Informatica & Economie

Data quality analysis

for the Dutch Algorithm register

Martin Janssen

s2937387

Supervisors:
Prof. dr. ir. Joost Visser

Prof. dr. ing. Bram Klievink

BACHELOR THESIS

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

August 2023

Abstract

The Dutch Government has taken the initiative for building and launching a Dutch Algorithm register. In this register all government departments can register their algorithms using personal information on or affecting citizens. Currently, the submission of algorithms by the government is not required by law, however, in the future this will become a requirement for every department. At the time of this study, the register contains about 160 algorithms, ranging from detection of wrongful parking to calculating a citizen's social aid.

The main objective of this research is to find out what the current data quality of the register is, how the data quality of the register can be improved and how much data quality dashboards contribute to that.

We will investigate this using the Design Science Research Process (DSRP) model [1]. This process is very useful when executing Design Science [2] in Information Systems. This is a good fit for our research regarding the algorithm register. Using this model, we will design software that aims to give clear insight into the data quality of the register. The software will mainly contain a data quality dashboard that visualizes a few different metrics of the quality in the register.,We will follow the steps of the Design Science research process and will then validate and quantify the success of our solution using the Technology Acceptance Model [3]. This gives us clear results on how the designed and implemented software contributes to the data quality of the register.

After analyzing the current state of the data quality of the data itself in the register, the data structure, setup of the register and building an artifact with visualizations about the data quality, we came to some interesting results. The data contains a lot of empty values, namely 54%. Next to this, we analyzed the readability of the data. 33% of the entries in the register is not easily accessible for the average citizen, when it comes to language level. Next to completeness and readability, the correctness of the data is quite low. On average only 15% of the typed fields is correctly filled out. If we look at the data structure, there are some improvements to be made, for example the use of enumerators in some text fields. After building and demonstrating the artifact we built during this research, the evaluation of the artifact came out very positive. The perceived usefulness scored high, the perceived ease of use got some improvement feedback, but the artifact is scored as helpful and will contribute to a better data quality in the near future of the register.

After this research we can conclude that the visualizations we built for the data quality of the dutch algorithm register are really helpful and will contribute to a better understanding of the data quality and will lead to concrete improvements of it. We can also conclude that in the current data there are a lot of missing, or difficult values, which lowers the data quality. For the register to be used as intended, the quality of the data needs to be improved as soon as possible.

Contents

1 Introduction

The Dutch Algorithm Register has been called to life during a rising trend of striving for more transparency and accountability regarding the use of AI, as spearheaded by the European Commission. The citizens were discriminated based on their race, cultural background or religion [4] and had to pay large amount of incorrect fines. This caused children to be taken away from their parents [5] and suicides because of the enormous stress. After this huge societal problem came to light, the government tried to do everything in their power to restore faith in government authorities. The parents and children are compensated and the whole allowance structure as we know it will be completely restructured. Next to this, the government wants to provide more openness on the digital side, hence the algorithm register.

The register currently contains about 160 algorithms [6], coming from government entities like municipalities and province houses. Algorithms in the register range from simple rule based algorithms to complex self learning algorithms that are sometimes checked by humans, but in many other cases operate on their own. These algorithms sometimes make decisions on whether a citizens gets any allowances or approval in permit cases.Currently, the registration of algorithms by government entities is on a voluntary basis. In the near future, however, every government entity is obligated to register all of their algorithms because of an accepted resolution by parliament that states that every government institution should use the register [7]. This will probably also increase the data quality of the register, since there will be legal consequences if the registrations are incorrect or have missing values.

Since the register is still in an experimental phase there are still some things to be improved. When looking at the data structure, we see that some fields that should have had fixed values are just freely filled by government entities. This causes noisy data, which can be prevented by defining enumerators. Next to this, we saw that a lot of fields remained empty or that fields had a very high level of language used in them. Currently, there is not yet a system or dashboard that gives the team working on the register insight into these flaws in the data quality of the register. Having these insights can make their job improving the register and its data a lot easier and will increase the accessibility of the register. Therefore we will introduce some recommendations on how to improve these basic structure and data mistakes, but moreover, we will design and build a dashboard that gives a lot of insight in the current problems with the data, which will help the team working on the register to improve it adequately.

Our main research question for this thesis is therefore: What can be improved on the current algorithm register when it comes to data quality and data structure, and how can visualizations such as dashboards and KPIs contribute to this improvement?

1.1 Thesis overview

In this thesis we will take you through some related work on this topic. There is not that much existing research on actual algorithm registers, since those do not really exist everywhere around the world yet, but there's quite some research on topics surrounding the register and algorithms. After introducing you to the world of algorithms and all of their risks and opportunities, we will continue by taking you through the current data quality of the register by using the data quality metrics, give you some exact numbers on the current quality and take you through the whole design and demonstration process of the visualizations for the team working on the register. We will conclude our thesis with the results of the evaluation of the visualizations as described in the Technology Acceptance Model.

2 Related Work

In this chapter we will go over some related research when it comes to the Dutch algorithm register. Since there is no direct research on the topic, we will explore topics surrounding the algorithm register, such as algorithm governance, accountability and audits. Next to this we will also look into the complexity of the use of algorithms and AI by governments, since this is closely related to the register.

2.1 Algorithm governance

Because of ever-increasing compute power and data sets, algorithms are able to perform and finish task with a complexity, that's just impossible in any way for a human [8]. With the increase of the sophistication of algorithms, they become more and more autonomous. The algorithms output is very hard to explain by scientists and this poses an important question. How can we guarantee that the algorithm's output is not biased or discriminating in any form? Since this is quite hard, we should look into algorithm governance. How can algorithms be governed in a responsible way?

Since it's hard to really put a finger on the working of an algorithm, there's the possibility of opening up the source code of all the algorithms. However, this poses some business threats, since competitors can also take a look at the algorithms. This means that this option is not really applicable in a real-world scenario.

Next to opening up the source code, we could also look into the data sets that the algorithm runs on [8]. If we make sure that the data the algorithm is run on, is clean and does not contain any bias or prejudice or discriminatory factors, the chance that the algorithm will create that bias itself, is made a lot smaller. It can never be ruled out, but the possibilities of bias occurring in the outcome of an algorithm is largely reduced.

The implementation of algorithm governance can exist on three different levels: private companies, industry wide, governmental oversight [8]. Private companies should make sure that their algorithms maintain a level of transparency that makes sure that their customers aren't harmed by the software. Industry standards could be defined, so everybody is working with the same standards and approaches algorithms in the same way. Governments could put oversight organizations into place that oversee the rules surrounding algorithms.

2.2 Algorithm accountability

Closely related to the subject of algorithm governance lies algorithm accountability. How are they related? Well, whenever clear rules and conditions on algorithms are set and kept, it becomes possible to hold certain entities and organizations accountable for their algorithms and their working. It's actually connected in a beautiful way: when having to be transparent about your algorithms and work, one starts to behave better [9]. So how can we define accountability? Accountability as defined by Bovens [10]:

> a relationship between an actor and a forum, in which the actor has an obligation to explain and to justify his or her conduct, the forum can pose questions and pass judgment, and the actor may face consequences.

So, the actor can be any entity, such as an individual, an organization or a government department. This actor has to explain itself to a 'forum' which can be something like a judge, the public, a government oversight organization or an independent authority.

But who should be kept accountable? It's quite hard to lay a finger on a specific actor, since not all companies create their own algorithms for example. Who is guilty? The developer of the algorithm or the company that uses the algorithm? Bovens defines four different accountability relations, based on the level of the actor [9]:*individual* accountability, *hierarchical* accountability, *collective* accountability and *corporate* accountability. Individual accountability means that the individual's conduct is held to their be their own responsibility. In other words, an individual will not be shielded from any investigation by their superiors or organization. Hierarchical accountability means that the person leading the organization, department or team is going to be held accountable. When one member of an organization can be held accountable for the actions of the whole organization they are related to, this is called collective accountability. Lastly, corporate accountability means that a non-human entity is held responsible and accountable [9].

So, what about the other party in algorithm accountability? The forum is the party to whom the account is directed. Kemper and Kolkman [11] argue that in order to be able to give account, the forum must be able to understand the matter and engage with the material in a critical way. This is important for the algorithm register, since this implies that the account that the organizations give in the register, should be understandable and accessible for citizens. Just as for the actors we can also define different accountability relations [10]: *political* accountability, *legal* accountability, *administrative* accountability, *professional* accountability and *social* accountability. These accountability relations are quite unambiguous, but we will shortly explain them. Political accountability means that an actor can be held politically accountable, for example when it comes to ministerial responsibility. Legal accountability means that some actions can be held accountable by, and found in the law and thus by a judge. Administrative accountability refers more to the semi-legal forums such as nationwide advisory fora. Professional accountability deals with the accountability relation between a professional and its peer group or supervisor. Lastly, social accountability can be seen as the accountability relation between a public agency and citizens for example. Fora that would take part in this relationship could be NGO's, interest groups or individuals. Next, when it comes to algorithm accountability explainability of the algorithm is important. there are several ways of making the algorithms explainable [12]. One can explain the model or the outcome of the algorithm. Next to this we can try to inspect the black box, which is the algorithm or the implementation of it. Last but not least, maybe the ideal situation would be to create a 'transparent' box, which would mean that the algorithm is created transparently and can be analyzed by the public. Lastly when it comes to accountability, it's important to notice that it is not necessarily accomplished in one single point of time or person. Rather, it is much more of a process where it is spread over multiple people and across multiple moments of time [13].

2.3 Algorithm audit

In order to hold actors accountable for their algorithms or the outcome of their algorithms, we should be able to audit algorithms properly. If we can do this is in a concise way and if we have clear sets of rules as mentioned in section 2.1, we can then hold the actors accountable in a fair way. So, how can we audit algorithms? The problem with algorithm audit lies in these three sub problems: in order to audit an algorithm, one must "(1) deeply engage with social and legal facets of 'fairness' in a given context, (2) develop software that concretizes these values and (3) undergo an independent algorithm audit to ensure technical correctness and social accountability of their algorithms" [14]. To date, there are only a few companies that have transparently and accurately finished these steps.

Wilson et al. (2021) have tried to solve these problems and put the results into practice at a firm called pymetrics. Pymetrics is a company that creates software for applicant selection. One important rule that is closely related to applicant selection is the Four-Fifths rule, as stated by the UGESP [15]. The Four-Fifths rule states that if the selection rate for a certain group is less than 80 percent of that of the group with the highest selection rate, there is adverse impact on that group [16]. These groups can also be different classes, such as race, sex or religion.

2.4 Government and algorithms

There is still little research on the use of AI and algorithms by governments. However, this research should be deemed important, since governments have the obligation of protecting their citizens and their privacy, this can conflict with AI and algorithms. But since the adoption of AI and algorithms is fairly new, research on the matter is only just beginning to identify the manifold challenges faced in the adoption of AI in the public sector [17]. The article by Campion et. al (2022) aims to contribute to the research on the use of AI and algorithms in the public sector by studying the challenges of interorganizational collaborations in the adoption of AI tools and the implementation of organizational routines used to address those challenges. Their findings are interesting and could be of help for the Dutch Algorithm Register as well.

2.5 AI and algorithm quality model

Measuring the quality and fairness of AI models and algorithms is a big part of AI research. How can we determine if the 'black box' is unbiased and trustworthy? Petra Heck (LinkedIn) is a researcher in the field of AI and Software Engineering and she has done research on creating a quality model for AI and software systems. She states that: *"we should be able to verify that the software systems fulfill both the functional requirements (what does the software system need to do?) and non-functional requirements (how shall the software system be?)"* [18]. For software engineering we usually use the quality requirements as defined in the ISO25000 standard [19]. Since this standard does not really work optimally on AI or algorithms, ISO is working on a new SQuaRE version [20] of the standard. Since that standard is not finalized yet, Mrs. Heck took some gray literature surrounding this subject and extended the ISO25000 to be useful for checking AI models. The EU has made eight guidelines [21] that AI systems should abide to: (1) human agency and oversight, (2) technical robustness and safety, (3) privacy and data governance, (4) transparency, (5) diversity, (6) non-discrimination and fairness, (7) environmental and societal well-being, (8) accountability. Habibullah et al. [22] have also done research on non-functional requirements such as: retrainability, traceability and reproducibility. Added to some other literature, we now have found some requirements, that are not found in the EU guidelines, nor in the ISO25000 standard, but are really important when it comes to the quality of AI or algorithms.

These requirements are: (1) Model correctness, (2) Controllability, (3) Explainability, (4) Collaboration effectiveness, (5) Model robustness, (6) Reproducibility, (7) Privacy, (8) Human autonomy risk mitigation and (9) Unfair bias risk mitigation. The arguments behind these additions are: "(1) For AI systems we do not program the rules ourselves, but instead we train machine learning models to learn the rules from data, (2) The datasets that are used for training machine learning models are potentially huge and thus difficult to check by hand and (3) The rules that AI systems learn from example data are used in workflows to automate (parts of) human decision making." [18]. These additions should in some way be seen in the Dutch algorithm register, since the register wants to create openness about AI and algorithms, it's important to report about these metrics. The register reports on a lot of these additions, as can be seen in table 1.

Field	**Corresponding attribute**
Impact	Human Autonomy
Proportionality	Collaboration Effectiveness
IAMA description	Explainability, Fairness
DPIA description	Privacy
Appeal procedure	Controllability
Connection to citizens register	Privacy
Data sources	Reproducibility
Methods and models	Reproducibility
URL to source code	Reproducibility
Monitoring	Model Correctness
Human intervention	Controllability

Table 1: Fields of register corresponding to quality metrics

The only metric that is currently missing in the data that the register registers, is the model robustness. This is quite important, since that is all about the safety and security of the algorithms in the register.

2.6 Technology Acceptance Model

In this research we use the Technology Acceptance Model [3] to evaluate the contribution the visualizations created during this research will bring to the quality of the data in the register. The Technology Acceptance Model states that the system you build is the end-point where users will use the technology. The behavioral intention is the main factor which leads people to use and adopt the system you built. The intention to use the technology is influenced by the attitude towards your created artifact. This attitude consists of two components: Perceived usefulness and Perceived Ease of Use. When evaluating your artifact you can ask potential users questions about these two components, if their feedback is positive, the chances are big that they will adopt your technology and that it will make their work more efficient.

3 Methodology

In order to do our research on the Dutch Algorithm register, we will follow the Design Science Research Process (DSRP) model [23]. This model enables us to go through the design process of designing a dashboard that improves insight into the data quality of the registrations in the register. So how does the model work?

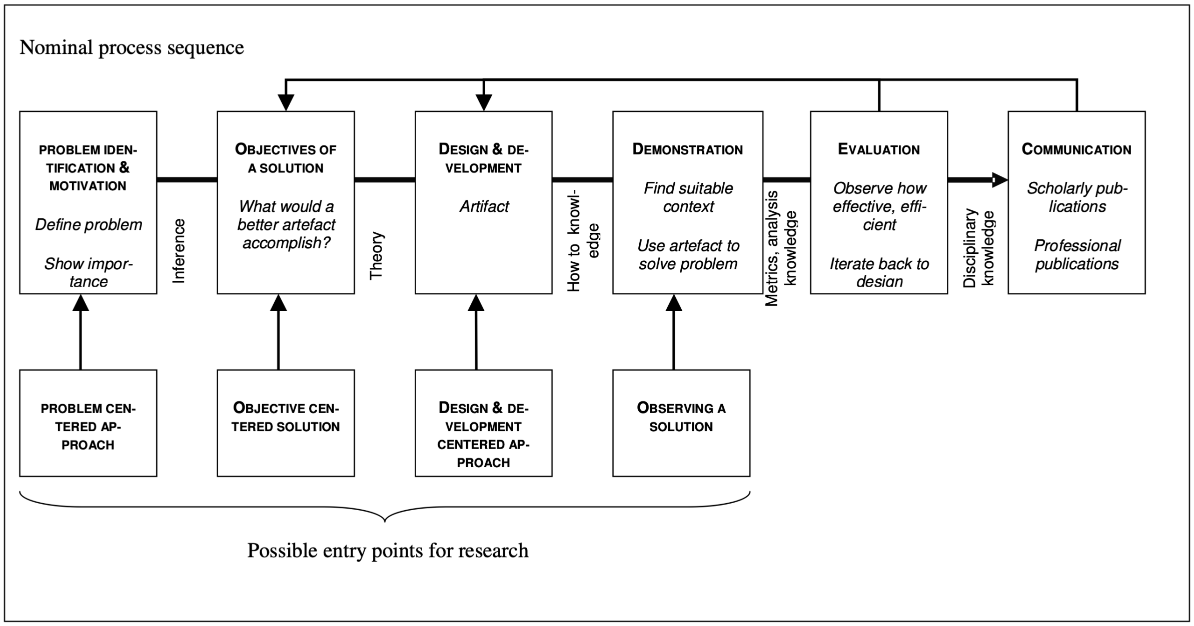

Figure 1: Design science research process model

As seen in figure 1 the DSRP model consists of six clear steps. The first one is the problem identification and motivation as to why this design and research is needed. Following up, the model discusses the objectives of the design. How will this artifact contribute to our goals? Thirdly, the model goes on to the design and development itself. After designing and developing a prototype, the process continues with a demonstration of the produced artifact. After demonstrating, the evaluation of the artifact starts, this is necessary in order to iterate back on certain decisions. Whenever we find that a feature or part of the artifact is not functioning is intended, we can circle back to our objectives or redesign our artifact. After all these steps are finished, we can finally communicate our artifact and the research surrounding it, to the world in a publication or thesis.

4 Current data quality

In this section we will analyze the current quality of the data in the register, using the 15 data quality dimensions. Next to this, we will shortly go over the exact results of our data quality measurements in the register.

Accessibility

The data is very much accessible through different methods. Firstly, the data is displayed on the website https://www.algoritmes.overheid.nl. The website enables visitors to search through all the available algorithms, as well as filtering by the responsible departments within the government. Secondly, the website allows for visitors to export all the data displayed on the website, to a CSV file. This can be useful for data analysis and reusage of the data.

Appropriate amount of information

Currently, the data contains about 160 algorithms. This does not seem that much, since the government as a whole has a lot more. This is also acknowledged on the website as they mention that more algorithms will be included, they even mentioned a legal obligation to register all algorithms in the future [24]. So, for now the data set is not that large and it would definitely need to be larger in order for it to be representative for the algorithms used within the Dutch government.

Believability

Since this is all data provided directly by the government, one can assume that the data is correct and not falsified. However, there can always be flaws in the explanation of an algorithm. Next to that, it is very hard to check and validate the assertions made in the data. If the government says that the outcome of an algorithm is always checked by humans (for example parking fines), it is very hard to check if the algorithm is actually checked in a proper way. It is possible that the outcome of the algorithm is always accepted by the human, that would not be a good check.

Completeness

This is a dimension of data quality that is completely missing in the data at this moment. There are some entries that are fully complete, but most of them miss a lot of values. Just as an indication, more than 50% of all data fields are empty. This is not up to standards and should be improved. It also makes it a lot harder to do analysis on the data, since empty values can mess up statistics.

Concise representation

When looking through the data, one thing really stands out, and that is the fact that a lot of the data fields, have lots of text. Sometimes even there is a whole answer to a question, for example about how the outcome of an algorithm is handled. These answers can be very long, and do not have the same structure for all entries. This makes it really hard to analyze the data, because it is hard to see if an answer is positive or negative using code.

Consistent representation

On the dimension of consistent representation of the data, there is a lot to improve. In the data there are only a few columns that have consistent answers. Just as an example: there is column to indicate whether or not an algorithm is active at the moment. This should be a true or false column. Instead this column contains very different text entries, ranging from: active to inactive but also 'in use', 'working' etc. This kind of data is very hard to do analysis on, since all the entries need to be converted.

Ease of manipulation

The way the system is designed right now, ensures that the data is checked in a good and thorough way and that manipulation is quite hard. An entity that wants to submit algorithms to the register, needs to contact the team behind the register in order to even start the process. They then fill out an Excel sheet with all the required fields on the algorithm. The team checks the submitted data and deploys it to the public site. This process ensures that only Government institutions get on the platform and that the data is not weird or contains errors. Once the data is submitted, it cannot be changed without contacting the team, so manipulation is almost impossible.

Currently, the team is busy with implementing an API interface for third parties, such as government institutions. This would mean that approved users can submit their algorithms on the platform, without the hassle of Excel sheets and that they can change and delete their submissions easily. However, the team will still keep control of the data, since every change, new submission or deletion will need to be approved by one of the team members. This ensures that no weird or falsified data will be posted on the platform.

Free-of-error

As we already mentioned in the paragraph above, every submission will be checked by team members. This prevent things like curse words, nasty data or data that is clearly incorrect. However, this does not really ensure that the data is necessarily correct and without any errors. For example, in the data there is at least one entry where the column that contains a `true` or `false` value about whether or not the Basis Registration for citizens is used by the algorithm, is set to `false`, while the column next to it, with explanation on the value itself, says: we use the Basis Registration for our algorithm. This is one problem we found, but there could be more.

Interpretability

The data can be interpreted quite easily. There are a lot of columns, not that much data, but every column is explained quite extensively in their metadata explanation. This ensures that people that don't really know the data or the meaning meant by the developers, can work with it and do their analysis on it.

Objectivity

It's hard to say if the data is really objective. The data is submitted by Government organizations that write about their own algorithms. A case could be made that very privacy sensitive algorithms could be dangerous to citizens. If an organization uses such an algorithm, it does not make sense to write down that the algorithm is very privacy sensitive, since that would raise a lot of questions and could give political trouble. We should believe that the government does not intentionally lie to us, but there is always a possibility that submission represent a scaled down version of the algorithm than reality.

Relevancy

The now submitted algorithms are a small amount, compared to the amount of algorithm the total government uses. However, submission of algorithms to the register is not yet legally required. This will be the case in one or two years. Looking from that perspective, it is quite neat that there are already 110 algorithms in the register. This will only be expanded until the legal requirement. This makes the register already relevant, but will make it even more relevant in the future.

Reputation

The team behind the algorithm register is working really hard on the reputation of the data and the register itself. They organize a lot of public meetings for everybody that is interested and also keep an online blog updated and filled with all noteworthy news items. There have been some news articles [25] that are quite critical about the register because of its experimental state. Inside government entities, however, the algorithm register has the reputation for being an innovative and important project.

Security

As already mentioned before, the team implemented quite some steps to ensure that data manipulation is very hard. We did not get insight in how the server is set up, but we were able to see the source code on their GitHub. [26] When looking at the code it is clear that they chose an easy, simple approach for the backend. They used FastAPI to build their backend. When looking at the code, it looks very clean and there do not seem to be any major security problems. The backend is not able to change any algorithm, so the backend does not pose any security issues. The frontend code also seems very well built and clean and is built using VueJS which is a widely supported framework.

Timeliness

While the data is probably valid for a while, and won't change that often, the team implemented a revision date in their data structure. This ensures that organizations revise their submissions regularly. The data is often not complete or it was impossible to submit it completely the first time. It's very well possible that certain descriptions and texts have been thought out after the revision period. Therefore this period is very important and should be held high.

Understandibility

The data is understandable for somebody who has some technical knowledge. However, it can be quite hard for somebody with a non technical background to understand all the data fields. As can be read in the results section, a large percentage of the data does not confirm to the Flesch Reading Ease score [27] for B1 level of language. This means that the data is not that understandable.

Value-Added

The value the register adds, is quite clear. It gives citizens the possibility to have insight in what kind of algorithms the government uses. Moreover, they have insight in the impact those algorithms have on them specifically. This enables citizens to ask specific questions to the government about the necessity of some algorithms, so that privacy concerns can be addressed adequately.

Data quality

In this research we will use three different metrics to visualize the data quality in the register. These metrics are completeness, correctness and accessibility.

Completeness

Currently the registrations in the register have 54.86% empty fields. The empty fields can be tracked back to specific fields and organizations. The top five of organizations with the most empty fields can be seen in table 2. The top five of empty fields can be seen in table 3.

Organization	**Percentage**
Gemeente Utrecht	83.33%
Sociale Verzekeringsbank	62.7%
Gemeente Amsterdam	60.71%
UWV	60.32%
Omgevingsdienst Noordzeekanaalgebied	48.02%

Table 2: Empty fields per organization

Fieldname	**Percentage**
iama_description	98.29%
provider	98.29%
publicode	95.73%
iama	88.89%
url	88.89%

Table 3: Empty percentage per field

Correctness

For the metric of correctness we have tested 4 different values: date correctness, url correctness, email correctness and algorithm type correctness. The results of the correctness test can be seen in table 4.

Metric	**Percentage**
Date correctness	10.26%
URL correctness	8.55%
Email correctness	22.22%
Algorithm type correctness	23.93%

Table 4: Correctness percentages

Accessibility

For the metric of accessibility we have tested the readability of some of the larger text fields in the register. We used the flesch reading ease score to calculate it [27]. The overall correctly readable texts in the register comes to 33.33%. The readability can also be tracked down to organizations and specific fields. The top five organizations with the most amount of readable texts can be found in table 5. The top five fields with readable texts can be seen in table 6.

Organization	**Percentage**
Sociale Verzekeringsbank	66.67%
Gemeente Utrecht	50%
Kadaster	42.86%
UWV	33.33%
Gemeente Rotterdam	30.77%

Table 5: Percentage readable texts per organization

Field name	**Percentage**
iama_description	100%
impact	57.58%
human_intervention	47.75%
performance_standard	41.51%
monitoring	32.2%

Table 6: Percentage readable texts per field

5 Dashboard Design and Evaluation

5.1 Problem identification & motivation

Currently, the Dutch algorithm register has over a hundred registered algorithms, with impacts ranging from municipal algorithms to nationwide algorithms. It can be quite hard to keep track of the quality all the different registrations, check all new registrations on quality metrics or define KPIs that visualize the quality of the whole register. That's where this research steps in. In order to keep track of the quality of the registration in the registers, the moderators of the register need clear insight in these quality metrics. The exact metrics will be mentioned later on in this thesis.

One may ask why this is so important. Well to begin with, the data in the register is about algorithms affecting Dutch citizens and their privacy. The whole purpose of the register is to be transparent about the use of these algorithms. Therefore, it's important to know if the quality of the data in the register is good and how that quality can be improved. If the quality in the register is bad, this means that citizens are still not able to really have an insight in the algorithms affecting them. Improving the quality of the algorithm register is therefore of utmost importance and building a dashboard that gives clear insights on the quality of the registrations, will contribute to this cause. Next to this, the techniques we will use to build the dashboards and check the data, can later on also be implemented on the registration side of the register. This means that in the future, registrations can be automatically checked on their quality and immediate feedback is provided to the supplying government officials.

5.2 Objectives

So, as mentioned before, the main objective is to improve the data quality in the register, by visualizing the problems that occur. How exactly do we want to do this? We want to (1) build an online dashboard that can be consulted by the team behind the register, (2) that gives insight into the data quality of the register. The exact metrics of the quality will be: (1) completeness, (2) correctness and (3) accessibility. Completeness is split up into completeness overall, per organization and per column. Correctness is split up into overall, date, email, URL and algorithm type correctness. Lastly, accessibility is mostly about the readability of the data. Readability is split up into readability overall, per organization and per column. These metrics are very clear and unambiguous to the people working on the register, in order to improve the data quality of the register.

5.3 Design and development

In this section we will discuss the process of the design of the final product, as well as the final product itself. We will introduce you to the steps taken during this process, why certain choices were made and how these choices contributed to obtaining the objective of the end product. Finally we will take an in depth look at the end product, with all the implemented techniques and frameworks.

5.3.1 Design process

The design process started when we joined the meetings of the team working on the algorithm register. This team is part of the Ministry of Internal Affairs of the Netherlands, and is in charge of the development, different policies and strategies regarding the register. After contacting the team working on the register, we were invited to join some meetings in order to get familiar with the whole process surrounding the register, and to investigate options for our research. There were some different options on the table: there was the option to improve the actual register itself. That would've meant research on user experience and design. Since that is not really where our expertise lies, we did not choose for this option. Secondly there was the option to build some dashboards and insights for the public. That would've been an extra dashboard on top of the existing register. Since the register already has some sort of dashboard, we did not choose this as our most interesting topic. Lastly, we got to the option of a data-analysis tool for the team working on the register. This dashboard and research would have direct impact on the work of the team and could easily be implemented fast, because it would not have to be approved for use by the public. That's why we chose this option as our research project.

Now that we had obtained our main research topic, we could start with gathering requirements and goals for the prototype that we would build. One thing that immediately came up, is the completeness of the data. Currently, the data fields are only filled for approximately 50 percent. That's very low, and insight into the specifics of this completeness, by displaying the specific columns that are left empty the most, or organizations that leave a lot of fields empty, can simplify solving the problem of getting more complete data in the register.

Next to completeness, it's quite important that the data in the register is correct. The register informs citizens on the governments' use of algorithms, therefore it should contain correct information. It can be quite hard to check the information in the register on correctness, since the data is supplied by governments institutions. Some types of fields can be tested though, for example: fields that should contain enumerators, email addresses, URLs or true/false. These fields seem quite unimportant, but they contain information that leads citizens to external sources to receive extra information on the registered algorithms for example. Therefore, it's important that these fields contain correct information. This can again be visualized per type of field.

Lastly, there was an important requirement from the team working on the register, to measure the readability of the data in the algorithm. The governments aims to have all the information fields on B2 level [28] of Dutch. This ensures that the average citizen, who is meant to be using the algorithm register, is able to read and understand the data. Readability can be measured in a lot of ways, for example with machine learning and language model training, but the team did not want to use any 'black box' techniques, since that would defeat the purpose of the algorithm register a bit. That's why we will use some statistical tests like the Flesch Reading Ease score [27] in order to define and test the readability of the data.

After defining the quality metrics we will measure on the dashboard, we began to think about the actual implementation, and what would be important when developing and implementing this dashboard. The current software design and structure of the algorithm register can be found on GitHub. As seen in the repository, the frontend and backend of the application are quite strictly divided, which makes it easier to add extra components, such as a dashboard, to the register. The backend is built using FastAPI and the frontend by using Vue.js. For a visualization of the current structure, refer to figure 2.

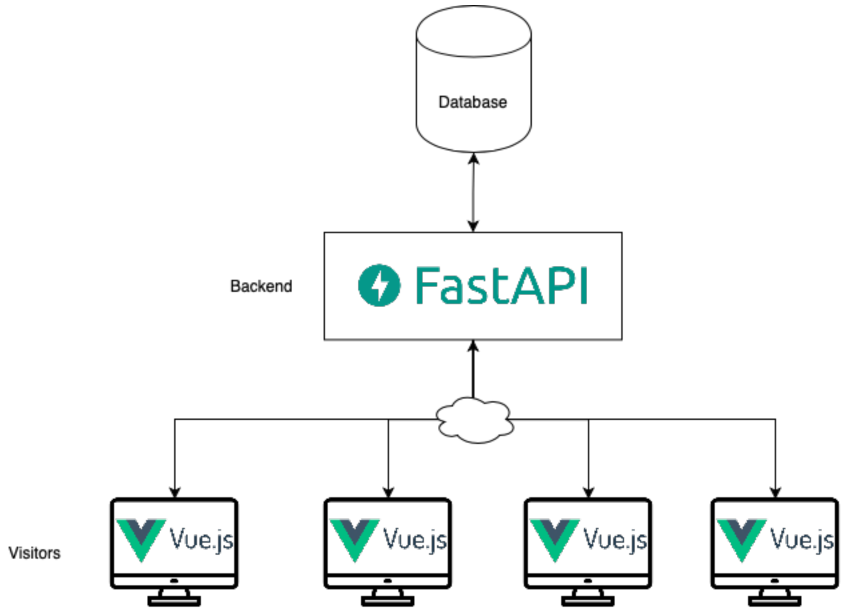

Figure 2: Software design of the current register

To ensure that our prototype could be easily adapted, if it would meet the requirements of the team, we adopted this structure as much as possible. We did choose, however, to start our own repository for this project, since the repository of the register is not updated that often, and pull requests would differ too much each time. We also adopted the frameworks used by the developers of the team. These frameworks are quite light weight and can be used to quickly launch basic applications, without having to program every single application feature yourself. For example, FastAPI already supports some basic middle ware functionalities and Vue.js has an easy structure for visualizations and dashboarding. Once the design ideas were approved by the team working on the register, we started the development.

5.3.2 Development

Now that we had a clear view on what the dashboard should visualize and why, and how we should ideally implement the dashboard, we could start the development of the prototype. Since the backend and frontend are divided, we first started out with the backend. We have a clear set of requirements of kind of data we need to have available on the backend side, so we started by defining the endpoints we would need. We would at least need endpoints for: getting all the algorithms, fields, organizations, the scores of the different algorithms and the register itself. The final defined endpoints can be seen in figure 7.

Endpoint	**Usage**
/algorithms	Get a list of all algorithms present in the register
/algorithms/{id}	Get an algorithm by ID
/organizations	Get a list of all organizations present in the register
/fields	Get a list of all fields present in the register
/algorithms/score	Get a score object with the scores on the whole register
/algorithms/{id}/score	Get a score object for one specific algorithm by ID

Table 7: All available endpoints

These endpoints need to be reachable easily by users on the front end side, and as discussed before we have used the FastAPI framework [29] to establish this. The framework ensures that REST requests get handled in a very simple way, which makes development of a stable backend a piece of cake. The design of the backend resulted in the UML class diagram as seen in figure 3. The data class is used by the main application to get the necessary data to do computations on, continued by the scorer class to calculate the results of the data quality score. It's a flexible setup that enables the scorer class to operate on different data sets or sub sets of the data and the data class to use different csv's as its data source. The data provided by the algorithm register is only csv format, so it's quite important to make the input of the backend flexible. Later on, if we were to use this artifact in the registration process of algorithms, the same software can be run by just providing another source.

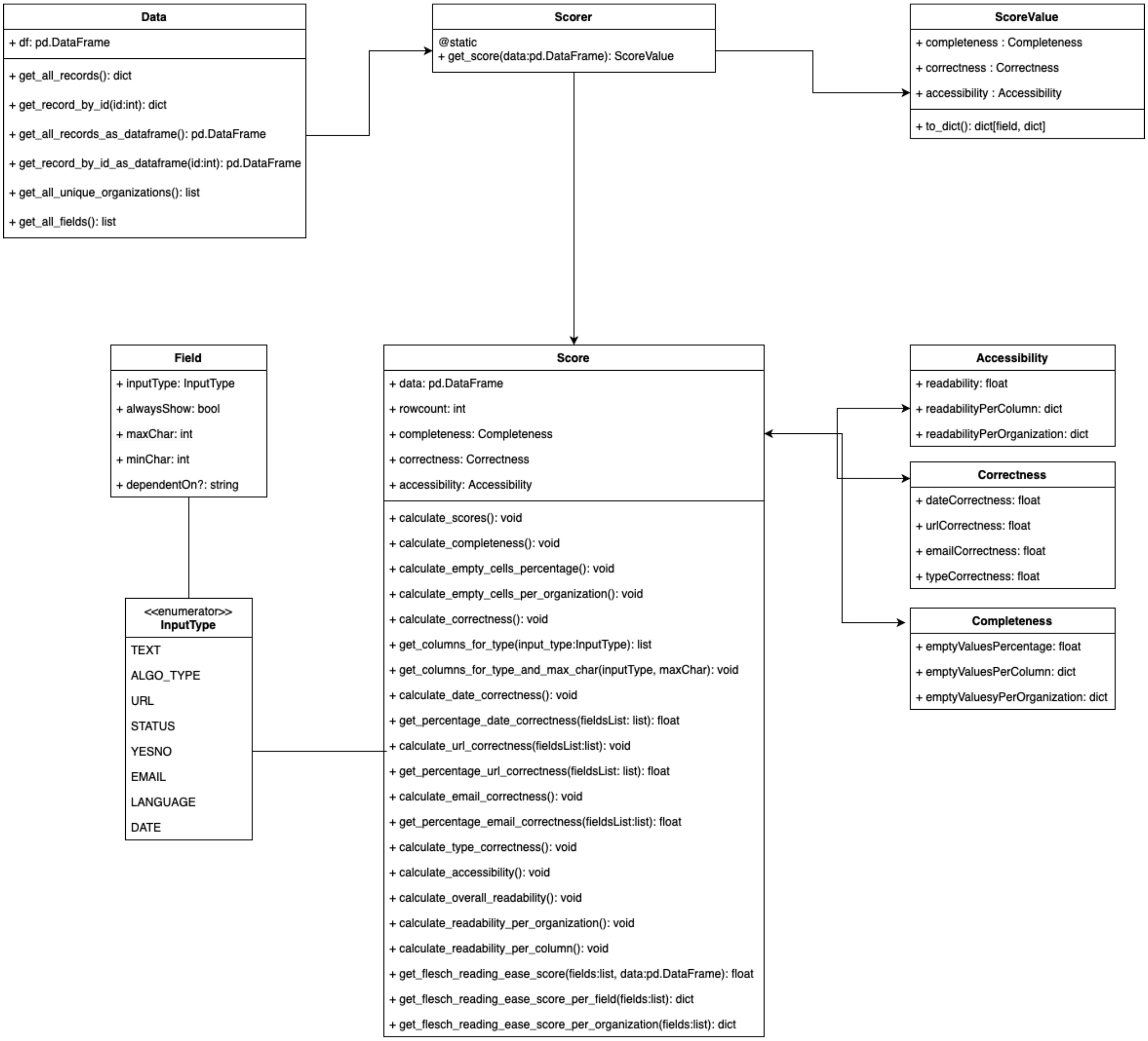

Figure 3: UML class diagram

Now that the backend is established, we can start by structuring the frontend side. The frontend is built using Vue.js [30] and therefore has way more functionality than basic HTML and CSS. The dashboard will consist of rows of visualizations and each metric shown (such as correctness or accessibility) will be its own component. Inside of these components, each visualization will be its own component again. This ensures flexibility design wise, whenever changes or redesign is desired. The visualizations itself are built using Apex Charts [31], this enables us to create complex visualizations with color schemes, multiple data sets and data hovers.

5.3.3 Demonstration

In order to demonstrate the artifact to the team working on the register, we had an online meeting with the head of the team and one of the members. During this meeting they could ask questions

about the artifact and we were able to explain the scope of the research and what was still missing from the dashboard because of time limitations.
So how does the dashboard look like? A few of the visualizations can be seen in figs. 4 to 7

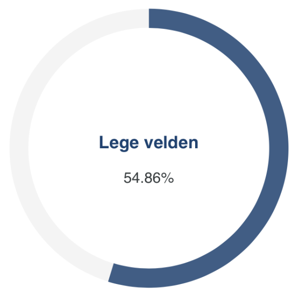

Figure 4: Empty fields percentage

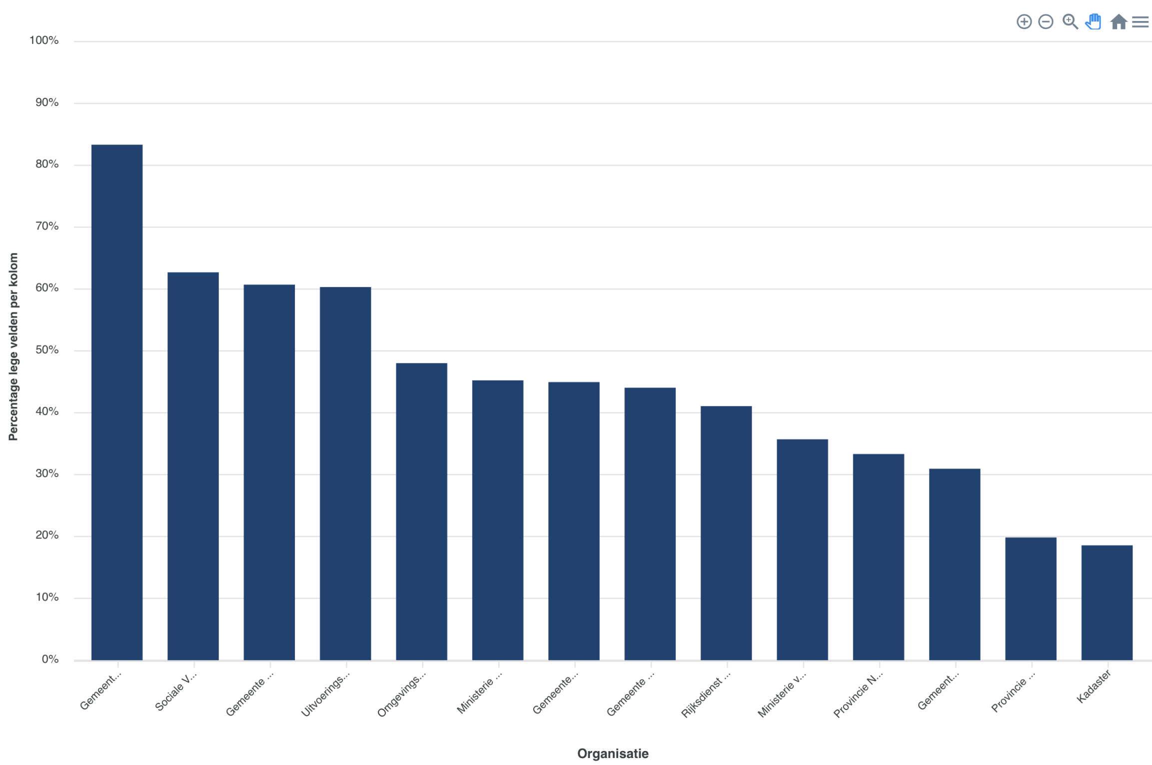

Figure 5: Empty fields percentage per organization

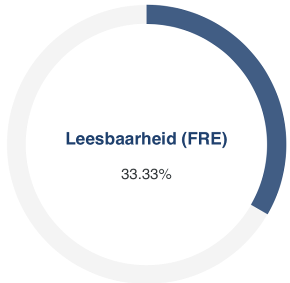

Figure 6: Readability percentage

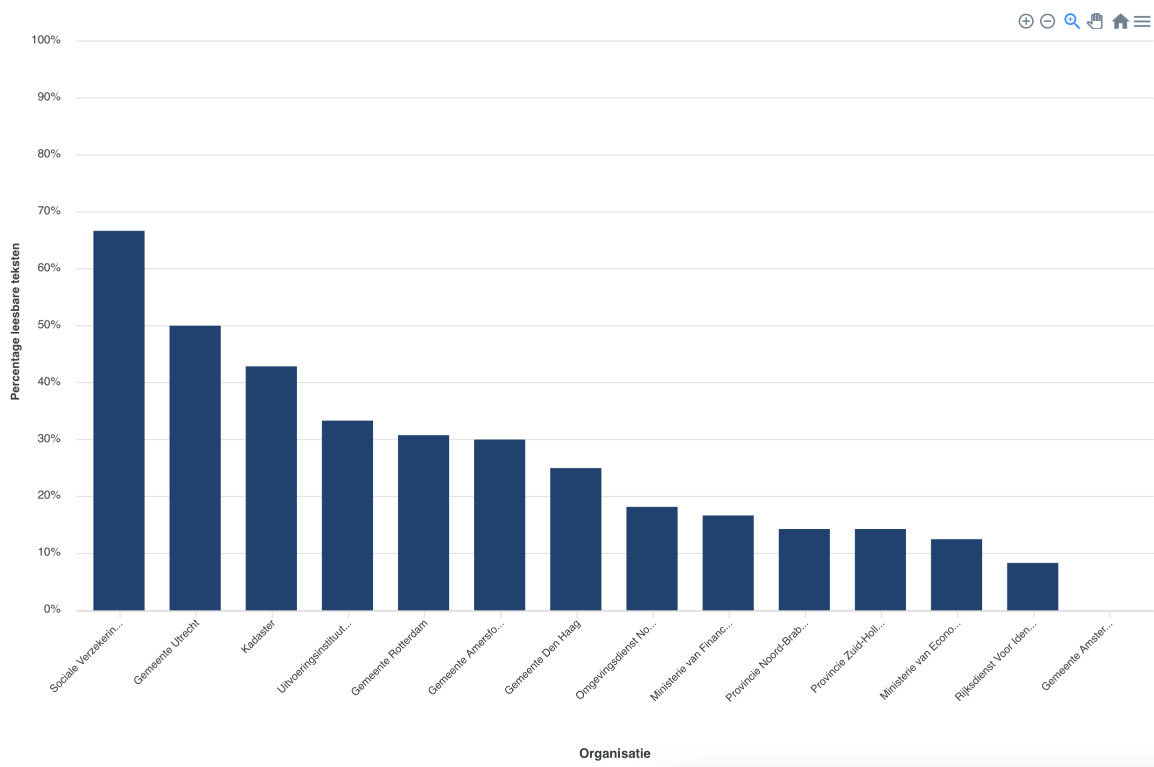

Figure 7: Readability percentage per organization

This is a selection of the visualizations on the dashboard, the rest can be seen when the code is run. The code can be found on GitHub Next to the visualizations there is also some filtering functionality that enables users to filter for certain fields or organizations.

5.3.4 Evaluation

To evaluate the success of our created artifact, we asked two team members to evaluate our artifact in accordance with the Technology Acceptance Model [3]. We presented the respondents seven statements divided in the categories Perceived Usefulness and Perceived ease of use. At the end of the questionnaire respondents could leave some additional feedback about the dashboard. The statements we presented the respondents about perceived usefulness were:

1. *The visualizations on this dashboard would help me to get better insight into the data quality of the register*

2. *The visualizations on this dashboard improve my insight on the data quality of the register when compared to the current information I have about the register*

3. *The visualizations on this dashboard make my job easier for me*

4. *I would like to start using the visualizations on this dashboard during my work*

The statements we presented the respondents about perceived ease of use were:

5. *Using and understanding the visualizations on this dashboard would be easy for me*

6. *It would be easy for me to get the information I need from the visualizations on the dashboard*

7. *I find the visualizations on the dashboard easy and flexible to work with*

The respondents could choose a number between one and five, one meaning totally disagree, five meaning totally agree. The average responses to the question can be seen in table 8.

Question	**Average result**
1	4.5 out of 5
2	4 out of 5
3	4.5 out of 5
4	5 out of 5
5	4 out of 5
6	3.5 out of 5
7	4 out of 5

Table 8: Results of the TAM questionnaire

The results show a very high perceived usefulness and a slightly lower perceived ease of use. In the feedback session we had with the members of the team, they provided some reasons for that. For example, in order to really start using the artifact in their day to day work the design would need to be improved. Next to that they would like to see some differentiation in the completeness metric of the dashboard. It would have been useful for the team if there was a difference between required fields being empty and non required fields. This is not yet implemented on the dashboard and would have made the ease of use scores a little higher. Nevertheless, the results are very interesting and show that this artifact can really contribute to improving the insight into the data quality of the registering, ultimately improving the data quality itself.

6 Discussion and further research

With our research we have created visualizations of the data quality of the Dutch algorithm register. These visualizations can contribute to the improvement of the quality of the registrations. However, there were some limitations to our research, that we would like to discuss here.

First of all, the visualization dashboard is not finished design wise. We are not that skilled in UI/UX design and this was therefore quite hard to implement. However, a better design probably would have contributed to a better user experience and a more informative dashboard. Next to this, some of the visualizations on correctness could be improved. The rules are quite harsh now, for example, when a URL field contains more than just a URL, the field is marked as incorrect, while it can be argued that that entry is still correct. The same goes for the completeness visualization. It would be a nice addition to have a differentiation on important and required fields and less important and non required fields.

There are always new questions to be answered and during this research we stumbled upon some questions that might be worth answering in the future. For example, it could be quite interesting to investigate where the most algorithms are implemented and registered. Compared to the population of those areas it's interesting to see which provinces score the best on algorithm registration and which provinces use a lot of algorithms.

Next to this addition to the dashboard itself, it could be a big improvement for the data quality to check quality before an entry into the register is even possible. This would mean that the checks we now implemented in the backend of our software, would be run during the entry process, providing immediate feedback to anybody that's registering algorithms.

Lastly, the dashboard could be improved by enabling users to see the quality scores per algorithm. Currently, the dashboard only gives information per field or organization, not per algorithm. This feature would enable the team to try to improve more specific entries.

7 Conclusion

The main research question for our thesis was: *What can be improved on the current algorithm register when it comes to data quality and data structure, and how can visualizations such as dashboards and KPIs contribute to this improvement?* Based on our research on the data quality of the register and the artifact we created to visualize the data quality of the register, it can be concluded that the current quality of the data in the register is not yet living up to the expectations of an accessible and transparent Dutch algorithm register. A lot of data is missing, texts are too difficult and some important fields that could provide extra information (such as email or URL fields), are left empty or are incorrectly filled out. We can also conclude that the structure of the data can be improved by using typed fields and enumerators where possible. This will take care of filling out correct email addresses and URLs, but will also make the data a lot cleaner. Lastly, we can conclude that the artifact we built to visualize the data quality of the register, can contribute largely to the improvement of the data quality in the register. It gives insight to the team working on the register where things need to be improved. They can coordinate the process of improving the quality using the dashboard.

When we started this research, we expected results like we found. The register is still in an experimental phase, the registrations are on a voluntary basis and there are no consequences for government entities if a registration is incomplete or incorrect. That's why we tried to improve the quality of the register by visualizing the quality on a dashboard for the team working on the register. In the current situation they are able to make some simple improvements on the quality of the register, while the obligation by law for government entities is on its way.

Based on these conclusion, we recommend that the team implements enumerators where possible and tries to verify the typed fields (like emails and URLs) upon registration. Next to this we recommend the team to analyze the results of the data quality on the dashboard and try to contact the government entities to get them to improve their registration. This will probably have the most effect on the short term data quality of the register.

This research has shown that the Dutch algorithm register is a very good concept that has the potential to create a lot of openness on the use of AI and algorithms by the Dutch government. However, we have also seen that the register has a lot of improvements to be made before that potential is reached. This has been shown through the analysis of the current data quality. Next to analyzing the quality, we have also built an artifact that can concretely help improve the data quality in the short term.

References

[1] Ken Peffers et al. *The Design Science Research ProcessFIX ME!!!!: A model for producing and presenting Information Systems Research*. Jan. 1970. URL: https://jyx.jyu.fi/handle/123456789/63435?show=full.

[2] Aline Dresch, Daniel Pacheco Lacerda, and José Antônio Valle Antunes. "Design Science Research". In: *Design Science Research: A Method for Science and Technology Advancement*. Cham: Springer International Publishing, 2015, pp. 67–102. ISBN: 978-3-319-07374-3. DOI: 10.1007/978-3-319-07374-3_4. URL: https://doi.org/10.1007/978-3-319-07374-3_4.

[3] Nikola Marangunić and Andrina Granić. "Technology acceptance model: a literature review from 1986 to 2013". In: *Universal Access in the Information Society* 14.1 (Feb. 2014), pp. 81–95. DOI: https://doi.org/10.1007/s10209-014-0348-1. URL: https://link.springer.com/article/10.1007/s10209-014-0348-1.

[4] *Belastingdienst wist in 2019 al van etnisch profileren toeslagenaffaire*. URL: https://nos.nl/nieuwsuur/artikel/2436477-belastingdienst-wist-in-2019-al-van-etnisch-profileren-in-toeslagenaffaire.

[5] *Ook na jaar kunnen uit huis geplaatste kinderen nu terug naar ouders*. URL: https://nos.nl/artikel/2425119-ook-na-jaar-kunnen-uit-huis-geplaatste-kinderen-nu-terug-naar-ouders.

[6] *Zoek een algoritme*. URL: https://algoritmes.overheid.nl/nl/algoritme?search=.

[7] *— Tweede Kamer der Staten-Generaal — tweedekamer.nl*. https://www.tweedekamer.nl/kamerstukken/moties/detail?id=2021Z18926&did=2021D40605. [Accessed 18-08-2023].

[8] Danilo Doneda and Virgilio AF Almeida. "What is algorithm governance?" In: *IEEE Internet Computing* 20.4 (2016), pp. 60–63.

[9] Maranke Wieringa. "What to account for when accounting for algorithms: a systematic literature review on algorithmic accountability". In: *Proceedings of the 2020 conference on fairness, accountability, and transparency*. 2020, pp. 1–18.

[10] Mark Bovens. "Analysing and assessing accountability: A conceptual framework 1". In: *European law journal* 13.4 (2007), pp. 447–468.

[11] Jakko Kemper and Daan Kolkman. "Transparent to whom? No algorithmic accountability without a critical audience". In: *Information, Communication & Society* 22.14 (2019), pp. 2081–2096.

[12] Amina Adadi and Mohammed Berrada. "Peeking inside the black-box: a survey on explainable artificial intelligence (XAI)". In: *IEEE access* 6 (2018), pp. 52138–52160.

[13] Daniel Neyland. "Bearing account-able witness to the ethical algorithmic system". In: *Science, Technology, & Human Values* 41.1 (2016), pp. 50–76.

[14] Christo Wilson et al. "Building and auditing fair algorithms: A case study in candidate screening". In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. 2021, pp. 666–677.

[15] *Uniform Employee Selection Guidelines on Employee Selection Procedures — uniformguidelines.com*. https://www.uniformguidelines.com. [Accessed 28-Jun-2023].

[16] *What is Adverse Impact? And Why Measuring It Matters — hirevue.com*. https://www.hirevue.com/blog/hiring/what-is-adverse-impact-and-why-measuring-it-matters. [Accessed 28-Jun-2023].

[17] Averill Campion et al. "Overcoming the challenges of collaboratively adopting artificial intelligence in the public sector". In: *Social Science Computer Review* 40.2 (2022), pp. 462–477.

[18] Petra Heck. *A Quality Model for Trustworthy AI Systems - Fontys — fontysblogt.nl*. https://fontysblogt.nl/a-quality-model-for-trustworthy-ai-systems/. [Accessed 26-07-2023].

[19] *ISO/IEC 25010:2011 — iso.org*. https://www.iso.org/standard/35733.html. [Accessed 26-07-2023].

[20] *ISO/IEC 25059:2023 — iso.org*. https://www.iso.org/standard/80655.html. [Accessed 26-07-2023].

[21] European Commission. *AI Ethics Guidelines*. https://www.aepd.es/sites/default/files/2019-12/ai-ethics-guidelines.pdf. [Accessed 26-07-2023].

[22] Khan Mohammad Habibullah, Gregory Gay, and Jennifer Horkoff. "Non-functional requirements for machine learning: understanding current use and challenges among practitioners". In: *Requirements Engineering* 28.2 (Jan. 2023), pp. 283–316. DOI: 10.1007/s00766-022-00395-3. URL: https://doi.org/10.1007/s00766-022-00395-3.

[23] K Pfeffers et al. "The design science research process: A model for producing and presenting information systems research". In: *Proceedings of the First International Conference on Design Science Research in Information Systems and Technology (DESRIST 2006), Claremont, CA, USA*. 2006, pp. 83–106.

[24] Dutch Government. *Over Het algoritmeregister*. URL: https://algoritmes.overheid.nl/footer/over.

[25] *Nauwelijks zicht op 'zwarte zoemende dozen' van overheid: 'Algoritmeregister wassen neus' — nos.nl*. https://nos.nl/artikel/2479334-nauwelijks-zicht-op-zwarte-zoemende-dozen-van-overheid-algoritmeregister-wassen-neus. [Accessed 18-08-2023].

[26] *AlgoritmeRegister GitHub*. URL: https://github.com/MinBZK/Algoritmeregister.

[27] *Flesch Reading Ease Score*. URL: https://en.wikipedia.org/wiki/Flesch%E2%80%93Kincaid_readability_tests.

[28] EF - Education First. *English B2 level - CEFR definition and tests — EF SET — efset.org*. https://www.efset.org/cefr/b2/. [Accessed 13-Jul-2023].

[29] *FastAPI — fastapi.tiangolo.com*. https://fastapi.tiangolo.com. [Accessed 25-07-2023].

[30] *Vue.js - The Progressive JavaScript Framework — Vue.js — vuejs.org*. https://vuejs.org. [Accessed 25-07-2023].

[31] *Apex Chart JS*. https://apexcharts.com. [Accessed 30-07-2023].