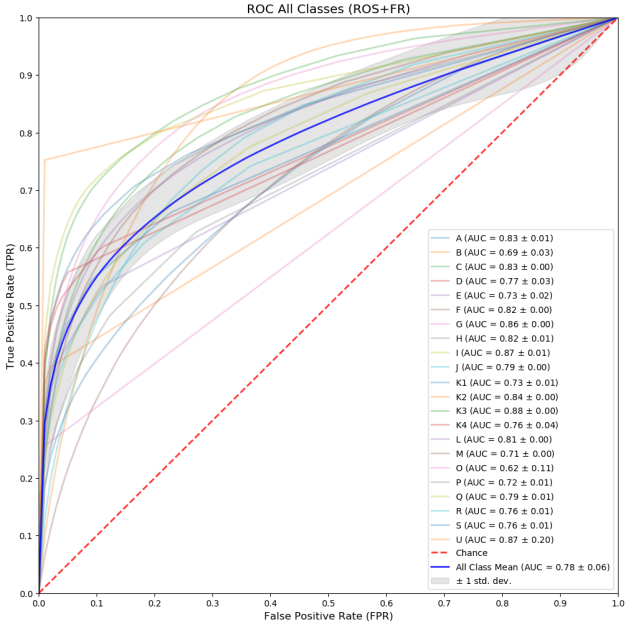
(a) **Baseline + FR** (AUC = 0.78 ±0.07)



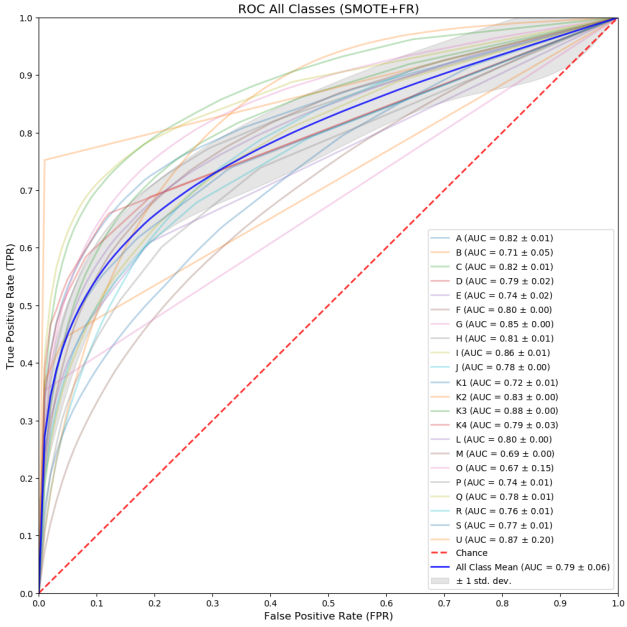
(b) **RUS + FR** (AUC = 0.59 ±0.08)



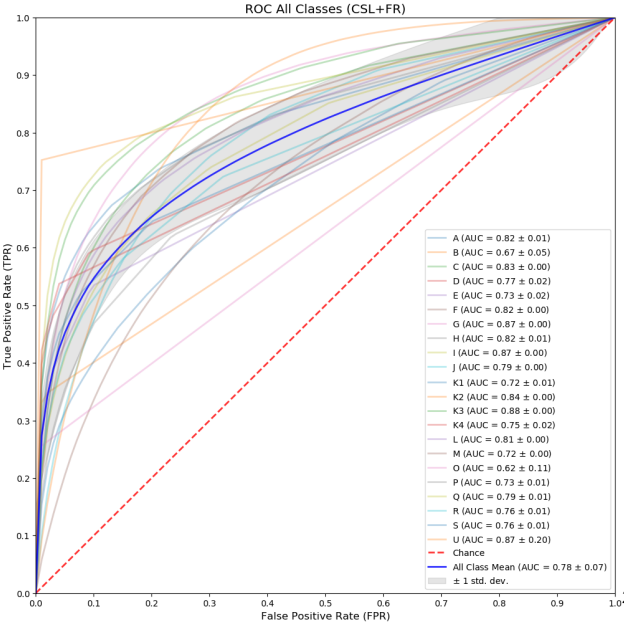
(c) **ROS + FR** (AUC = 0.78 ±0.06)



(d) **SMOTE + FR** (AUC = 0.79 ±0.06)



(e) **CSL + FR** (AUC = 0.78 ±0.07)



(f) **PvsR + FR** (AUC = 0.78 ±0.08)

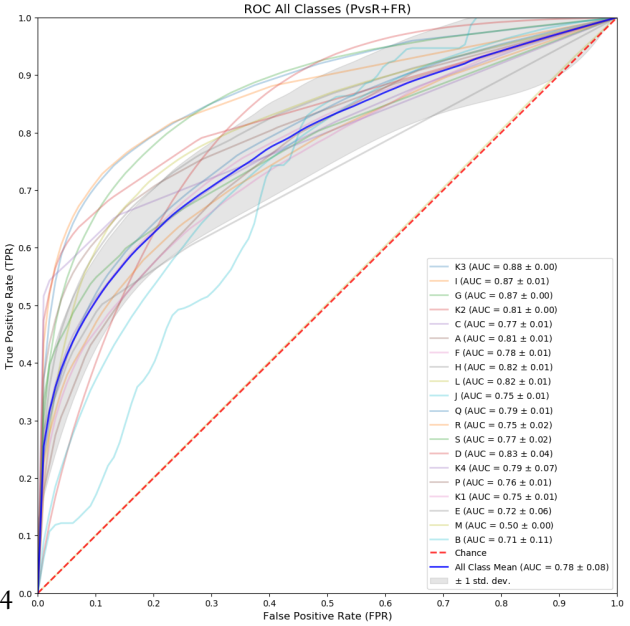


Figure 5.3: ROC curves of all approaches including financial ratios.

SMOTE is the only approach with a significant improvement compared to the baseline. Therefore, SMOTE and SMOTE+FR are used for determining feature importance (FI). The FI of random forest is computed with Gini variable importance measure [36] ($FI \in [0, 1]$), where a higher value denotes a higher importance for that feature. The ten most important features per class for the dataset without financial ratios are presented in Table 5.2 and those for the dataset with financial ratios are presented in Table 5.3. The missing-indicator of feature x_i is represented as M_x_i, financial ratios as $x_{156} - x_{160}$. The feature that occur in the top 10 of SMOTE and SMOTE+FR are shown in Table 5.4 and Table 5.5, respectively. Those features are considered as most important.

Table 5.2: **Feature importance**. Ten most important feature for each class in the dataset without financial ratios.

Class	x # 1	x # 2	x # 3	x # 4	x # 5	x # 6	x # 7	x # 8	x # 9	x # 10
A	x_{135}	x_{22}	x_{20}	x_{33}	x_{52}	M_x_{74}	M_x_{34}	x_{148}	x_{105}	x_{129}
B	x_{135}	x_{22}	x_{20}	x_{52}	x_{33}	x_{105}	x_{148}	M_x_{65}	x_{110}	x_{111}
C	x_{135}	M_x_{74}	x_{33}	x_{22}	x_{20}	x_{52}	x_{129}	M_x_{34}	x_{105}	x_{92}
D	x_{135}	x_{22}	x_{20}	x_{33}	x_{52}	M_x_{65}	M_x_{34}	x_{105}	x_{148}	x_{129}
E	x_{135}	x_{20}	x_{22}	x_{33}	x_{52}	M_x_{34}	x_{105}	x_{129}	x_{148}	x_{92}
F	x_{135}	x_{22}	x_{20}	x_{33}	x_{52}	M_x_{34}	x_{129}	x_{105}	x_{92}	x_{148}
G	M_x_{74}	x_{135}	x_{33}	x_{22}	x_{20}	x_{52}	x_{129}	M_x_{34}	x_{105}	x_{92}
H	x_{135}	x_{22}	x_{20}	x_{33}	x_{52}	M_x_{34}	x_{129}	x_{105}	x_{92}	x_{148}
I	M_x_{74}	x_{135}	x_{33}	x_{22}	x_{20}	x_{52}	M_x_{67}	x_{129}	M_x_{34}	M_x_{23}
J	x_{135}	x_{20}	x_{22}	x_{33}	x_{52}	M_x_{34}	x_{105}	M_x_{65}	x_{129}	x_{92}
K1	x_{135}	x_{20}	x_{22}	x_{33}	x_{52}	M_x_{42}	M_x_{65}	x_{105}	M_x_{34}	x_{129}
K2	x_{135}	x_{22}	x_{20}	x_{33}	x_{52}	M_x_{42}	M_x_{65}	x_{111}	M_x_{34}	x_{57}
K3	M_x_{42}	x_{135}	x_{20}	x_{22}	x_{33}	x_{52}	M_x_{121}	M_x_{65}	M_x_{34}	x_{105}
K4	M_x_{42}	x_{135}	x_{22}	x_{20}	x_{33}	x_{52}	M_x_{65}	M_x_{121}	M_x_{34}	x_{105}
L	x_{135}	x_{52}	x_{20}	x_{22}	x_{33}	M_x_{23}	M_x_{65}	M_x_{34}	x_{129}	x_{105}
M	x_{135}	x_{22}	x_{20}	x_{33}	x_{52}	M_x_{34}	x_{105}	x_{129}	x_{92}	M_x_{65}
O	x_{135}	x_{52}	x_{22}	x_{20}	x_{129}	x_{33}	x_{92}	M_x_{34}	x_{50}	x_{148}
P	x_{135}	x_{20}	x_{22}	x_{33}	x_{52}	M_x_{65}	M_x_{34}	x_{105}	x_{129}	x_{92}
Q	x_{135}	x_{22}	x_{20}	x_{33}	x_{52}	M_x_{65}	x_{148}	M_x_{34}	x_{105}	x_{57}
R	x_{135}	x_{22}	x_{20}	x_{33}	x_{52}	x_{105}	M_x_{65}	M_x_{34}	x_{129}	x_{148}
S	x_{135}	x_{22}	x_{20}	x_{33}	x_{52}	M_x_{34}	x_{148}	x_{105}	M_x_{65}	x_{129}
U	x_{22}	x_{20}	x_{33}	x_{57}	M_x_{22}	x_{88}	x_{52}	x_{135}	M_x_{65}	x_{60}

Table 5.3: **Feature importance**. Ten most important feature for each class in the dataset with financial ratios.

Class	x # 1	x # 2	x # 3	x # 4	x # 5	x # 6	x # 7	x # 8	x # 9	x # 10
A	x_{107}	x_{61}	x_{20}	x_{33}	x_{52}	M_x_{49}	x_{126}	M_x_{82}	x_{144}	x_{47}
B	x_{107}	x_{61}	x_{52}	x_{20}	x_{33}	x_{144}	x_{126}	x_{71}	x_{19}	x_{9}
C	M_x_{49}	x_{107}	x_{33}	x_{61}	x_{20}	x_{52}	x_{47}	M_x_{82}	x_{144}	x_{28}
D	x_{107}	x_{20}	x_{61}	x_{33}	x_{52}	x_{144}	x_{126}	M_x_{82}	x_{47}	x_{28}
E	x_{107}	x_{61}	x_{20}	x_{33}	x_{52}	M_x_{82}	x_{144}	x_{47}	x_{126}	x_{28}
F	x_{107}	x_{61}	x_{20}	x_{33}	x_{52}	M_x_{82}	x_{47}	x_{144}	x_{28}	x_{94}
G	M_x_{49}	x_{107}	x_{33}	x_{61}	x_{20}	x_{52}	x_{47}	M_x_{82}	x_{144}	x_{28}
H	x_{107}	x_{61}	x_{20}	x_{33}	x_{52}	x_{47}	M_x_{82}	x_{144}	x_{28}	x_{94}
I	M_x_{49}	x_{33}	x_{107}	x_{61}	x_{20}	x_{52}	M_x_{83}	x_{47}	M_x_{30}	M_x_{82}
J	x_{107}	x_{61}	x_{20}	x_{33}	x_{52}	x_{144}	M_x_{82}	x_{47}	x_{94}	x_{28}
K1	x_{107}	x_{61}	x_{20}	x_{33}	x_{52}	M_x_{30}	x_{144}	M_x_{82}	x_{47}	x_{126}
K2	x_{107}	x_{61}	x_{20}	x_{33}	x_{52}	M_x_{30}	x_{19}	M_x_{82}	x_{144}	x_{126}
K3	M_x_{30}	x_{107}	x_{61}	x_{20}	x_{33}	x_{52}	M_x_{110}	M_x_{82}	x_{28}	x_{144}
K4	M_x_{30}	x_{107}	x_{61}	x_{20}	x_{33}	x_{52}	M_x_{110}	x_{28}	M_x_{82}	x_{144}
L	x_{52}	x_{61}	x_{107}	x_{20}	x_{33}	M_x_{30}	x_{144}	M_x_{82}	x_{47}	x_{28}
M	x_{107}	x_{61}	x_{20}	x_{33}	x_{52}	M_x_{82}	x_{144}	x_{47}	x_{94}	x_{28}
O	x_{107}	x_{52}	x_{47}	x_{61}	x_{20}	x_{33}	x_{94}	M_x_{82}	x_{126}	x_{50}
P	x_{107}	x_{61}	x_{20}	x_{33}	x_{52}	x_{144}	M_x_{82}	x_{47}	x_{28}	x_{94}
Q	x_{107}	x_{61}	x_{20}	x_{33}	x_{52}	x_{126}	x_{144}	M_x_{82}	x_{47}	x_{28}
R	x_{107}	x_{61}	x_{20}	x_{33}	x_{52}	x_{144}	M_x_{82}	x_{47}	x_{28}	x_{126}
S	x_{107}	x_{61}	x_{20}	x_{33}	x_{52}	x_{126}	x_{144}	M_x_{82}	x_{47}	x_{28}
U	x_{61}	x_{20}	x_{33}	x_{57}	x_{88}	M_x_{67}	x_{60}	x_{142}	x_{107}	x_{31}

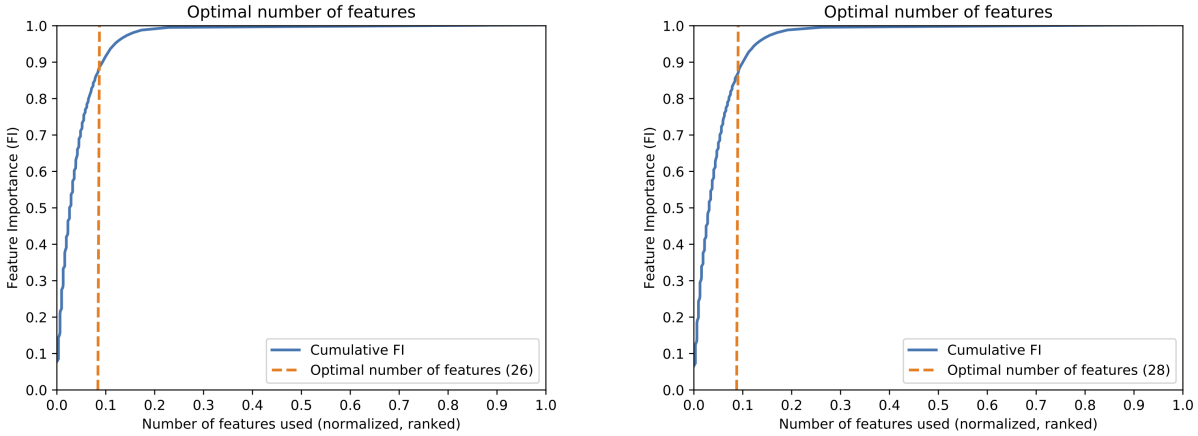
Table 5.4: **Most important features regarding SMOTE.** '# in top 10' is the occurrence in Table 5.2. 'Mean FI' is the average weight of a feature (for all folds and classes), where the cumulative weight of all features equals 1.

x_i	**Description**	**# in top 10**	**Mean FI**
x_{135}	InterestReceivedClassifiedAsInvestingActivities	22	0.075
x_{20}	AssetsNoncurrentOther	22	0.067
x_{22}	Inventories	22	0.067
x_{33}	CalledUpShareCapital	22	0.065
x_{52}	CashFlowFromOperations	22	0.055
M_x_{34}	M_InvestmentProperties	20	0.040
x_{105}	ProceedsSalesIntangibleAssets	18	0.038
x_{129}	PaymentsReclaimingValueAddedTax	16	0.037
M_x_{65}	M_InterestReceivedClassifiedAsOperatingActivities	14	0.037
x_{148}	ResultBeforeTaxOrdinaryActivities	10	0.036
x_{92}	ChangesValueFinancialAssetsSecurities	9	0.034
M_x_{42}	M_SumOfExpenses	4	0.033
M_x_{74}	M_IncreaseDecreasePayablesCreditInstitutions	4	0.025
x_{57}	CashAndCashEquivalentsCashFlow	3	0.029
M_x_{23}	M_SharePremium	2	0.029
x_{114}	LineItemsOtherIncomeStatementReceiptsPaymentsNotConsidered-OperatingActivities	2	0.024
M_x_{121}	M_UnrealisedChangesInValueOfInvestments	2	0.022
x_{50}	CashFlowOperatingActivities	1	0.024
M_x_{67}	M_IncreaseDecreaseProvisions	1	0.016
x_{110}	RevaluationReserveRelease	1	0.014
M_x_{22}	M_Inventories	1	0.005
x_{88}	CashFlowsOperatingActivitiesOther	1	0.004
x_{60}	CashFlowFinancingActivities	1	0.003

Table 5.5: **Most important features regarding SMOTE+FR.** '# in top 10' is the occurrence in Table 5.3. 'Mean FI' is the average weight of a feature (for all folds and classes), where the cumulative weight of all features equals 1.

x_i	**Description**	**# in top 10**	**Mean FI**
x_{107}	InterestPaidClassifiedAsFinancingActivities	22	0.063
x_{61}	InterestPaidClassifiedAsOperatingActivities	22	0.062
x_{20}	AssetsNoncurrentOther	22	0.060
x_{33}	CalledUpShareCapital	22	0.058
x_{52}	CashFlowFromOperations	21	0.050
M_x_{82}	M_IntangibleAssets	20	0.036
x_{144}	PaymentsInstallmentsOperationalLeasing	19	0.036
x_{47}	OperatingResultCashFlow	17	0.035
x_{28}	SecuritiesCurrent	15	0.033
x_{126}	ReceiptsCustomers	10	0.034
x_{94}	ChangesInValueIntangibleAssetsPropertyPlantEquipment	6	0.032
M_x_{30}	M_RevaluationReserve	4	0.030
M_x_{49}	M_IncreaseDecreaseCashAndCashEquivalents	4	0.024
M_x_{26}	M_ResultForTheYear	2	0.028
x_{19}	Liabilities	2	0.023
M_x_{110}	M_RevaluationReserveRelease	2	0.021
x_{57}	CashAndCashEquivalentsCashFlow	1	0.025
x_9	ShareCapital	1	0.024
x_{50}	CashFlowOperatingActivities	1	0.021
M_x_{83}	M_IncreaseDecreaseInConstructionContracts	1	0.015
x_{71}	ReinvestmentReserve	1	0.013
x_{142}	ReceiptsPaymentValueAddedTax	1	0.006
x_{60}	CashFlowFinancingActivities	1	0.003
x_{88}	CashFlowsOperatingActivitiesOther	1	0.003
M_x_{67}	M_IncreaseDecreaseProvisions	1	0.003
x_{31}	StatutoryReserves	1	0.003

After determining feature importance, the optimal number of features was established. Using less features decreases the dimensionality and thereby computation time, although the question remains whether this difference is significant. Figure 5.4 contains the cumulative feature importance against the normalized number of features used, ranked by performance. The cutoff point is based on the point where the derivative is closest to one. From that point, increasing the number of features by 1% leads to less than 1% gain in cumulative feature importance.



(a) Optimal number of features for SMOTE is 26

(b) Optimal number of features for SMOTE + FR is 28

Figure 5.4: **Optimal number of features for SMOTE and SMOTE+FR**. Plot of the cumulative FI against number of features used (normalized and ranked). After the cutoff point (orange), the marginal improvement is below 1.

5.3 Discussion

As baseline, we used the dataset without financial ratios and did not apply a class imbalance approach. We obtained an AUC of 0.78 $\pm$ 0.07, which can be interpreted as a reasonable result. The results vary per class with a minimum AUC of 0.62 (class O: Public administration, government services and compulsory social insurance) and maximum of 0.88 (class K3: Financial institutions - Investment institutions). Despite that we observe a relationship between the frequencies of classes and their performances, we can not conclude that this is the only dependency. This is illustrated by classes U (relative frequency = 0.00002, AUC = 0.87), K2 (relative frequency = 0.36828, AUC = 0.84) and O (relative frequency = 0.00004, AUC = 0.62). Possibly are the characteristics of some classes more expressed by unique attributes in financial statements, yielding a better classification result. Five class imbalance methods were applied to determine whether this would enhance classification. SMOTE increased the AUC with 0.01, which is statistically significant according to the Wilcoxon signed-rank test. Undersampling the data set in the RUS approach caused a large decrease in performance compared to the baseline, while ROS, CSL and PvsR performed comparable to the baseline. Adding five financial ratios to the data set did not lead to an increase in AUC for any of the approaches. Besides, financial ratios feature importance were not of significant enough to appear in the top 10 most important features for any class (Cash Ratio ranked highest with FI 0.001), nor did their missing indicator. However, adding financial ratios caused a shift in the ranking of most important features. Two features that were used for the

computation of financial ratios ('Liabilities' and 'Securities Current') are listed in the most important feature for SMOTE+FR, but absent in the listing for SMOTE.

Most essential features were listed in Table 5.4 and Table 5.5. Five features ('Interest Received Classified As Investing Activities', 'Assets Noncurrent Other', 'Inventories', 'Called Up Share Capital' and 'Cash Flow From Operations') were among the ten most important features for all classes for SMOTE, and four ('Interest Paid Classified As Financing Activities', 'Interest Paid Classified As Operating Activities', 'Assets Noncurrent Other' and 'Called Up Share Capital') for SMOTE+FR. In both lists, nine features (M_x_{34}, x_{105}, x_{129}, M_x_{65}, x_{52}, M_x_{82}, x_{144}, x_{47}, x_{28}) were present in the top 10 for a majority of the classes. Remarkably, only two of the twenty most frequent attributes are among the aforementioned lists ('Assets Noncurrent Other' and 'Inventories'). Among the 23 unique features in the top ten of SMOTE, eight missing indicators are listed. This ratio is seven out of 26 for SMOTE+FR. According to three missing-indicators, the presence of the attributes 'Investment Properties', 'Interest Received Classified As Operating Activities', and 'Intangible Assets' appear to hold considerable information for classification for a majority of the classes. For SMOTE and SMOTE+FR, the top 26 and top 28 features are optimal for classification, respectively. At those point, the cumulative FI is 0.879 for SMOTE and 0.868 for SMOTE+FR. A cumulative FI of 0.99 is reached by using the top 56 features for SMOTE or the top 64 features for SMOTE+FR. These insights can help in picking the amount of attributes to use when the model deployed.

There are several points to note about the obtained results. First, all insights are obtained from one classification algorithm without tuning hyperparameters. The latter also applies to the three class imbalance approaches from the imbalance-learn package. Third, only five financial ratios with a low frequency could be computed from the data set and added to the 155 other features. At last, all results are obtained from one data set, and as stated in Section 2.2 this data set is not a perfect representation of all Dutch companies. The combination of considerable data sets for more training data could lead to better classification, the ability to specify more classes, a more thorough analysis of the impact of financial ratios, and more generally applicable insights.

Chapter 6

Conclusions and Future Work

In this study we have performed section prediction by applying data mining techniques on financial statements. This enables applications such as data completion for sector analysis, and the detection of mislabeled company statements. A baseline was established by determining the performance of random forest classifiers, in terms of a ROC curve for each class and mean of their AUCs. An AUC of 0.78 ± 0.07 demonstrates the suitability of such an approach. Divergent frequencies of classes indicated the presence of the ubiquitous problem of class imbalance, and impacted the performances of minority classes. Besides, we observed that other factors impact the performances, which should be further investigated. Five approaches to deal with class imbalance were applied, among one custom approach. SMOTE was the only approach with a slight improvements of the results (AUC 0.79 ± 0.05), and therefore used to answer our sub questions:

1. *Can financial ratios be used to improve sector prediction using financial statements?*
   The application of five financial ratios to our data set did not improve sector prediction for any of our approaches, nor did it decrease performance.

2. *What are the most important attributes of financial statements useful for sector prediction?*
   We concluded that a small subset of all features from both balance sheets and income statements are the top features for the majority of the classes, without strong connection regarding the frequency of the feature. Financial ratios did not appear in this list. Missing-indicators appeared as important feature for a majority of the classes and hereby indicated that the presence of a attribute on a financial statement is occasionally more important than the value itself.

3. *What is the optimal number of attributes of financial statements for sector prediction?*
   For a data set without financial ratios, we concluded that the use of the 26 most important features is optimal. For a data set with financial ratios, this optimum is 28.

In summary, we conclude that data mining techniques by means of supervised learning on financial statements can be used for business sector prediction.

We make five recommendations for future work. The first two imply a repetition of the experiments on a later moment in time: the combination of several previously downloaded versions of the data set will yield a bigger data set that would result in a better performance. Besides, an increased data set can be used for time series analysis of the development of all companies or specific sectors. Another future direction could be the comparison between different classification algorithms for sector prediction. Due to global and European standards for sector categorization, another contribution could be made by combining international data sets. At last, future work could deepen our knowledge of sector categorization by using other data sources, such as text mining on written reports.

Bibliography

[1] XBRL-International, "An introduction to XBRL." https://www.xbrl.org/the-standard/what/an-introduction-to-xbrl/, 2014. Accessed: 2019-06-09.

[2] A. Sharma and P. Panigrahi, "A review of financial accounting fraud detection based on data mining techniques," *International Journal of Computer Applications*, vol. 39, p. 3747, 2012.

[3] M. Zieba, S. Tomczak, and J. Tomzak, "Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction," *Expert Systems with Applications*, vol. 58, pp. 93–101, 2016.

[4] KvK, "Home." https://www.kvk.nl/english/, 2019. Accessed: 2019-05-05.

[5] KvK, "Jaarrekeningen open data set." https://www.kvk.nl/producten-bestellen/koppeling-handelsregister/kvk-jaarrekeningen-open-data-set/, 2019. Accessed: 2019-01-21.

[6] E. Evink and L. van der Tas, "Digitalisering van de jaarrekening: het gebruik van XBRL in gedeponeerde jaarrekeningen van middelgrote ondernemingen," *Maandblad voor Accountancy en Bedrijfseconomie*, vol. 92, p. 361373, 2018.

[7] KvK, "Handleiding jaarrekeningen maart 2019." https://www.kvk.nl/download/Handleiding_jaarrekening_tcm109-464530.pdf, 2019. Accessed: 2019-05-05.

[8] CBS, "Standard industrial classifications (dutch sbi 2008, nace and isic)." https://www.cbs.nl/en-gb/our-services/methods/classifications/activiteiten/standard-industrial-classifications--dutch-sbi-2008-nace-and-isic--, 2018. Accessed: 2019-06-10.

[9] CBS, "Relaties tussen (inter)nationale standaardclassificaties." https://www.cbs.nl/nl-nl/onze-diensten/methoden/classificaties/algemeen/relaties-tussen--inter--nationale-standaardclassificaties, 2012. Accessed: 2019-06-10.

[10] P. Kruiskamp, *Standaard Bedrijfs Indeling 2008*. Centraal Bureau voor de Statistiek, www.cbs.nl, January 2019. Accessed: 2019-05-05.

[11] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Pratical Machine Learning Tools and Techniques*. Morgan Kaufmann, 4th ed., 2017.

[12] S. Bagui, D. Mink, and P. Cash, "Data mining techniques to study voting patterns in the us," *Data Science Journal*, vol. 6, pp. 46–63, 2007.

[13] I. Kavakiotisab, O. Tsavec, A. Salifoglouc, N. Maglaverasbd, I. Vlahavasa, and I. Chouvardabd, "Machine learning and data mining methods in diabetes research," *Computational and Structural Biotechnology Journal*, vol. 15, pp. 104–116, 2017.

[14] D. Enke and S. Thawornwong, "The use of data mining and neural networks for forecasting stock market returns," *Expert Systems with Applications*, vol. 29, no. 4, pp. 927–940, 2005.

[15] J. Peral, A. Mate, and M. Marco, "Application of data mining techniques to identify relevant key performance indicators," *Computer Standards Interfaces*, vol. 54, no. 2, pp. 76–85, 2017.

[16] J. Huang and C. X. Ling, "Using auc and accuracy in evaluating learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 3, pp. 299 – 310, 2005.

[17] T. Fawcett, "An introduction to roc analysis," *Pattern Recogniton Letters*, vol. 27, pp. 861–874, 2006.

[18] P. Ravisankar, V. Ravi, G. Raghava Rao, and I. Bose, "Detection of financial statement fraude and feature selection using data mining techniques," *Decision Support Systems*, vol. 50, pp. 491–500, 2011.

[19] E. Kirkos, C. Spathis, and Y. Manolopoulos, "Data mining techniques for the detection of fraudulent financial statements," *Expert Systems with Applications*, vol. 31, pp. 995–1003, 2007.

[20] R. Kanapickiene and Z. Gundiene, "The model of fraud detection in financial statements by means of financial ratios," *Procedia - Social and Behavioral Science*, vol. 213, pp. 321–327, 2015.

[21] H. Dalnial, A. Kamaluddin, Z. Mohd Sanusi, and K. Syafiza Khairuddin, "Accountability in financial reporting: Detecting fraudulent firms," *Procedia - Social and Behavioral Science*, vol. 145, pp. 61–69, 2014.

[22] Y. J. Kim, B. Baik, and S. Cho, "Detecting financial misstatements with fraud intention using multi-class cost-sensitive learning," *Expert Systems with Applications*, vol. 62, pp. 32–43, 2016.

[23] S. Kotsiantis, E. Koumanakos, D. Tzelepsis, and V. Tampakas, "Forecasting fraudulent financial statements using data mining," *Economics and Management Engineering*, vol. 1(12), pp. 844–849, 2007.

[24] T. K. Sung, N. Chang, and G. Lee, "Dynamics of modeling in data mining: Interpretive approach to bankruptcy prediction," *Journal of Management Information Systems*, vol. 16, pp. 63–86, 1999.

[25] K. Ishibashi, T. Iswaki, S. Otomasa, and K. Yada, "Model selection for financial statement analysis: Variable selection with data mining technique," *Procedia Computer Science*, vol. 96, pp. 1681–1690, 2016.

[26] D. L. Olson, D. Delen, and Y. Meng, "Comparative analysis of data mining methods for bankruptcy prediction," *Decision Support Systems*, vol. 52, p. 464473, 2012.

[27] G. Dattilo, S. Greco, E. Masciari, and L. Pontieri, "A hybrid technique for data mining on balance-sheet data," in *Proceedings of the Second International Conference on Data Warehousing and Knowledge Discovery*, DaWaK 2000, (London, UK), pp. 419–424, Springer-Verlag, 2000.

[28] A. B. Cook, "Missing values." https://www.kaggle.com/alexisbcook/missing-values, 2019. Accessed: 2019-07-03.

[29] N. V. Charla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[30] T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "Comparing boosting and bagging techniques with noisy and imbalanced data," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 41, pp. 552 – 568, 2011.

[31] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, p. 832844, 1998.

[32] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[33] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[34] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning," *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017.

[35] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics*, vol. 1, pp. 80 – 83, 1945.

[36] K. J. Archer and R. V. Kimes, "Empirical characterization of random forestvariable importance measures," *Computational Statistics Data Analysis*, vol. 52, pp. 2249–260, 2008.

Appendix

Appendix A: All features

The table on the next page displays all features in the data set and their relative frequency.

x_i	Feature name	Frequency
x_1	FinancialYear	1.0000000
x_2	DocumentAdoptionDate	0.9999888
x_3	BalanceSheetBeforeAfterAppropriationResults	0.9978865
x_4	Assets	0.9946507
x_5	Equity	0.9929511
x_6	EquityAndLiabilities	0.9929201
x_7	AssetsCurrent	0.9648240
x_8	AssetsNoncurrent	0.8293851
x_9	ShareCapital	0.6433109
x_{10}	LiabilitiesCurrent	0.6215645
x_{11}	Receivables	0.5951621
x_{12}	CashAndCashEquivalents	0.5794669
x_{13}	ReservesOther	0.5405753
x_{14}	SbiBusinessCode	0.3908587
x_{15}	Provisions	0.3872703
x_{16}	FinancialAssets	0.3692738
x_{17}	PropertyPlantEquipment	0.3689271
x_{18}	AssetsCurrentOther	0.3344431
x_{19}	Liabilities	0.3302379
x_{20}	AssetsNoncurrentOther	0.2856880
x_{21}	LiabilitiesNoncurrent	0.2139858
x_{22}	Inventories	0.1319039
x_{23}	SharePremium	0.1251740
x_{24}	RetainedEarnings	0.1165948
x_{25}	IntangibleAssets	0.0789726
x_{26}	ResultForTheYear	0.0586885
x_{27}	LegalReserves	0.0454692
x_{28}	SecuritiesCurrent	0.0305733
x_{29}	ConstructionContractsAssets	0.0231989
x_{30}	RevaluationReserve	0.0220555
x_{31}	LegalStatutoryReserves	0.0178041
x_{32}	CostsIncorporationShareIssue	0.0175254
x_{33}	CalledUpShareCapital	0.0156564
x_{34}	InvestmentProperties	0.0118235
x_{35}	Securities	0.0101727
x_{36}	StatutoryReserves	0.0077481
x_{37}	ResultAfterTax	0.0032490
x_{38}	ResultBeforeTax	0.0032384
x_{39}	GrossMargin	0.0032292
x_{40}	FinancialIncomeExpenses	0.0032220
x_{41}	OperatingResult	0.0032180
x_{42}	SumOfExpenses	0.0032022
x_{43}	OperatingExpensesOther	0.0031462
x_{44}	IncomeTaxExpense	0.0031416
x_{45}	EmployeeBenefitsExpenses	0.0030770
x_{46}	DepreciationPropertyPlantEquipmentAmortisationIntangibleAssets	0.0029583
x_{47}	OperatingResultCashFlow	0.0025023
x_{48}	ChangesWorkingCapital	0.0024865
x_{49}	IncreaseDecreaseCashAndCashEquivalents	0.0024707
x_{50}	CashFlowOperatingActivities	0.0024456
x_{51}	IncreaseDecreaseInOtherPayables	0.0024318
x_{52}	CashFlowFromOperations	0.0024311
x_{53}	AdjustmentsDepreciationAndAmortisationExpense	0.0023817
x_{54}	CashFlowInvestingActivities	0.0023751
x_{55}	AdjustmentsReconcileOperatingResult	0.0023738
x_{56}	IncreaseDecreaseInOtherReceivables	0.0023731
x_{57}	CashAndCashEquivalentsCashFlow	0.0023402
x_{58}	PurchasePropertyPlantEquipment	0.0023086
x_{59}	NetCashFlows	0.0022545
x_{60}	CashFlowFinancingActivities	0.0022334
x_{61}	InterestPaidClassifiedAsOperatingActivities	0.0022196
x_{62}	IncomeTaxPaidClassifiedAsOperatingActivities	0.0020014
x_{63}	IncreaseDecreaseInInventories	0.0019870
x_{64}	ShareInResultsParticipatingInterests	0.0019606
x_{65}	InterestReceivedClassifiedAsOperatingActivities	0.0017972
x_{66}	EquityGroup	0.0017543
x_{67}	IncreaseDecreaseProvisions	0.0016772
x_{68}	NetResultAfterTax	0.0016047
x_{69}	ProceedsSalesPropertyPlantAndEquipment	0.0015382
x_{70}	ConstructionContractsLiabilities	0.0011915
x_{71}	ReinvestmentReserve	0.0011790
x_{72}	DividendsPaidClassifiedAsFinancingActivities	0.0010913
x_{73}	RepaymentsBorrowings	0.0010887
x_{74}	IncreaseDecreasePayablesCreditInstitutions	0.0009661
x_{75}	IncreaseDecreaseInTradeAccountsReceivable	0.0009497
x_{76}	OtherIncomeExpensesAfterTax	0.0008956
x_{77}	ProceedsFromBorrowings	0.0008304
x_{78}	ProceedsSalesOtherFinancialAssets	0.0008205
x_{79}	IncreaseDecreaseInTradeAccountsPayable	0.0007249
x_{80}	CashFlowsFinancingActivitiesOther	0.0006689
x_{81}	PurchaseOtherFinancialAssets	0.0006498
x_{82}	PurchaseIntangibleAssets	0.0006076
x_{83}	IncreaseDecreaseInConstructionContracts	0.0005088
x_{84}	NoncontrollingInterest	0.0004923
x_{85}	ResultAttributableNoncontrollingInterest	0.0004712
x_{86}	AdjustmentsReconcileOperatingResultOther	0.0003934
x_{87}	IncomeReceivablesNoncurrentSecurities	0.0003256
x_{88}	CashFlowsOperatingActivitiesOther	0.0003064
x_{89}	OutcomeAssetsCurrentLessLiabilitiesCurrent	0.0002873
x_{90}	IncomeTaxReceivedClassifiedAsOperatingActivities	0.0002511
x_{91}	CashFlowsInvestingActivitiesOther	0.0002432
x_{92}	ChangesValueFinancialAssetsSecurities	0.0001779
x_{93}	SellingExpenses	0.0001766
x_{94}	ChangesInValueIntangibleAssetsPropertyPlantEquipment	0.0001760
x_{95}	ProceedsIssueShares	0.0001713
x_{96}	EffectExchangeRateChangesCashAndCashEquivalents	0.0001568
x_{97}	DividendsReceivedClassifiedAsOperatingActivities	0.0001529
x_{98}	GeneralAdministrativeExpenses	0.0001522
x_{99}	AdjustmentsImpairmentLossReversalImpairmentLossRecognisedInProfitLoss	0.0001239
x_{100}	PurchaseGroupCompanies	0.0001239
x_{101}	OutcomeAssetsLessLiabilitiesCurrent	0.0001127
x_{102}	IncreaseDecreaseInSecurities	0.0001081
x_{103}	ImpairmentCurrentAssets	0.0000942
x_{104}	PaymentsAcquireRedeemEntitysShares	0.0000942
x_{105}	ProceedsSalesIntangibleAssets	0.0000929
x_{106}	PurchaseNonConsolidatedEntities	0.0000844
x_{107}	InterestPaidClassifiedAsFinancingActivities	0.0000725
x_{108}	ProceedsSalesNonconsolidatedEntities	0.0000718
x_{109}	ProceedsSalesGroupCompanies	0.0000606
x_{110}	RevaluationReserveRelease	0.0000540
x_{111}	LineItemsOtherIncomeStatementNoImpactReceiptsPayments	0.0000468
x_{112}	DividendsPaidClassifiedAsOperatingActivities	0.0000455
x_{113}	PurchaseInvestmentProperties	0.0000330
x_{114}	LineItemsOtherIncomeStatementReceiptsPaymentsNotConsideredOperatingActivities	0.0000316
x_{115}	ChangesValueInvestmentProperties	0.0000310
x_{116}	DividendsReceivedClassifiedAsFinancingActivities	0.0000264
x_{117}	ProceedsSalesInvestmentProperties	0.0000250
x_{118}	FiscalReservesOther	0.0000224
x_{119}	ProceedsCashObtainedAcquisition	0.0000165
x_{120}	InterestReceivedClassifiedAsFinancingActivities	0.0000158
x_{121}	UnrealisedChangesInValueOfInvestments	0.0000145
x_{122}	RealisedChangesInValueOfInvestments	0.0000132
x_{123}	ChangeValueAgriculturalStocks	0.0000125
x_{124}	EarningsPerShareBasic	0.0000125
x_{125}	IncomeTaxPaidClassifiedAsFinancingActivities	0.0000125
x_{126}	ReceiptsCustomers	0.0000099
x_{127}	NetRevenue	0.0000092
x_{128}	PaymentsSuppliers	0.0000092
x_{129}	PaymentsReclaimingValueAddedTax	0.0000086
x_{130}	ExpensesOther	0.0000079
x_{131}	IncomeTaxReceivedClassifiedAsFinancingActivities	0.0000079
x_{132}	InterestPaidClassifiedAsInvestingActivities	0.0000079
x_{133}	DepreciationAmortisationAndDecreaseInValueAssets	0.0000072
x_{134}	IncomeOther	0.0000072
x_{135}	InterestReceivedClassifiedAsInvestingActivities	0.0000072
x_{136}	IncomeTaxPaidClassifiedAsInvestingActivities	0.0000066
x_{137}	PaymentsEmployees	0.0000066
x_{138}	IncomeTaxReceivedClassifiedAsInvestingActivities	0.0000059
x_{139}	WagesSalaries	0.0000059
x_{140}	EarningsPerShareDiluted	0.0000053
x_{141}	PaymentsPurchaseGoodsServices	0.0000053
x_{142}	ReceiptsPaymentValueAddedTax	0.0000053
x_{143}	ReceiptsRoyaltiesCommissionAndSuch	0.0000053
x_{144}	PaymentsInstallmentsOperationalLeasing	0.0000046
x_{145}	PaymentsProductionProcess	0.0000046
x_{146}	CostsRawMaterialsConsumables	0.0000026
x_{147}	IncomeTaxExpenseOrdinaryActivities	0.0000026
x_{148}	ResultBeforeTaxOrdinaryActivities	0.0000026
x_{149}	FinancialExpenses	0.0000020
x_{150}	CostOfSales	0.0000013
x_{151}	Expenses	0.0000013
x_{152}	FinancialIncome	0.0000013
x_{153}	GrossOperatingResult	0.0000013
x_{154}	Income	0.0000013
x_{155}	NetOperatingResult	0.0000013
x_{156}	OperatingExpenses	0.0000013
x_{157}	OperatingIncomeOther	0.0000013
x_{158}	WagesSalariesSocialSecurityCharges	0.0000007