MASTER

An interpretable model for rework prediction at WVDB Adviseurs Accountants

de Vries, Timothy

Award date:
2020

Department of Mathematics and Computer Science
Information Systems Group

An interpretable model for rework prediction at WVDB Adviseurs Accountants

Master Thesis

T. de Vries

Supervisors:
B.F. van Dongen (TU/e, Information Systems WSK&I)
S.A.A.H.G. Verhoeven (WVDB Adviseurs Accountants)
B.J.M. Wijlaars (WVDB Adviseurs Accountants)

19-11-2020

Management summary

In this thesis, we look into the audit process. The aim is to develop a model to predict rework and find typical causes for rework within the audit process.

During the first part of the thesis, the audit process was analysed. In particular, we looked into attributes of the process that distinguish it from other, more typical process mining, processes. This was done based on the process as documented in the standards by governing bodies such as the NBA or AFM, and by interviews with auditors working at WVDB Adviseurs Accountants. During this phase, a large number of characteristics specific to the audit process were found, which are not common in other processes (See Table 1).

The unique set of characteristics of the audit process was taken into account when creating a list of requirements for the modelling technique. As the model is to be used within the audit process, the requirements by the audit process also apply to the prediction model (e.g. the requirement to be able to explain every decision). A lot of typically used modelling techniques are not usable when applied to the audit process (see Table 3) because they violate the functional requirements or have technical limitations. Several modelling techniques were scored on the requirements and random forest models scored best, while not violating any of the requirements.

As the aim of this thesis is to both predict rework and find root causes for rework, the trained model has to fit both purposes. Within the audit process, the cost of an error is large compared to additional time spend. Therefore, it is important to minimize false negatives. However, for finding the root causes for rework, the distribution of false predictions does not matter. Because of this distinction, two different random forests with specific hyperparameters (found using a grid search) are trained, both serving one of the purposes. There is a large difference between the performance of the models, as the accuracy is respectively 92.9% with 2252 false negatives and 81.8% with 608 false negatives (see Table 8 and Table 9). Both models had similar ROC curves (see Figure 4) and the area under the curve was respectively 0.87 and 0.89.

Besides the random forests, a basic neural network was trained. This neural network was used as a baseline to compare the performance of the random forests against. The loss function which was used to train the neural network was based on maximizing the accuracy. The trained network could predict with an accuracy of 89.0%, with 2393 false negatives. When compared to the random forests, one sees that the performance of those can actually outperform the baseline neural network.

The trained random forest models can be used in two ways. Firstly, they can be used to find key indicators for rework in the audit process. For this purpose, one can determine from the trained random forest which features have the most predictive power (determined by the highest cumulative gini-gain). For this purpose, the model with the highest accuracy was used, as the distribution of false predictions does not matter in this case. When verifying the results with auditors at WVDB Adviseurs Accountants, the results of the model were in line with their views and confirmed their expectations. However, most of the parameters are related to the complexity of the task (where a more complex task has a higher rework probability) or the time of review (where a later review has a lower rework probability). The outcome of this analysis does not directly result in improvements for the process (as one cannot simply change the process to tune one parameter without touching others), but more general process improvements can be deducted.

Secondly, the models can be used as an operational support tool, giving auditors real-time feedback when tasks are marked as reviewed. For this purpose, it is important to minimize the probability of

false negatives, as they could lead to errors or neglect in the audit file. As the model not only makes binary predictions but also gives class probabilities, one has to look into the ROC curve as well. Based on this curve, one can see that a large proportion of the models’ predictions is predicted with a large certainty, and these predictions are often correct. When implementing this model as an operational support tool, the most important aspect is for auditors to trust the predictions made by the model. When discussing the model results with auditors at WVDB Adviseurs Accountants, they do acknowledge that the model could be helpful in their work. They are however hesitant to blindly trust the model, as it still predicts 1.4% false negatives. Being able to understand how the model works, as opposed to a black-box model, increases the confidence they have in the model. Therefore, the model is very suitable as an operational support tool but should be used in addition to the auditor and not be blindly trusted upon.

Table of Contents

1 Introduction

1.1 Context and Topic

Every year, large corporations publish their annual reports. These reports provide deep insights into how the company is doing and are therefore used by stakeholders to get an idea of the company. As there are a lot of different stakeholders (e.g. shareholders, investors, banks, and the government) it is important that the annual statements of a corporation are correct and give a true impression of how the corporation is doing. Because the information in annual statements of corporations has a lot of influence on (financial) transactions (e.g. tax, investments, and loans), it is important, not just to the corporation and its direct stakeholders, that the annual statements can be trusted as the effects of errors can create a ripple effect and impact a large number of people. Because of the importance of the annual statements corporations need to get a statement from an independent auditor stating their annual report represent the actual position and proceedings. Because of this role, auditors can be seen as protectors of the public interest. This purpose of these audits is **the same across the entire world**, and the process is therefore very similar in different countries and different auditing firms. Furthermore, the audit process is **strictly regulated** by national governing bodies, such as the NBA in the Netherlands.

In this paper, **the audit process from the auditors' point of view is evaluated**. During the audit process, the auditors indicate risks where the annual statements might differ from the actual situation. They then verify if the risk is actually there and if it has been mitigated by the corporation being audited. As one can imagine, the larger and more complex the corporation is, the more intense the audit process is. The audit process is not automated but performed by humans, which makes it prone to errors, especially oversight. Furthermore, there is a large liability for auditing firms and individual auditors when they oversee things or make a mistake or wrong judgement. In extreme cases, auditors and their firms can lose their auditing licence. To ensure high quality, auditor firms let senior staff review all work being done by the (cheaper) junior staff. Because most of the work is being performed by less experienced auditors, the process has a lot of reworks, which require additional hours spend and make the process more expensive.

Auditor firms, just like every company, want to minimize their costs. In the audit process, this means minimizing the required reworks and expensive hours of senior personnel. As most of the inefficiencies in the process lie in the reviewing and reworking, optimizing this process could result in large cost savings. Currently, the entire review process is performed manually by the responsible senior auditor. In this paper, the executed audit process is being mined and a structured, automated way of finding possible inadequate work (and therefore required rework) is investigated. Furthermore, the root causes of this rework are being investigated.

1.2 State of the Art

1.2.1 Audit process

Accounting is an old profession that is facing increasing pressure from external parties to monitor and improve the quality of their audit processes (Kinney, 2005; O'Regan, 2010; Sutton, 1993). Users of financial statements perceive audit reports to provide absolute assurance that the financial statements have no material misstatements and do not perpetrate fraud. There is however a gap between the assurances auditors provide and the expectations there off (Geiger, 1994).

There is a vast body of literature relating to audit quality and its measurement. Despite this, no single generally accepted definition or measure of audit quality has emerged yet (Aghaei Chadegani, 2013). Most literature derives from DeAngelo's definition. He defines audit quality as the marketassessed joint probability that a given auditor will both detect material misstatements in the client's financial statements and report material misstatements (DeAngelo, 1981). This means that the quality of the audit is a function of the auditor's ability to detect material misstatements (technical capabilities) and reporting the errors (auditor independence). In 1988, Palmrose defined audit quality in terms of the level of assurance (Palmrose, 1988). As the purpose of an audit is to provide assurance on financial statements, audit quality is the probability that financial statements contain no material misstatements. This definition uses the results of the audit to reflect audit quality.

Because audit quality has different definitions for different people, the quality has been investigated with a variety of perspectives in the literature (Aghaei Chadegani, 2013). Most studies focus on the audit as a given, and access the quality by the result of the audit, eg by examining their amount of and/or success in litigation (Palmrose, 1988). The commonality in this research is the attempt to evaluate the quality of audit work by ex-post reactions to the outcome of the completed process.

While an understanding of how audit quality is important, such ex-post measures based on the output of the process fail to aid in determining what could have been dome during the process to improve the quality of the audit. Considerably less research has been done from the perspective of the auditor regarding its own audit processes. Literature focusing on the perspective of the auditor often uses input proxies to estimate audit quality (Aobdia, 2015).

Until this date, no academic research is performed on the executed process. This could be because of the secrecy most auditors keep of their files, or because of the flexible nature of the audit process. The use of information technology, such as workflow management systems, to control the audit process changes the auditors' behaviour (Dowling & Leech, 2014). Also, commercial parties (eg Oracle) started leveraging the execution of audit processes (King & Magnusson, 2003).

1.2.2 Process mining

One could argue that while no specific research has been done on the audit process, research on other processes could also apply to the audit process. Process mining is a relatively young research area but has already a lot of applications (W. van der Aalst et al., 2012). The field of process mining provides new techniques to discover, monitor and improve processes.

Process prediction extends process mining from a posthoc analysis method to real-time operational support (W. M. P. van der Aalst et al., 2010). Most existing process prediction research focusses on the prediction of process outcomes, including the time to completion, rather than predicting the existence of rework in a subprocess, as we do here (Verenich et al., 2019). Most of these techniques rely on an explicit model representation (eg a state-transition or Markov model) for their predictions (Breuker et al., 2016; Ceci et al., 2014; Unuvar et al., 2016). As users can add risks to the audit process during its execution (King & Magnusson, 2003), one cannot make an explicit model representation of the entire audit process based on historical data, making these techniques inapplicable.

Evermann, J. proposes a technique that does not require an explicit model but uses a recurrent neural network to predict the next process step (Evermann et al., 2017). While this could be applied to the audit process, we are not only interested in the next step in the process, but interested in the probability of rework happening at any number of process steps in the future.

Currently, there are no techniques that focus on predicting rework, nor predicting the probability that a specific activity (rework of a task) is going to happen in the process.

1.3 Research Question

The audit process is a typical process that has some unique characteristics. This thesis focusses on providing a method to build a real-time operational support model, that can be used to predict task rework in an audit process being executed, and finding the root causes for rework. The following research goal is determined:

> **Develop an interpretable model to predict rework in the audit process and find typical causes for rework.**

To reach this goal, the following research questions will be answered:

- How does the audit process differ from other processes?
- What interpretable techniques/algorithms can be used to make predictions on rework tasks within the audit process?
- How can these techniques/algorithms be applied to the specific context of auditing at WVDB Adviseurs Accountants?
- What are the typical causes for rework in the audit process at WVDB Adviseurs Accountants?

This thesis is structured as follows (see Figure 1). In chapter 2 we look into the audit process and how it differs from other processes. In chapter 3, we investigate which interpretable techniques/algorithms there are and which is most suitable for the audit process. Furthermore, we discuss how this technique/algorithm can be applied to the specific context of auditing. In chapter 4, we train the model using the data of the audit process at WVDB Adviseurs Accountants and discuss the performance of the model. In chapter 5, we use the model to find the root causes for rework at WVDB Adviseurs Accountants and provide a human interpretation for these causes.

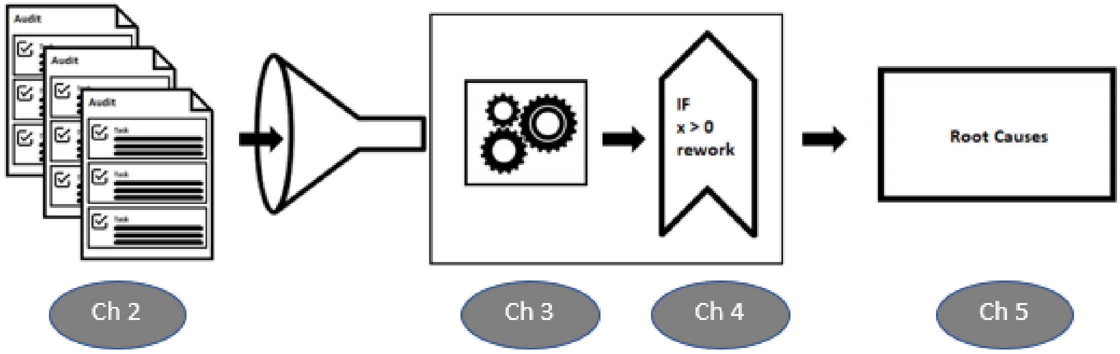


Figure 1: Process in this thesis

1.4 Method or Approach

During this study, the CRISP-DM methodology is used. This methodology describes a common approach for data and process mining, existing of six phases (business understanding, data understanding, data preparation, modelling, evaluation, and deployment) as shown in Figure 2. The deployment phase is out of scope for this master thesis.

To answer the first research question, a combination of the NV COS (Nadere Voorschriften Controle- en Overige Standaarden (Dutch auditing standards, based on the International Standard on Auditing (ISA)), literature available on studies about audit quality in the Big Four (the four largest auditing firms in the world, currently Deloitte, Ernst & Young, KPMG, and PricewaterhouseCoopers) and interviews with auditors at WVDB Adviseurs Accountants will be used. Combining these resources will give a good idea of what the audit process should look like, and what differentiates the audit process from other processes. Also, this will provide a good idea of the available data being logged in the systems during the process. This is part of the first two phases of the CRISP-DM model (business understanding and data understanding).

After we have a clear business and data understanding, a set of requirements can be made of what potential prediction techniques/algorithms should be able to do. Comparing available candidate techniques/algorithms with the requirements should result in a modelling technique best suited for the task.

After this, we will look into how the selected technique/algorithm can be applied in the audit process. This includes the data preparation and modelling of the model (phase three and four of the CRISP-DM model). The data preparation includes determining the required data to feed the model, as there is probably more information stored in the system than required by the model, and data transformation as the data most likely cannot be fed into the model directly.

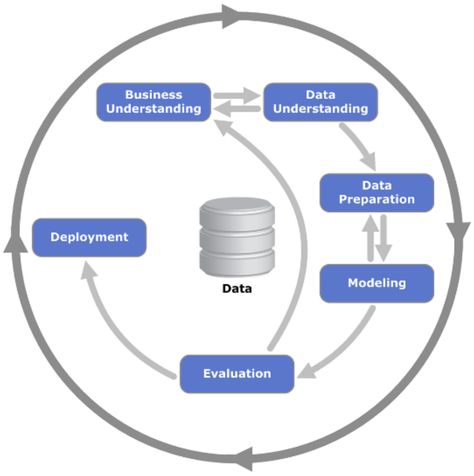


Figure 2: CRISP-DM model

After an appropriate modelling technique is chosen and the data is prepared, the model is trained. For this, we use a case study. This case study consists of the audits performed by WVDB Adviseurs Accountants over a three year period, the fiscal years from 2016 until 2018. During this case study, the developed model will be trained and interpreted (the evaluation phase of the CRISP-DM model). During this interpretation, we will look into the most typical causes for rework within the audit process and do recommendations on the audit process as implemented by WVDB Adviseurs Accountants.

2 The audit process

One of the core activities of accountants is auditing the annual statements of other corporations. The purpose of an audit is to reduce the knowledge gap between the corporation providing its annual statements and the stakeholders of these annual statements, by increasing the trustworthiness of the validity of the information in the annual statements. This results in an auditor's report, in which the auditor states his judgement of how accurate the annual statements represent the actual state of the corporation. This judgement is used by stakeholders of the corporation (e.g. banks providing loans, shareholders, customers) and the government to ensure that the information they get from the company is to be trusted, which enables them to make well-founded decisions and prevents them from facing hidden catches. Besides the auditor's report, which is the core product of the audit, the auditor also has a duty of care ("zorgplicht" in Dutch, which is one of the five ground responsibilities defined in the professional conduct for Accountants in the Netherlands) for its clients, which means he will report everything he finds that is of interest to the corporation in a management letter.

Corporations are often obliged by law to have their annual report audited by an independent accountant. In the Netherlands, a corporation is obligatory when in two consecutive years two of the following three criteria are met:

- The turnover is over 12 million euro,
- The balance sheet totals over 6 million euro,
- There are more than 50 employees.

Besides statutory audits, there are also voluntary audits. There are multiple reasons to do a voluntary audit. Often they are done by corporations that want to increase the trustworthiness of their annual statements (e.g. non-profits that want to prove that the donations are handled well or a corporation wanting to request a big loan), corporations that are just below the requirements to be obligated for a statutory audit, or corporations that have shrunk below the statutory audit requirements.

For corporations, the audit of their annual report can often feel like an obligation and a burden. However, the audit provides the corporation not only with the auditor's report which they can present to their stakeholders but also with deep insights in the state of affairs within their company. The auditor looks for hidden catches, oversights and risks within the corporation, which are often also not clear or addressed by the management.

2.1 Process

In essence, the audit is very simple. The board of a corporation is obliged to make an annual report that is a good representation of the company's (financial) position. The auditor ensures himself that the annual reports are indeed a good representation of the company's position and writes up a document in which he states so. However, before the auditor can say anything about the annual reports and the state of the company, there are lots of things that need to be checked and taken into account. The auditing process consists of four major phases: The planning & risk analysis, the interim audit, the final audit and the closure & audit report. During the audit, the auditors are looking for discrepancies between the actual position of the corporation and the documented position in the annual report. The chance of such a discrepancy is considered a risk during the audit.

It is important to note that while there is a planning & risk analysis phase at the beginning of the audit, risk analysis is an integral part of the audit and has to be reconsidered at all times. As a result, additional risks can be included in the audit process at any point during the audit, even if the initial risk analysis is long finished.

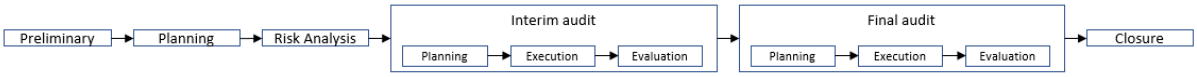


Figure 3: Audit process phases

2.1.1 Planning & risk analysis

Before the audit can start, the auditor needs to get to know the basics of the corporation and determine if he is allowed to accept the client and the audit order. This includes checks such as compliance with the WWFT (Wet ter voorkoming van Witwassen en Financieren van Terrorisme, the law for preventing money laundering and financing terrorism) and conflicts of interest. These checks are prescribed by external parties/regulations, such as the VGBA (Verordening gedrags- en beroepsregels accountants, the professional conduct for Accountants in the Netherlands) or by internal regulations. If things come up during these checks, additional measures need to be taken (e.g. an independent quality check) or in extreme cases, the audit cannot be performed by the auditor or firm and needs to be delegated or even aborted.

After accepting the audit, the scope and materiality are determined. The materiality is a threshold which refers to the impact of an omission or misstatement of information in the annual reports. If the omission or misstatement is small and users of the annual report would not have altered their actions, then it is considered immaterial. This materiality determination is partly arbitrary but is the base of the audit process as this determines how strict the checks need to be. One can imagine, if large variations are allowed, checking if the deviations are within the thresholds is less work. The materiality needs to be benchmarked to determine if it is appropriate for the size of the corporation (larger corporations can have a larger materiality). During the audit, **a risk is defined as a chance that there is a specific material discrepancy in the annual report**. The scope of the audit is very important, as corporations are often a group of different companies, which all have different risks.

After the scope and materiality of the audit are determined, the auditors make a planning of which activities need to be performed during the audit. This involves judging what work needs to be done in the project and signalling significant risks that need to be checked during the audit. Some risks are very standard and occur in a lot of companies (e.g. creditors/debtors, liquid assets, equity) and are therefore part of the standard work program, but there could also be specific risks for the company (e.g. bribery of government employees) that the auditor needs to identify during this phase.

2.1.2 Interim audit & final audit

Both the interim audit as the final audit exist of three phases:

- Planning,
- Execution,
- Evaluation.

During the planning phase, the auditor (again) thinks about risks within the audit. If he finds new risks, these risks are taken into the audit. For each identified risk, the auditor documents what needs to be done to determine if the risk is not applicable, sufficient disclosed with the annual reports and therefore is mitigated, or within the materiality determined previously.

During the execution phase, the work determined in the planning phase is executed and documented. During the execution, the impact of the execution is taken into account. If the auditor finds that the financial reporting is not good enough and there is a material impact, this should be reported to the client. The client then has the option to explain to the auditor why things are recorded as such in the annual report. If the client agrees that his records are indeed wrong, he has the opportunity to change them. If the client chooses not to update his records, the auditor should start an inquiry to find out why and to make sure that the client does not intentionally break the law. These discrepancies are recorded on an error list but do not prohibit the process from continuing or limit the auditor to give his judgement in the auditor's report. During this phase, it is important to keep the independence of the auditor with the client into account.

Finally, in the evaluation phase, each risk is being evaluated. Based on the executed work, the accountant should have all the information needed to determine if the risk has a material impact on the annual report. If during this phase, the auditor needs additional information to make the decision, this is documented and the necessary work executed. The best outcome of the evaluation is a risk being immaterial. If this is not the case, the auditor can discuss with the client to change the annual report or add the deviation in the auditor's report.

An example of a risk which is part of most standard work papers is, if a corporation has debtors (other corporations/persons that ow money to the corporation). The auditor works under the hypothesis that there is no risk related to debtors, but has to validate this hypothesis. A typical check for this example could include:

- Documenting the scope of this risk and what will be checked
- Doing a high-level data analysis of this post and discussing it with the client
- Comparing the general ledger with the debtor administration
- Looking for special debtors (large amounts / creditamounts / internal debtors / taxes / pensions / loans to the board or stakeholders) and determine the impact on the annual statements
- Determine if the debtor amounts are calculated against the correct exchange rates
- Doing a test of details, subsequent cash collection for open items at years end
- Check if the reported dubious debtors' buffer is financially justified
- Check the dubious debtors based on the ending-/originate check and outstanding position

The most common risks are already identified and part of the standard work papers. Based on some basic questions in the first planning phase, these could be already mitigated if the risk does not apply to the corporation. In general, the audit works with the hypotheses that a risk is not applicable and this hypothesis is validated by appropriate data-focussed tests. In a typical audit, an auditor identifies about 10 additional risks, which each have a different impact on the annual reports and different checks. In the end, all risks need to be evaluated and the remaining impact on the annual report determined.

2.1.3 Closure & audit report

When all the risks are sufficiently covered, the final phase starts. In this phase, the auditor makes conclusions based on the evaluated risks and writes the auditor's report. In this report, the auditor presents his reasoning and final judgement in a statement. Typically, companies want an unqualified opinion, meaning that the auditor didn't find any discrepancies in the annual report and there are no hidden audit risks. As this is not always possible, the auditor can decide to give a different opinion. Besides the unqualified opinion, an auditor can provide a qualified opinion, adverse opinion, or disclaimer of opinion.

A **qualified opinion** is given when the auditor was not able to provide an unqualified opinion, but the annual report does not misstate the financial position as a whole. This could have multiple causes and does not directly mean that the corporation did something wrong (eg if a company has a warehouse in Italy but due to a global lockdown nobody is allowed to verify that the stated stock is indeed in the warehouse). When giving a qualified opinion, the auditor provides an additional paragraph to point out why no qualified opinion could be given. An **adverse opinion** is given when the auditor finds material discrepancies between the annual reports and the actual position, that the client does not want to change. An adverse opinion is often an indication of fraud or other misconducts and the auditors are obliged to report this to the AFM (Autoriteit Financiele Markten, the Dutch financial governing body). It is also possible that the auditor is unable to complete an audit due to the absence of financial records or insufficient cooperation from the client. In this case, a **disclaimer of opinion** is issued.

During the final phase, there could be independent quality checks within the accountant's organisation. These checks are sometimes mandatory (eg when the corporation that is being audited is in a risky business segment or when the external accountant changed) or could be randomly selected. The purpose of such checks is to ensure the audit file is correct and the team is truly independent of the corporation.

2.2 Process at WVDB Adviseurs Accountants

As there are strict regulations for the audit process, it makes sense that the audit process at WVDB follows the same general structure as discussed in section 2.1. However, there is a little freedom for firms on how to implement the process.

2.2.1 Planning & risk analysis

At WVDB, the acceptance of the orders is not recorded in the same software as the rest of the audit, but in a central customer relationship management system. The outcome of these checks is recorded in the audit file, making it part of the audit. This is because the order acceptance is not to be determined by the audit team alone. If the audit has an above-normal risk, the accounting board WVDB has to approve the order. They oversee the entire company and are therefore able to determine if the risk is acceptable to WVDB.

When all the preliminary checks are performed, and the audit assignment can be accepted, the actual audit process starts. At WVDB, the process is started with a meeting where the audit team discusses the company and prepares the yearly work papers. In this document, they state the scope of the audit, the materiality, and most important, what they need to do to be able to cover everything in scope and be able to give an auditor’s report. During this phase, they also access if there are special circumstances (either required by law or by WVDB) that prevent the process from being executed as planned (eg. To ensure independence, the external accountant should rotate every couple of years or an extra (internal or external) quality check should be performed). Part of the work papers is prepared using a default template, which includes a questionnaire to identify a lot of frequent risks and standard checks that are applicable to most corporations. This template is enriched with additional risks, based on the specific corporation under audit.

These yearly papers are shared with the client, so the client knows what to expect of the audit. Furthermore, most of the checks require information prepared by the client. As all the required work is recorded in the yearly work papers, the information required by the client is also recorded in this document. After the client agrees to the scope as proposed in the work papers, the required work is recorded in Auditor and the first phase of the project is completed.

2.2.2 Interim audit & final audit

The interim audit at WVDB is relatively simple compared to the final audit. During the interim audit, not all risks are being evaluated. The main focus of the interim audit is to identify things that could lead to problems during the final audit (Eg. a substantially large amount of long-standing debtors). This enables the client to resolve some of the issues that could otherwise arise during the final audit and would prevent the external accountant to provide an unconditional auditor's report or would result in delays in the process. For the risks during the interim audit, the performed work and outcomes are logged in Auditor and are reviewed by the external accountant. If required, rework can already occur at this stage.

The final audit is where most of the work is performed by the audit team. All the work steps as described in the yearly work papers are performed by the auditors and recorded in Auditor. During this phase, there are differences in how the external accountants work. Some work very closely with the team and guide the auditors a lot during the work steps, while others guide the team by reviewing the finished work steps. In both cases, if the external accountant deems the work sufficient to assume the hypothesis that the risk is mitigated this is recorded. After this, the task is deemed finished. In this thesis, reopening a task after it is reviewed by the external accountant is considered rework.

2.2.3 Closure & audit report

During the final phase, there could be quality reviews of the audit. WVDB as a firm is responsible to give guidance to their external accountants, and they are also responsible to test if the work performed by their external accountants is up to their standard. To test this, high-risk audits are reviewed by an independent auditor (either from WVDB or externally). Also, every external accountant has a yearly review on an audit file picked at random.

During this phase, most of the rework of the work steps happens. There are different reasons why, for example, because the quality review deemed the performed work not enough to assume the hypothesis that the risk is mitigated, or because the external accountant itself requires additional work to be performed when writing the final auditor's report.

2.3 Differences with typical processes

The differences between the audit process and other typical processes can be split into two groups, functional grounds (eg. the legal implications of the process) and technical grounds (eg. the flexibility of the process, making it impossible to establish a reference process model).

By law, not every accountant is allowed to lead audits. Only external accountants are allowed to lead audits, and they are the ones that are ultimately responsible for the audit. Because the external accountants are individually responsible for the audit report (as opposed to the company they work for) they have a large incentive to make sure their audits are of high quality. If it turns out that they neglected their work, they can lose their auditing licence (just like a doctor loses his doctor license if he breaks rules). Because the stakes for external accountants are high, everything is documented and multiple quality checks are in place. While the audit team exists of several people (ranging from junior assistants to senior partners), every task needs to be reviewed by the external accountant and/or a senior partner. This is done on a task level, meaning that during the audit, the external accountant and/or a senior partner reviews the work multiple times, after the first risk analysis, after the interim audit and after the final audit.

The audit process is unique in its output. Most processes have an identifiable end product. The audit process distinguishes itself, as it is a product in its own. The audit does not only deliver an auditor's report, but the full process of creating this report is part of the end product, as it is auditable by external parties. The audit also contains legal weight, as the outcome is used to reduce the knowledge gap and increase the trust between other parties. This requires a high level of trust in the audit process by external parties. Unique is that these parties cannot influence the process, but rely a lot on the quality of the process, while the ones executing the process do not directly rely on the outcome. In the eyes of the auditors, every audit process is successful (a qualified or adverse opinion is still a successful outcome of the process).

Most processes have clear process steps, that don't vary a lot over time. However, during audits, the audit team has to consider all possible risks. This can include very specific risks, that were not observed before. Therefore, the process can contain activities that have never been seen before. Also, there is no clear order into how the tasks must be executed. It makes sense to execute the audit following a natural order, but this is not enforced as during every phase of the audit the process steps can change. For a typical audit, there are over 200 different tasks that need to be executed. Combined with the unique tasks and free order of execution, this results in all audits being unique. Therefore, it is impossible to make a reference process model.

Category	Typical process	Audit process
Outcome	The outcome is only of interest to the process owner / the case responsible	The outcome of the process has legal weight
Outcome	Processes can fail	Always has an output (a qualified opinion is still a good output with respect to the process)
Outcome	The process itself is not part of the outcome	Every process step needs to be motivated and this can be audited.
Responsibility	Typically part of a large organization with no clear responsible per case	There is one person responsible for a process case and it's outcome, which requires a special, government-protected title and that person is personally responsible
Responsibility	The process owner is not the one executing the activities	The external accountant (the owner of the process) also does the reviewing and sometimes the execution.
Process activities	Typically a small number of activities	A large number of activities
Process activities	A limited set of activities which are known	People can add additional tasks resulting in additional activities
Process activities	Activities are handled once	The same activity is addressed multiple times (typically

		created during the risk analysis, work logged during interim and final audit and reviewed multiple times)
Process activities	If an activity is finished, it isn't reopened again (does happen, but not often)	Even after review / making a task definitive, it is often reopened
Flexibility	Set of possible flows through the system	Tasks within a phase can be executed in any order and are often being worked on simultaneously
Flexibility	Set of activities is known before the process execution starts	Tasks can be added at any point during the process
Endstate	Process has clear endstates	After the process is finished, (external) audits can happen which could reopen the process. Also, accessing the audit file after the audit is logged and part of the file
Recurrency	Often no clear recurrency	Process normally happens yearly for every corporation. Recurrent years have special requirements (eg changing the external accountant to ensure independence)
Type of work	Automated processes and clear manual steps	No pre-defined process model

Table 1: Differences between the audit process and a typical process (in process mining)

As stated, in the end, the process always reaches a successful end state with an auditor's report of sufficient quality according to the responsible external accountant. However, the process can go very good or bad. For WVDB, a good process is a process where everything went fluent and as planned. This means a process without rework, as this results in extra hours spend (and therefore additional costs) and if possible as less work as possible from expensive employees, such as external accountants and senior partners. Minimizing reworks, or being able to detect earlier what needs to be reworked, can result in large savings on hours spend within an audit process.

In conclusion, the audit process differs from more typical processes because it has a lot of quirks, making the process unique (see Table 1). Because of this, the process can not always be handled as one would do with a more typical process.

3 Predicting rework

When the field of process mining started to develop, the first techniques focussed on the analysis of historical data (W. van der Aalst et al., 2012). Today, however, process mining is not restricted anymore to analysing historical data but can be used for online operational support. Three operational support activities can be identified: detect, predict, and recommend. The moment the system identifies a case deviating or requiring attention, an alert can be generated. To be really useful, one would like to generate such alerts immediately, so one can act on them, instead of only once in a while.

In this chapter, we focus on how we can train an operational support model that predicts the rework of tasks. The purpose of this model is twofold. Firstly it should be able to work as an online operational support model, which supports the auditors in detecting rework early on in the process, improving the efficiency. Secondly, the models' workings should be interpretable, so one can determine key factors that impact the possibility of rework, and use this to change processes to prevent rework in the first place.

3.1 Algorithm

First, we will determine a set of requirements for the prediction algorithm. These requirements are determined based on the predictions we want to make and the type of process. After the requirements are determined, possible candidate techniques/algorithms are proposed and scored on the requirements. Finally, the best fitting technique is chosen.

3.1.1 Requirements

When selecting techniques or algorithms to make predictions, one should start thinking about what actually should be predicted. In our case, we are interested in **predicting the reopening of individual tasks** within the process as opposed to if there is any rework in the entire audit process. Therefore, techniques that look at the entire process and do not take the individual sub-process corresponding to the specific task into account will not suffice. However, as all tasks in the audit process are related the **entire process should be taken into account**. Failing to do so would potentially neglect a lot of information which could improve the model.

The usage of the model is twofold. Firstly, the model should be usable during the audit, as an **operational support tool**, to assist the auditors in detecting rework early on. Detecting potential rework as early as possible in the audit reduces the amount of work needed to improve the task. Secondly, the model should **provide insights** into the process in general. This means that by interpreting a trained model, one should be able to determine what are typical causes for rework.

The model is used in a very specific field, that puts additional requirements on the model. As the model will be used by auditors, who have an accounting background, the model should be **usable by non-experts** with respect to prediction models. However, in our case, one can assume that the users have a lot of domain knowledge.

Because the auditors are responsible for the validity of their work, and making errors could have large consequences to both the auditor as the firm, the users should be able to **explain/understand** how the model makes its predictions. Blaming a computer model for false predictions will not stand. Because of the large impact of missing things in the audit, a false negative is a lot worse than a false positive. Therefore, one cannot simply train the model by maximizing the accuracy, but one should be able to change the **performance measurements** such that the model for instance minimizes the number of false negatives.

As discussed, the auditors can add new risks to the audit. This would result in tasks that are unique to a specific audit. The model should be **able to deal with unknown data** that it hasn't seen before. All the requirements are shown in Table 2.

Requirement	Explanation
Predict the reopening of individual tasks	We are not interested in predicting if there will be any rework in the entire audit process, but we are interested in predicting which specific tasks require rework.
Take the entire process into account	All the tasks in an audit process are related. This means that the relation to other tasks holds information as well.
Provide operational support	The technique should be able to run online to provide operational support.
Usable by non-experts	The output will be used by auditors, non-experts with regards to process mining.
Explainable	Because of the nature of the process, auditors should be able to explain and document everything they do. Therefore, they must understand why the model makes a certain prediction.
Performance measurements	There can be large consequences if the wrong things end up in the auditor's report. Therefore, it is more important to minimize false negatives than improve overall accuracy.
Deal with unknown data	As auditors can add additional tasks to the audit, the technique should be able to deal with tasks not seen during training.
Provide insight	As we want to use the model to determine what are indicators of rework, it should be able to extract this from the model.

Table 2: Requirements of the modelling technique

3.1.2 Techniques / algorithms

Based on the requirements set in section 3.1.1, a list of potential techniques / algorithms that fit most of the requirements is composed. A full list of the candidate modelling techniques being considered is shown in Table 3.

The first technique being considered is a **linear regression model**. A linear regression model is very easy to understand, as it is in essence a single mathematical function, making it very explainable/understandable. However, a linear regression model cannot directly deal with categorical input parameters and requires them to be one-hot encoded. Because a lot of input variables are categorical, with a large number of possible values, this would result in a huge matrix being too large to handle by current generation computer systems.

A newer technique is a **neural network**. A neural network can discover complex relations in the data that are undiscoverable with other techniques. In our case, where the data is a combination of information from the specific task and information about the entire audit process, this could prove very beneficial in terms of performance. However, neural networks can be seen as a black box. It is often unclear on what grounds the neural network makes its predictions. This makes it impossible to explain why a decision is being made, which is one of the most important requirements.

A technique that is very understandable is a **random forest**. The output of a random forest is very similar to that of a decision tree, a tree with questions that sends you to a particular route down the tree and eventually to a leaf node, yielding a specific result. A random forest constructs multiple decision trees and combines them into one tree, to correct some of the overfitting habits of normal decision trees. The resulting trees are easy to understand, even by non-experts that haven't worked with prediction models before. Even more useful is that the trees can be used to determine which input parameters the model determines good predictors, either by looking at the splits high in the model or by calculating the cumulative gini gain on splits per predictor. The downside of such a model is that it is hard to find complex relations between the input variables, as these would often require a lot of splits.

A way to increase the performance of a model is to combine multiple models, for instance, the **combination of RF/NN**. To combine the models, one simply uses the prediction of the first model as an input parameter for the second model. This way, the second model can improve the prediction made by the first model. As both models have different prediction strengths, combining them in this way could result in a higher performance. However, combining them also includes the downsides of both models. For instance, the combination of the random forest and neural network would result in a model that is (partly) unexplainable and hard go gasp by non-experts.

A **Recurrent neural network** is a form of a neural network where connections between nodes form a directed graph among a temporal sequence. In process mining, each trace can be seen as such a directed graph. RNNs can store information about the current state of a process and can therefore also be used to predict future states of the process. In our case, we would be interested if one of the next states would be the reopening of the task. Because RNNs are a form of neural networks, they have the same downside: they behave like a black box and it is often unclear why an RNN makes a certain prediction.

Model	Predict the reopening of individual tasks	Take the entire process into account	Provide operational support	Usable by non-experts	Explainable	Performance measurements	Deal with unknown data	Provide insight
Linear regression	+	+	-	+	+	-	--	+
Neural network	+	+	+	-	--	++	+	+/-
Random forest	+	+	+	++	++	+	+/-	++
Combining NN/RF	+	+	+	-	--	+	+/-	+/-
RNN	+	++	+	-	--	+	+/-	+/-

Table 3: Modelling techniques fitting the requirements

In Table 3, the discussed techniques are scored against the requirements discussed in section 3.1.1. As one can see, all the proposed techniques are able to make the predictions that we want and take the process into account. The biggest difference is in the explainability and the usability by non-experts of the model, which is low with the techniques using a neural network. Based on the fit with the requirements, a random forest is the best suitable technique. However, it is possible that a neural network outperforms a random forest. Therefore, a neural network is also trained using the same data and used to compare the performance of the random forest.

3.2 Data

As explained in chapter 3.1.2, a random forest is used to predict task rework and a neural network is used to compare the performance against. Independent of the technique chosen, the quality of the data to train the model influences the real-world performance of the model.

Because of the legal implications, there are laws regulating that audits performed by accountants should be auditable by external organizations (such as the government). To be able to explain all decisions and trade-offs by the audit team, this should all be logged and part of the audit file. Because of this, one can expect the logging of the audit process to be very good, independent of the system used to log this process. To quantify this, to meet the requirements the level of logging should be 4 or 5 on the scale proposed in the process mining manifesto (W. van der Aalst et al., 2012).

Because of the extensive logging, there is also a lot of data logged that is not needed for this model. What parts of the log should be extracted should be driven by the questions one wants to answer, not simply by what is available in the log (W. van der Aalst et al., 2012).

3.2.1 Case selection

When creating the dataset, it is important to look at what is considered to be a case with respect to process mining. Obviously, one could argue that the entire audit is a case. However, the predictions are on rework for individual tasks that are part of the audit process and not the entire process. Therefore it is logical to use tasks within the process as cases. To include information about the parent (audit) process, case attributes can be used. Encoding the data this way enables one to make predictions on a single task while including information about the larger process.

As predictions are made during the audit process, one has to carefully select this moment. As we predict if a specific task will be reworked later on in the process, our prediction must happen after the task is successfully reviewed for the first time.

Using this approach, it is important to remember at what point in time predictions are made. As it is an online model that makes predictions about task rework during the audit process, not all information is available at the moment the prediction takes place. During the training phase, finished audit processes are used. This means that there is information stored in the database that was not yet available at the moment when the prediction is made. This information can obviously not be used to train the model.

3.2.2 Case / activity attributes

The first step in creating the model is to prepare the data. There is a lot of information stored in the database, but as a lot of this is uncategorized and free text, this cannot all be used without preparing. To prepare the right set of input parameters, one has to think about what would be good indicators for a review (W. van der Aalst et al., 2012). As we are only interested in making one prediction, at the moment when the task is reviewed, the attributes are encoded for a single predictive model (Ceci et al., 2014).

There has been no quantitative research yet on audit processes and what are indicators of rework, as most research measure the quality of the audit using post-audit measurements. In 1993, Sutton did a subjective research to develop and validate a set of key factors influencing the quality of the audit process and a corresponding set of measures for evaluating audit quality by surveying a group of experienced auditors. While these factors are indicators of the quality of the entire audit, some of them could also be a good indicator of rework when measured on a single task.

To find more potentially good indicators, and see what factors found by Sutton can be applied to our case, interviews with auditors at WVDB have been performed. Based on these interviews, the list of indicators as shown in Table 4 was determined and these indicators are used to feed the model. This table consists of a combination of attributes related to the entire audit and the specific task.

Attribute	Explanation	Related to the task or the audit
Start on	The time between the end of the fiscal year and when work on the task started	Task
First review on	The time between the end of the fiscal year and when the task was first reviewed	Task
First definitive on	The time between the end of the fiscal year and when the task was marked definitive	Task
Work time	The time between the first and last activity on the task	Task
Waiting time until the review	The time between the last work on a task and the review	Task
Review type	Indication if the review was performed by the external accountant or a partner	Task
First audit event	The name of the first activity	Task
Number of involved employees	Employees that handled the task (including the reviewer)	Task
Reviewer edits	Number of edits the reviewer made to the task before the review	Task
Reviewer interactions	Number of interactions the reviewer had with the task	Task
Absolute time from another review	The minimal time between this review and the previous/next review by the reviewer	Task/Audit
System audit phase	The system phase the audit is in	Task/Audit
Audit phase	The phase the audit is in (eg interim audit or final audit)	Task/Audit
Task identifier	An identifier for the task, compromised of the task folder and title	Task/Audit
Task title	The title of the task	Task
Audit phase progression	The percentage of tasks in the audit phase already reviewed	Task/Audit
Task folder progression	The percentage of tasks in a specific folder already reviewed	Task/Audit
Risk severity	The severity of the risk	Task
Number of attachments	The number of attachments	Task
Number of assign times	The number of times the task was assigned to someone	Task
Reviewer name	Name of the reviewer	Task

Work ended on	The time between the end of the fiscal year and when the work on the task was finished	Task
Duration start till review or definitive	The time between when work started on the task and the review	Task
Creation type	Indication if the task is part of the standard work papers or added as a custom risk	Task

Table 4: Attributes from the process used as input parameters

As explained in chapter 3.1.1, there are strong requirements on the traceability of the audit process. To fulfil these requirements, a high level of logging is available. Therefore, one can safely assume that all the indicators required can be determined based on the logging of the system.

3.2.3 Training / test data

When splitting a dataset into a training and test set, one should try to keep both sets as independent as possible. If cases within the sets influence each other, the model can learn strange behaviour during the training phase and result in overfitting. Because the sets are correlated, overfitting on the training data improves the results on the test data as well. When applying the model to new, real data, one would find out that the model performs badly.

Since individual tasks within an audit process are used as cases, it is easy to see that the cases are not independent (cases within the same audit process are obviously highly correlated). This means that the data cannot randomly be divided into a training and test set, as these two sets should be as independent as possible.

At first glance, one could think that tasks belonging to different audits are independent and assigning cases to the test or training set based on the audit they belong to is a good idea. This is not the case, as all audit processes are executed by the same company and therefore have shared resources (e.g. employees or natural disasters). For example, if an employee works on two audits at the same time and gets sick, both processes are affected. If one is in the training and one in the test data, the model could learn that the employee is sick and use that information to make a better prediction on the test data.

Because the audit processes are correlated, randomly splitting the cases or even using k-fold cross validation bears the risk of overfitting the model. To keep the training and test data as independent as possible, the cases should be split in time. The earlier cases are put in the training set, while the more recent cases are put in the test set. Since the audit process should be completed before a specified date (as required by the rules and legislation) and cannot start before the year is closed, it is easy to split the cases based on the year.

3.3 Training

Both model techniques selected in section 3.1.2 (random forest and neural network) cannot directly deal with textual / categorical variables. There are multiple ways to encode variables, depending on if they are ordinal or nominal. The most basic method is one-hot encoding, but as the cardinality of the variables is expected to be high, this results in very big matrixes that cannot be handled by current generation computer systems. Research has shown that label encoding, if possible based on the ordinality, does not perform significantly worse than more advanced encodings while being very explainable. In our data, most data is ordinal, even the textual names of tasks can be seen as ordinal, as they are part of a sequential process and therefore normally would follow each other in time. Therefore, (ordinal) label encoding is used to encode textual / categorical variables.

Even though a **random forest** prevents the over-fitting tendency of decision trees, this is still a pitfall. To tune the model, there are several hyperparameters that one can control. Just as with a decision tree, one can control the depth of the tree, by either limiting the depth, controlling the size of leaf nodes, or requiring a minimal information gain/impurity decrease for every split.

Random forest tries to introduce randomness to make individual trees more unique and thus reduce the correlation between trees and improve the overall performance. The randomness can be introduced by selecting a subset of the training set for each tree or by only trying a random subset of the available features for each split. Both of these options can be controlled by hyperparameters.

Part of the process of training the model is finding the best hyperparameters. The optimal set of hyperparameters is however not universally, as it depends a lot on the dataset. If, for instance, you determine an optimal value for the minimum size of leaf nodes for one dataset, and try to train the model using the same hyperparameter to a larger dataset it is very likely to overfit. Therefore, the tuning of hyperparameters needs to be performed on the dataset at hand. For this tuning, a grid search can be used to try multiple parameters.

As explained in section 3.1.2, the performance of the random forest is compared against the performance of a baseline **neural network**. Just as with a random forest, a neural network can be controlled with hyperparameters such as the type and shape of the layers. Finding the best shape for a network is often done by trial and error, as there are no scientific methods on determining the optimal shape of a neural network. Because we only use the neural network as a baseline to compare the performance of our random forest model against, a simple neural network is enough.

3.4 Evaluation

The performance of a model is often expressed in terms of accuracy, where a higher accuracy is obviously better. However, a high accuracy doesn’t always mean that the models' performance is good. For example, if the predicting outcome is skewed, a model can have a high accuracy while always predicting a particular class wrong. Therefore, depending on the domain and application of a model, other evaluation metrics can be used.

In our case, there are two different applications of the model. Firstly, the model can be used as an operational support system. As the audit focusses on finding risks within the audit, the model must support this goal. This means that the costs of a false negative could be a lot higher than a false positive. In the case of a false positive, one would spend unnecessary time on a task because it is indicated by the model, but in the case of a false negative one would neglect a risk and this could have huge consequences for the audit. Therefore, the model must **minimize the number of false negatives**. During the evaluation of this model, one could also look at the ROC curve, which plots the true positive rate against the false-positive rate. On the other hand, we are interested in finding the key indicators in the log for rework. For this purpose, the amount of false positives doesn’t matter, but we want the model to find the most influential key indicators. For this purpose, the model is trained to **maximize the accuracy**.

For both purposes, both models that are used in this thesis are trained. This way, the performance of the random forest can be compared.

4 Case study: Predicting rework at WVDB

During an audit, the auditors should leave no stone unturned within the corporation they are auditing. During this process, they often receive sensitive information that the corporation does not want out in the open. Because of this, there is a high level of confidentiality between the corporation under audit and the auditors, in which the responsibility of the audit file lies with the external accountant. To ensure that no confidential information escapes, auditors normally do not enclose information about the audit process. This behaviour is even stronger due to the risks of bad publicity for the auditors' firm if someone would find inconsistencies in audit files, something that has happened a lot over the last decade. As a result of this, there are no public datasets available on the audit process.

To validate the method discussed in chapter 3, a case study is performed at WVDB Adviseurs Accountants. WVDB is among the twenty largest accountancy firms in the Netherlands, with an audit department with about 60 auditors, of which 10 are external accountants, and a licence to accept all types of audits. Their audit department performs about 600 audits per year.

Because of the requirements set for audits by both the Dutch and the international governing bodies as discussed in chapter 2, audit firms have limited flexibility in how to implement processes. Therefore, it can be assumed that all audit firms have similar processes if they want to comply with the standards. For this thesis, this means that the method proposed in chapter 3 will yield similar results when applied to other firms than this case study.

During this thesis, the proposed method is implemented in Python 3.7, in combination with SQL to fetch the data from the databases. Python was chosen for the implementation because there are a lot of high quality, publicly available libraries for data modelling. The libraries used in this thesis include keras, tensorflow, sklearn, pandas, and numpy.

4.1 Data

Within WVDB, Unit4 Auditor is used to control the audit process and log the identified risks with the work and reasoning to mitigate the risk. Unit4 Auditor works both as a business process management (BPM) tool, making sure the auditors follow the right procedure, as a collaboration tool by enabling auditors to assign tasks and attach notes for other auditors at specific steps, and as a documentation tool, documenting all the reasoning by the auditors and thereby helping them to make a solid auditor's report.

Because Unit4 Auditor was designed with the auditing process in mind, including the legal consequences and liabilities, it has extensive audit trailing. The audit trail is split into 128 different actions, which refer to actions that can be executed within the application. Combined with the action performed, the employee performing the action, the time and date, the table and id on which the action is performed (if applicable), and a short description of the action is stored.

Not only mutations to the data are being stored in the audit trail, but also some read actions (eg. opening attachments) are stored. As the audit dossier could be consulted by auditors after the process is finished, and read actions are logged in the audit trail, one can see that the audit trail for a dossier in Unit4 Auditor is never ended in a particular endstate, but that even after an audit process is ended, more actions are being added to the audit trail. Also, during the audit process, there could be actions in the audit trail that are not directly related to the current audit (for instance an accountant from a different legal entity within the group accessing some documents of the audit).

The lack of a process definition in the Unit4 Auditor is displayed by the audit trailing actions, as the phases the process goes through are not displayed within the audit trail. Instead, the process steps are logged in a Task table, and mutations on that table are logged in the audit trail.

Unit4 Auditor is a program that was developed years ago, with a different mindset from current generation development. During the development, the application was being developed with the idea of a stand-alone application, that did not have to communicate with other applications. This is reflected in the implementation of the application, as it is developed as a closed system that does not provide a structured way to access its data, such as a REST API or export functionalities.

It is however possible to connect with the backend SQL Server directly. Because documentation on how to do this, such as the database structure, is not publicly available, one has to go through the database server. Within the database server, there is one master database instance, that is copied for every new client. This results in a sort of Chinese walls within Auditor, where all the information for a client is isolated from the other clients. Also, every database instance includes the system tables, making the database nondependent on the master database, ensuring that, when requested, a full export can easily be done for a client.

Within Unit4 Auditor, different template processes can be designed. Within WVDB, there is one template for auditing, that is updated every year. Since 2016, this template has only changed slightly (mainly updating descriptions of risks and how to handle them when laws or legislation has changed) and the variation between the years is neglectable compared to the freedom an auditor has to modify the process. Because audits need to be filed within one year after the end of a fiscal period, one can safely assume that all audits over 2018 are finished at the moment of writing. Because of this, all the audits from 2016-2018 (executed in 2017-2019) can be used in our dataset.

As explained in section 3.2.3, one cannot simply split the dataset at random, because all the data within the audit is correlated. The split therefore has to happen in time, where the first part of the dataset is used as training data, the second part as test data, and the third part as validation data. As the audit process is a yearly process, all audits start after the end of the fiscal year, and are, as required by the rules and legislation, finished within one year. Because of this, there are no running audits at the first of January, ensuring that if we decide to split the data on this date, there is no overlap in cases between the training and test data. The entire dataset comprises of three years. The three consecutive years are evenly split over the training (2016), test (2017) and validation (2018) sets. Note that this split is not exactly one-third each, as WVDB is slowly growing and therefore the number of cases per year increase.

To fetch the data required to build the model (as shown in Table 4), a SQL query is used (see appendix 9.3). Because the audit files and corresponding logs are stored over a lot of database instances, this query is executed on every database instance, to fetch all data. This query also filters the data so we don’t have processes before 2016 or from other processes (e.g. accounting files instead of audit files) which are also logged in Auditor.

4.2 Training the models

Before the models can be trained, the data extracted from the database need to be transformed. As discussed in section 3.3, we have some categorical variables that have a very high cardinality. An example of this is the name of the task. One could think of this variable as unrelated, but they are all part of the same process and therefore have an ordinal ordering among them. In our dataset, this is made easy by the naming of the tasks, as the name of the task is constructed of a code indicating where in the process the task is and a human interpretable name. An example of this is “E.0.1.14

Debiteuren" where E.0.1.14 indicates in what folder the task is and Debiteuren is the human interpretable part. Because of the code proceeding the name, one can keep the ordinal encoding in the categorical variable by simply sorting them alphabetically when label encoding them.

As explained in section 3.3, both models have hyperparameters that have an impact on how the model is trained. For the **random forest**, these hyperparameters provide a lot of control over the depth of the forest. The deeper the forest is, the more the model can learn from the training data and the higher the performance on the training data becomes. Growing the forest to deep often results in overfitting behaviour, reducing the performance on the test data. Hyperparameters to control the depth of the random forest are the **maximum depth, the minimum number of samples required to split, and the minimum number of samples required at a leaf**. These hyperparameters have some overlap, e.g. if the minimum number of samples required at a leaf is 50, the minimum number of samples required to split is at least twice as much, namely 100. To introduce variation into every tree that composes the forest, one can control the **number of features to consider** when looking for the best split and the **maximum number of samples** to train each tree. If the number of features to consider is set equal to the number of features available in the dataset and all the data points are used, one would eliminate the variability between the trees. Two often used options are sqrt or log2 of the number of available features. Setting the maximum number of samples to train each base tree, one can increase the amount of variability between the trees as they are trained on only part of the dataset. In Table 5, the hyperparameters and the values used for the grid search are named. In total, 560 different hyperparameter combinations are tried. Note that the values used for the grid search are partly based on the training dataset size.

Hyperparameter	Value(s)	Degrees
Minimum number of samples required at a leaf	1, 2, 5, 10, 20, 50, 100	7
Minimum number of samples required for a split	2, 5, 10, 20, 50, 100, 200, 500	8
Number of features to consider when looking for the best split	Sqrt(nr of available features), log2(nr of available features), Nr of vailable features	2
Maximum number of samples to train each base tree	100%, 95%, 90%, 80%, 70%	5

Table 5: Values for grid search of random forest

Compared to the random forest, a **neural network** has a lot of hyperparameters to control how the model looks like. For the most basic configuration, one has to set up a network with a number of layers and nodes per layer. Since we only use the neural network as a baseline model to compare the performance of the random forest against, we are not focussed on finetuning the network until we have the best performing model. For the baseline model, we use a model with four layers of 80, 80, 20, and 10 nodes respectively. Between each layer we perform a batch normalization and a dropout, to prevent overfitting. As our model only contains four small layers compared to the cardinality of some of the categorical input variables, these variables are not directly fed to the neural network but each categorical variable is first embedded in an embedding layer of half the cardinality (with a maximum of 50), which are then fed to the neural network together with the other input variables. As the predictions are binary, the prediction layer uses a sigmoid activation and the binary cross-entropy is used for the loss function during training.

As explained in section 3, the model has two purposes. The first purpose is to work as an online operational support model and the second purpose is to be interpretable and indicate key factors that impact the possibility of rework. Both purposes of the model require the model to be trained differently. When using the model as an operational support model, the number of false negatives should be minimized, as the costs of those are very large compared to the costs of false positives. When using the model to determine the key indicators for predicting rework, the number of false negatives does not have a larger cost than the number of true positives, as in this case one is not going to use the models' predictions to influence processes. Even though this model is not used for predictions in the end, it still needs to be evaluated using the test data, as we are looking for indicators that are part of the normal process, not just in a specific period.

Because both purposes of the model require a different model, two models are trained (besides the baseline neural network). These two models are trained based on the same data and the same values are used for the grid search. The selection of the optimal values of the grid search is different between the two models.

Hyperparameter	Model High Accuracy	Model Minimum False Neg
Minimum number of samples required at a leaf	2	100
Minimum number of samples required for a split	2	500
Number of features to consider when looking for the best split	sqrt(available features)	sqrt(available features)
Maximum number of samples to train each base tree	70%	95%

Table 6: Optimal hyperparameters for the Random Forest models found using a grid search

In the results of the grid search displayed in Table 6, one can clearly see that both models have very different hyperparameters.

4.3 Model results

When looking at the hyperparameters configurations for the grid search in Table 5, one can clearly see how some of the hyperparameter configurations will lead to overfitting, getting an (almost) perfect accuracy on the training data, while having limited prediction ability on the test data. An example that obviously overfits, but is included in the grid search, is the random forest that basically eliminates the randomness by considering all features when looking for the best split and splitting until the leafs cannot be split further. While this performs very good on the training data, it is overfitted and therefore does not perform well on data it hasn't seen before. Because of the overfitting, the accuracy on the training data is between 100% and 81.7% while the accuracy on the test data is between 92.9% and 81.7%.

In Table 7, the evaluation metrics of both models are displayed. From this model, one can clearly see that both models perform quite differently. This is partly due to the skewed data (only 7.8% of the tasks requires rework) which gives the model an incentive to predict more negatives than positive results. This can be seen when we compare the accuracy with the weighted accuracy. In the best performing model with respect to the accuracy, the weighted accuracy is a lot lower than the normal accuracy (64.9% against 92.9%) while in the model where we minimize the false negatives the accuracy and weighted accuracy are almost equal (81.8% against 81.7%). This clearly shows that the high accuracy model has a bias towards no rework.

Evaluation metric	Model High Accuracy	Model High F1	Model Minimum False Neg

Accuracy	92.9%	91.0%	81.8%
False Negatives	2252	1353	608
Area under ROC curve	0.87	0.89	0.89
F1	0.41	0.51	0.41
Weighted Accuracy	64.9%	76.3%	81.7%

Table 7: Evaluation metrics of the prediction models

For finding the key indicators for rework, we don’t care about how the false predictions are spread (positive or negative) but only look at the accuracy. Therefore, for this version of the model, we use the model with the highest accuracy (see Table 6 for the hyperparameters). From the confusion matrix of this model (see Table 8), one can see that this model has an accuracy of 92.9%, but has 5.3% of false negatives. When optimizing the hyperparameters for minimizing the number of false negatives, we see that the model becomes less tightly fitted, with much larger leaf nodes. As a result, the overall accuracy of the model goes down, while the amount of false negatives goes down. While the accuracy of this model is more than 10% lower, the amount of false negatives is reduced by 73%. This means that the amount of false positives is increased a lot, but a false positive is a lot more acceptable by the users. This also results in an increase in the weighted accuracy, from 64.9% to 81.7%.

Accuracy: 92.9%	Actual Rework	Actual No rework
Predicted Rework	1044 2.6%	751 1.8%
Predicted No rework	2252 5.3%	38406 90.5%

Table 8: Confusion matrix for Random Forest with the highest accuracy

Accuracy: 81.8%	Actual Rework	Actual No rework
Predicted Rework	2688 6.3%	7131 16.8%
Predicted No rework	608 1.4%	32026 75.4%

Table 9: Confusion matrix for Random Forest with the least false negatives

From the confusion matrix in Table 8, one can clearly see that the model has a bias towards predicting no rework. As the no rework class makes up 92% of both our training and our test sets, this behaviour is explainable, as just predicting no rework for all cases would already give an accuracy of 92%, which is almost the same as the model with the highest accuracy. For this, one could argue that the model is not useful. However, the models do not only return the predicted classes for every case but also return the chance that the case is in that particular class. Using this, the ROC curves shown in Figure 4 can be constructed.

While the accuracy of both models is not a lot better than predicting no rework for every task, the ROC curve shows that the model clearly outperforms random guessing or assigning the same class to every task as the class probabilities predicted by both models are good estimates of the prediction being correct.

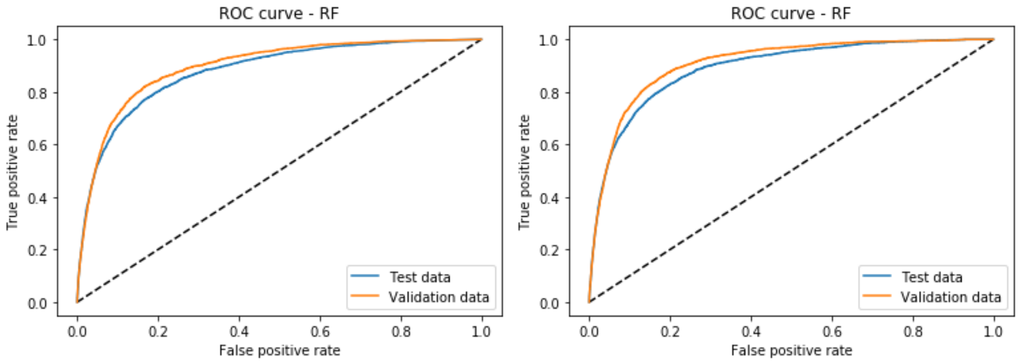


Figure 4: ROC curves for the Random Forests (Left: Max accuracy, Right: Min nr of False Negatives)

Because we only use the neural network as a baseline to compare the performance of the other models against, and there are numerous ways to structure a neural network, the hyperparameters of this network are not optimized. A basic neural network with four layers with respectively 80, 80, 20 and 10 neurons is used. Because of the nature of the data, using fully connected layers will result in overfitting the data. To prevent this and support training with respect to the validation data, dropout layers are added between the neuron layers. The model is trained by minimizing the binary cross-entropy and uses a batch size of 128. The neural network is trained using early stopping with a patience of 25 epochs, meaning that if the loss on the test data hasn't improved for 25 epochs, training is stopped and the best model until then is used. The model doesn't improve on the test data from the 5th epoch. The trained model has an accuracy of 89.0% and the confusion matrix is shown in Table 10.

Accuracy: 89.0%	Actual Rework	Actual No rework
Predicted Rework	903 2.1%	2288 5.3%
Predicted No rework	2393 5.6%	36869 86.8%

Table 10: Confusion matrix for the Neural Network

The accuracy of the trained neural network (89.0%) is in between the accuracy of the random forest models. This indicates that the models generated perform well compared to the baseline Neural Network. When we compare the confusion matrix of the neural network (Table 10) with those of the random forest, we see that the neural network has difficulties predicting the rework cases. This is also shown in the ROC curve for the neural network (Figure 5). Because of this, the neural network is less suitable to make predictions as an operational support tool as the class probabilities are less accurate. It is important however to realize that we did not optimize the hyperparameters of the neural network and that with other layers in the network the performance could improve.

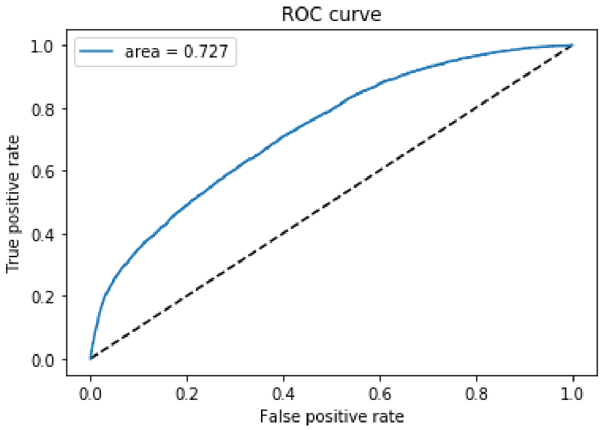


Figure 5: ROC curve for the Neural Network

5 Root cause analysis

To determine the importance of the input parameters for the random forests, one could open and analyse each tree manually. As the trained random forests exists of 100 decision trees, with well over 50 splitting nodes, one can quickly oversee things when analysing the trees by hand. It is however possible to extract the importance of each parameter from the random forest. Every node in every tree is a condition on a single feature, chosen to split the dataset into two sets with similar outcome probabilities. The measure on which the optimal split is chosen is the gini impurity. After training the trees, one can see how much each split decreases the impurity in a tree. For a random forest, the impurity decrease for each feature per tree can be averaged and this can be used as a measure of importance of the particular input parameter. In Table 11, the importance of the parameters in the random forests are shown.

Feature	**Parameter name in model**	**RF High Accuracy**	**RF Min False Negatives**
Number of attachments	Attachments	0.135916	0.263966
Waiting time until the review	TimeWorkFinishedTillReview	0.126138	0.167284
First review on	EersteReview	0.077187	0.088547
Work time	WorkTime	0.074300	0.087987
Task identifier	TF_cat	0.062641	0.046818
Duration start till review or definitive	DuratieStartTotReviewOfDefinitief	0.057478	0.045027
Work ended on	FirstWorkEndedOn	0.054163	0.032336
Audit phase progression	ProgressionTaskFolderType	0.050549	0.025628
Start on	StartOn	0.050540	0.022621
Absolute time from another review	TimeFromOtherReview	0.045579	0.030297
Task title	Title_cat	0.039774	0.008893
Audit phase	TaskFolderName_cat	0.036291	0.018884
Reviewer name	Reviewer_cat	0.036256	0.018059
Reviewer interactions	ReviewerInteractions	0.033439	0.049904
Task folder progression	ProgressionTaskFolder	0.029457	0.007055
First audit event	FirstAuditEvent_cat	0.026693	0.036409
Review type	ReviewType_cat	0.014273	0.015942
Number of involved employees	NrOfInvolvedEmployees	0.012613	0.011713
Audit phase	TaskFolderTypeName_cat	0.009441	0.006970
System audit phase	SysTaskFolderTypeName_cat	0.008465	0.009705
Reviewer edits	ReviewerEdited	0.005494	0.001836
First definitive on	EersteDefinitief	0.004441	0.000954
Number of assign times	TaskAssignedCount	0.003730	0.000249
Creation type	SysCreationTypeID_cat	0.002657	0.000522
Risk severity	Risicocount	0.002485	0.002396

Table 11: Feature importance of the Random Forest with the least false negatives

As we can see from Table 11, the importance of the parameters is quite similar in both random forests. The 7 most important parameters are the same for both models, and also the other parameters do not differ too much. Interesting to see is that the number of attachments is by far the most important parameter in the random forest minimizing the number of false negatives, with an importance of 0.264, while the importance of this parameter is almost half (0.136) in the other model.

Just knowing the importance of the parameters is however not enough to conclude things about those parameters. One should also take the direction (how it is correlated with the model outcome) of the parameter into account. Note that with a random forest, the parameters are not linear (as for example with a linear regression model) and therefore cannot be expressed in a single number. To analyse the effect of a parameter, it is necessary to isolate the parameter and look at the effect on the outcome. In the appendix (chapter 9.1), the effect of each parameter is shown for the random forest maximizing the accuracy. On the x-axis, the values of the parameter are shown and on the y-axis the probability that the task requires rework. Five lines are plotted, showing the actual probabilities for the training, test, and validation data and the predicted probabilities on the test and validation data. Also, a histogram of the parameter is shown, indicating the distribution of the variables.

5.1 Root causes

All the parameters were discussed with experienced auditors at WVDB Adviseurs Accountants. This included discussing if the behaviour of the parameter was as expected, or otherwise could be explained and if the parameter could be influenced by the auditors during the process. The next paragraphs discuss the most influential parameters for both models.

When we look at the most important parameter, the **number of attachments**, it is immediately clear why this parameter is important in both models. The probability for rework is very low if the task does not have attachments, and it is increased a lot if there is at least one attachment. Because of the low probability for rework when there are no attachments, this is a very good indicator. Process wise, this can be explained as there are a lot of small, fairly simple tasks in the audit process (especially during the preparation and planning phases). These tasks often don't have attachments, while they rarely get reopened.

The next parameter, the **time between the work is finished and the review**, shows a distinct bump in the predictions at the lower end of the parameter, before it slowly reduces the rework probability when the time increases. This is because a lot of time, the review happens fairly fast after the last work on a task is done. This is either because the reviewer makes some changes before reviewing or because there is a close involvement of the reviewer, meaning he reviews the task as soon as the auditor indicates that the task is finished. The prediction curves show that they mostly only use this bump of short times between the last activity and the review for the prediction.

When we look at the **time between the end of the auditable period and the review**, it shows a small decline over time. This could be because the more time passes between the end of the audit period and when an audit is done, more information is available. If an audit is done soon after the period is closed, there is a larger chance that more information will come up that was not available at the time of the audit, requiring the task to be reopened. Also, the more time between the end of the auditable period and the review, the smaller the estimation component of the audit. For instance, if there are a large number of (dubious) debtors at the end of the auditable period, the auditor has to judge the risk this imposes. If the audit was performed later, there is a chance that these debtors already paid which mitigates the risk, making the analysis a lot easier and less prone to errors and therefore rework. Furthermore, some audits (especially early on in the year) have a strict deadline which puts the audit team under pressure. In these cases, the reviewer is reviewing tight, as soon as the tasks are finished, and this can increase the chance on rework.

When looking at the **worktime spend on a task**, one sees that the distribution of this parameter looks like a gamma distribution, which can be expected of a duration. The chance of reopening follows a particular shape. If the time spend is very short or almost neglectable, the chance of reopening is very small. After this, there is a negative correlation between the time spend on a task and the chance of reopening the task after the review. The low chance of reopening tasks that are finished really fast can be explained when taking a look at the work steps. There are a lot of steps, mostly in the preparation and planning phase, that are very small and are rarely changed. This includes steps as providing background information for each of the team members and answering simple questions to determine the scope of the audit.

The **task** parameter shows a more complex shape than other parameters. When looking at the histogram, there are a couple of bumps. This is because there are a lot of standard tasks that occur in every process (especially at the preparation, planning, and closing phases) while there are not much custom added tasks in these steps. In Table 12, the phases and their associated task parameter intervals are shown. One can clearly see these phases in the histogram for the task parameter, as the preparation, planning, and closure phases start with standard tasks and end with custom tasks or tasks that only show depending on the answers in the standard tasks.

The interim is interesting, as the chance of rework is very high. This is because the tasks in the interim are almost all custom tasks, added by the audit team. These tasks are a lot more complex and not as standardized as the tasks already part of the audit template. Therefore, the chance of rework is a lot higher. The same thing can be seen for the custom tasks at the end of year phase.

The task parameter is also the only parameter to show a trend over the years. For the interim and end of year phases, there is a significant increase in the probability that a task is reopened between the datasets. As the datasets are from consecutive years, this could signal a trend.

Phase	Start	End
Preparation	0	411
Planning	411	1115
Interim	1116	1359
End of Year	1360	2202
Closure	2203	2439

Table 12: Distribution of phases over task parameter

The **duration between the start to review** shows a negative correlation between the parameter and the chance of rework. Interesting here is the validation data. In the training and test data, there is no significant increase in rework chance around 0.6*10^7, but the validation data has an increase here. The predictions for this dataset also show the increase, while it couldn't have learned this from the training set. This means that it has made these predictions based on other parameters that have changed.

Both decision tree models show similar curves for the predicted and actual rework probability (see chapter 9.1 and 9.2) when looking at a single parameter. This also holds for the parameters that the models do not deem important for the prediction (see Table 11). This means that the models are able to make good predictions based on other parameters, that also result in predictions that fit the non-important parameters.

5.2 Conclusion root cause analysis

The insights delivered by the models are in line with the expectations of experienced auditors, and all effects of the parameters can be explained by reasoning. This proves that the model does function in line with the expectations that the auditors have of the audit process. It is however noted by them that the parameters can be traced back to two root causes. The **more complex and/or comprehensive** a task is, the higher the rework chance. This can also be seen in the parameters. For example, complex tasks have generally more attachments and take longer to be completed. It also makes sense that more complex/large tasks have a higher rework probability, as these tasks have more room for mistakes/oversight. The **later the review is**, the lower the rework chance. This is for example seen in the time from work finished to review or the duration from the start to finish. It also makes sense that postponing the review decreases the likelihood of rework, as the auditors are still able to edit the task without it being counted as a review. Based on the parameters and rework probability, one could argue that reviewers should review as late as possible. This would however not increase efficiency as the model suggests, as with tight reviews mistakes can be spotted early on. Because of this, one cannot simply take the most important parameters by the model and change the processes to minimize the rework probability accordingly.

The random forest is however very useful as an operational support tool. Because the predictions line up with the expectations of the auditors, they trust the predictions of the model. Even though the amount of false negatives is low (about 7.9% of the negatives is a false negative), auditors cannot blindly follow the predictions. Because the model also outputs a rework probability (and not only the binary classification) this can be presented to the auditors after reviewing a case. As the ROC curves in Figure 4 indicate, the extreme (either very low or very high) probability estimations by the model provide accurate results. Therefore, the probability estimations can be used by the reviewer as indications if additional review work is needed.

6 Conclusion and recommendations

The parameters all behave as expected by experienced auditors, which shows that the model is explainable. Furthermore, because they are able to understand how the model makes its predictions, they have confidence in the models' predictions. They think that it would be useful during the audit process to have the model running as an operational support tool if it could give them the chance of rework directly after reviewing including the reasoning behind the rework probability.

Besides using the model as an operational support tool, it is possible to find indications of rework using a random forest model. However, these indicators cannot directly be used to change the process. For instance, a tight review (a review where the reviewer reviews a task soon after it is finished) is an indication that the task is more likely to be reopened. Postponing reviews would however not result in a better executed audit as detecting rework in an early stage is beneficial for the audit. The model does expose a bigger problem within the audit process that needs to be addressed.

As stated in chapter 5.2, larger/more complex tasks have a higher rework probability. For some of the most comprehensive tasks in large audits, it is almost certain that rework is required. It is however possible to change the standard work papers to split the task into multiple subtasks. When auditing a corporation that exists of multiple legal entities, the audit team has to do the same activity for each entity, but this is not correlated. In the current template, this is grouped in one task. As a result, the task cannot be reviewed until the task is performed for every entity and when something needs to be changed for one entity, the entire task needs to be reworked. It would be recommended to split tasks that are executed per entity, as this would allow the audit team to have smaller deliverables and allows the reviewer to make finer-grained reviews.

Besides changing the work papers template, the auditors are recommended to change the way the information system (Unit4 Auditor) is used. The system has a lot of options currently used by only a part of the auditors. When observing the log, it becomes evident that every external accountant uses the system in a different way to guide their audit teams. Simple examples are that some external accountants assign the tasks to particular auditors in the system, but others do this outside the system. Streamlining the usage of the system into a best practice used by all external accountant and therefore audit teams would not only increase the usability of the log for process mining but would also increase the performance of teams currently using suboptimal ways. Part of this best practice should be tight reviews, meaning that reviewers should review tasks as soon as the auditor marks it finished. This gives the audit team a better understanding of how much of the audit is finished and detects rework in an early stage, minimizing the effort to redo it.

7 Future work

While this research has shown that random forests models can be used as an operational support model within WVDB, this is only tested on one dataset. Even though the audit process is controlled by national and global governing bodies, auditing firms are allowed to give their own interpretation to this process. Therefore, however likely, it is not proven that this method will work on all datasets from audit processes. As explained in chapter 2.1, audit data is very sensitive and a lot of auditing firms are hesitant to allow access to this data, making it hard to get additional datasets.

Future work can be done to make predictions at different moments. The created models in this thesis make their prediction when a task is reviewed for the first time. It can be discussed that making predictions at other points during the audit (Eg. before the task is finished, at the end of a phase or even live) would result in an even more useful operational support tool for auditors.

A downside of using predicting models is that you need to split the available historical data into multiple datasets to determine the optimal hyperparameters and be able to say something about the model performance. In this study, three consecutive years were used as training, testing and validation sets. As a result, the final model is trained on data that is already a few years old. If the process were to change significantly, the data used to train the model is probably not representative of the actual process. To overcome this, one could look into techniques to use more recent data as training data.

In this study, a total of 25 parameters (see Table 4) were used based on existing literature and interviews with auditors. In future work, other parameters could be used. The parameters with limited importance to the predictions can probably be removed from the prediction model without losing a lot of predicting power. Also, one could add new, not yet tested, parameters to investigate predictive strength or improve the model.

8 References

Aghaei Chadegani, A. (2013). Review of Studies on Audit Quality. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2227359

Aobdia, D. (2015). The Validity of Publicly Available Measures of Audit Quality: Evidence from the PCAOB Inspection Data. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2629305

Breuker, D., Delfmann, P., & Becker, J. (2016). Comprehensible Predictive Models for Business Processes Mittelstand 4.0 Kompetenzzentrum Lingen View project CrowdStrom View project. *Mis Quarterly*, *40*(4), 1009–1034. https://doi.org/10.25300/MISQ/2016/40.4.10

Ceci, M., Lanotte, P. F., Fumarola, F., Cavallo, D. pietro, & Malerba, D. (2014). Completion Time and Next Activity Prediction of Processes Using Sequential Pattern Mining. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8777*, 49–61. https://doi.org/10.1007/978-3-319-11812-3_5

DeAngelo, L. E. (1981). Auditor size and audit quality. *Journal of Accounting and Economics*, *3*(3), 183–199. https://doi.org/10.1016/0165-4101(81)90002-1

Dowling, C., & Leech, S. a. (2014). A big 4 firm's use of information technology to control the audit process: How an audit support system is changing auditor behavior. *Contemporary Accounting Research*, *31*(1), 230–252. https://doi.org/10.1111/1911-3846.12010

Evermann, J., Rehse, J. R., & Fettke, P. (2017). Predicting process behaviour using deep learning. *Decision Support Systems*, *100*, 129–140. https://doi.org/10.1016/j.dss.2017.04.003

Geiger, M. A. (1994). Investor Views of Audit Assurance: Recent Evidence of the Expectation Gap. *Journal of Accountancy*, 60–66. https://scholarship.richmond.edu/accounting-faculty-publications/18

King, N., & Magnusson, A. (2003). *Continuous audit process control objectives* (Patent No. US8005709B2). https://patents.google.com/patent/US8005709/en?oq=US8005709

Kinney, W. R. (2005). Twenty-Five Years of Audit Deregulation and Re-Regulation: What Does it Mean for 2005 and Beyond? *AUDITING: A Journal of Practice & Theory*, *24*(s-1), 89–109. https://doi.org/10.2308/aud.2005.24.s-1.89

O'Regan, P. (2010). Regulation, the public interest and the establishment of an accounting supervisory body. *Journal of Management and Governance*, *14*(4), 297–312. https://doi.org/10.1007/s10997-009-9102-0

Palmrose, Z.-V. (1988). An Analysis of Auditor Litigation and Audit Service Quality. *The Accounting Review*, *63*(1), 55–73. https://doi.org/10.2307/247679

Sutton, S. G. (1993). Toward an Understanding of the Factors Affecting the Quality of the Audit Process. *Decision Sciences*, *24*(1), 88–105. https://doi.org/10.1111/j.1540-5915.1993.tb00464.x

Unuvar, M., Lakshmanan, G. T., & Doganata, Y. N. (2016). Leveraging path information to generate predictions for parallel business processes. *Knowledge and Information Systems*, *47*(2), 433–461. https://doi.org/10.1007/s10115-015-0842-7

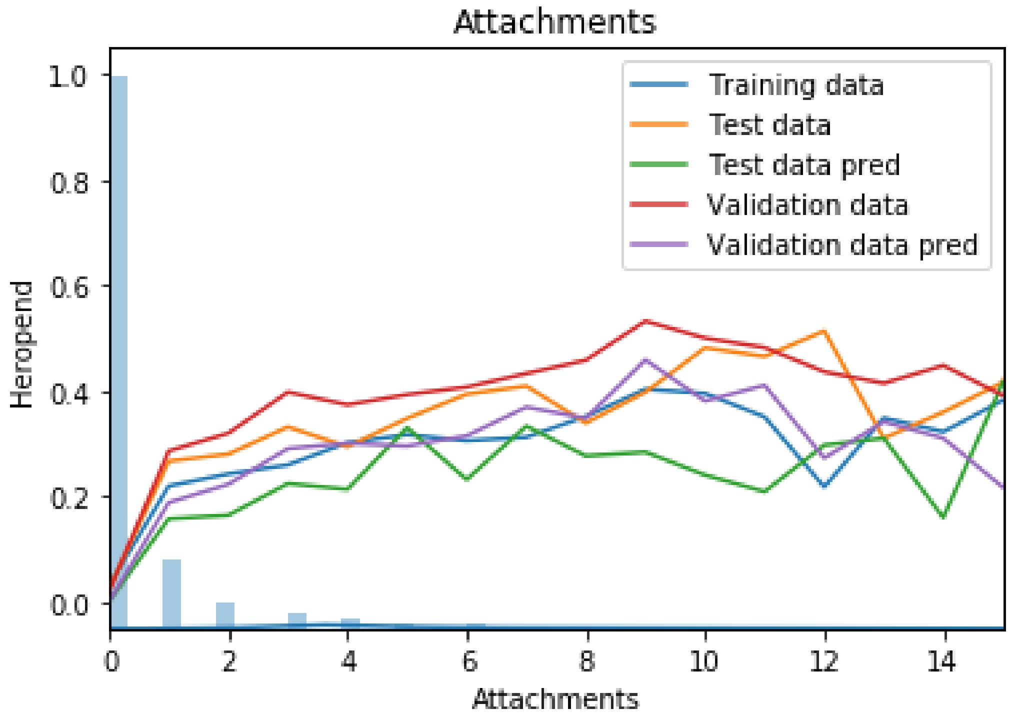
van der Aalst, W., Adriansyah, A., de Medeiros, A. K. A., Arcieri, F., Baier, T., Blickle, T., Bose, J. C., van den Brand, P., Brandtjen, R., Buijs, J., Burattin, A., Carmona, J., Castellanos, M., Claes, J., Cook, J., Costantini, N., Curbera, F., Damiani, E., de Leoni, M., … Wynn, M. (2012). Process mining manifesto. *Lecture Notes in Business Information Processing*, *99 LNBIP*(PART 1), 169–194. https://doi.org/10.1007/978-3-642-28108-2_19

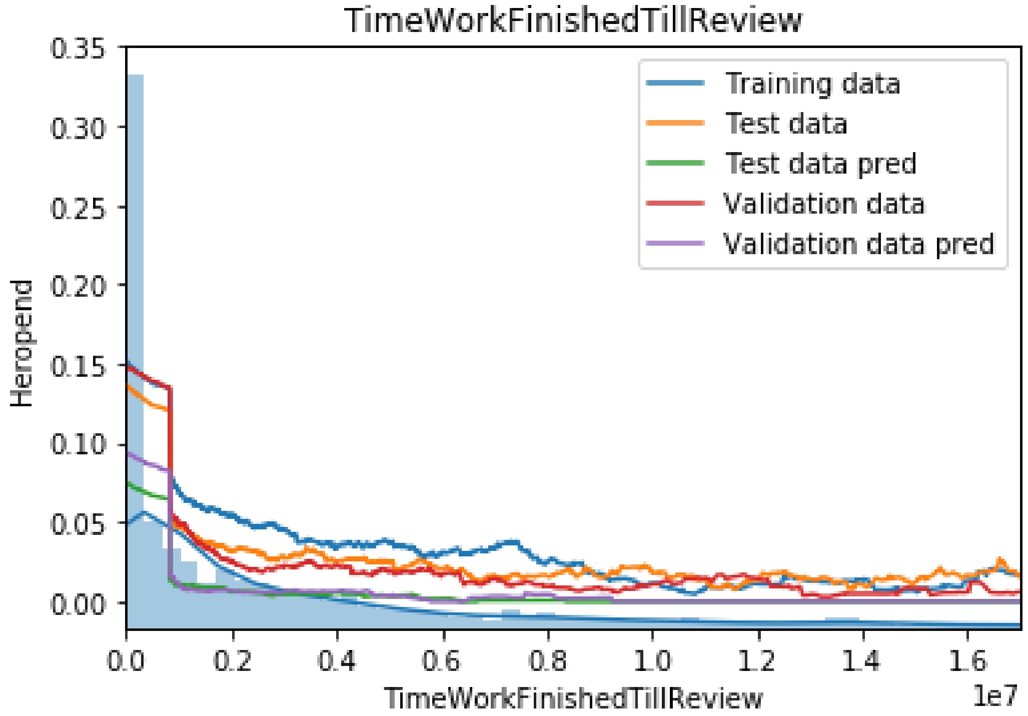
van der Aalst, W. M. P., Pesic, M., & Song, M. (2010). Beyond process mining: From the past to present and future. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *6051 LNCS*, 38–52. https://doi.org/10.1007/978-3-642-13094-6_5

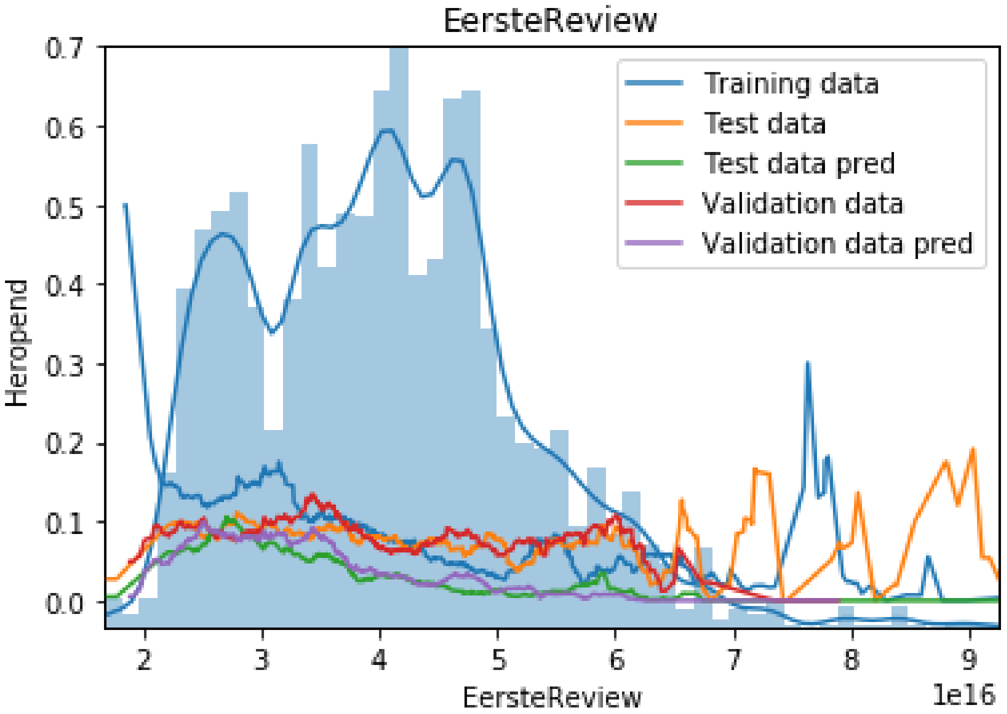
Verenich, I., Dumas, M., la Rosa, M., Maggi, F. M., & Teinemaa, I. (2019). Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring. *ACM Transactions on Intelligent Systems and Technology*, *10*(4), 1–34. https://doi.org/10.1145/3331449

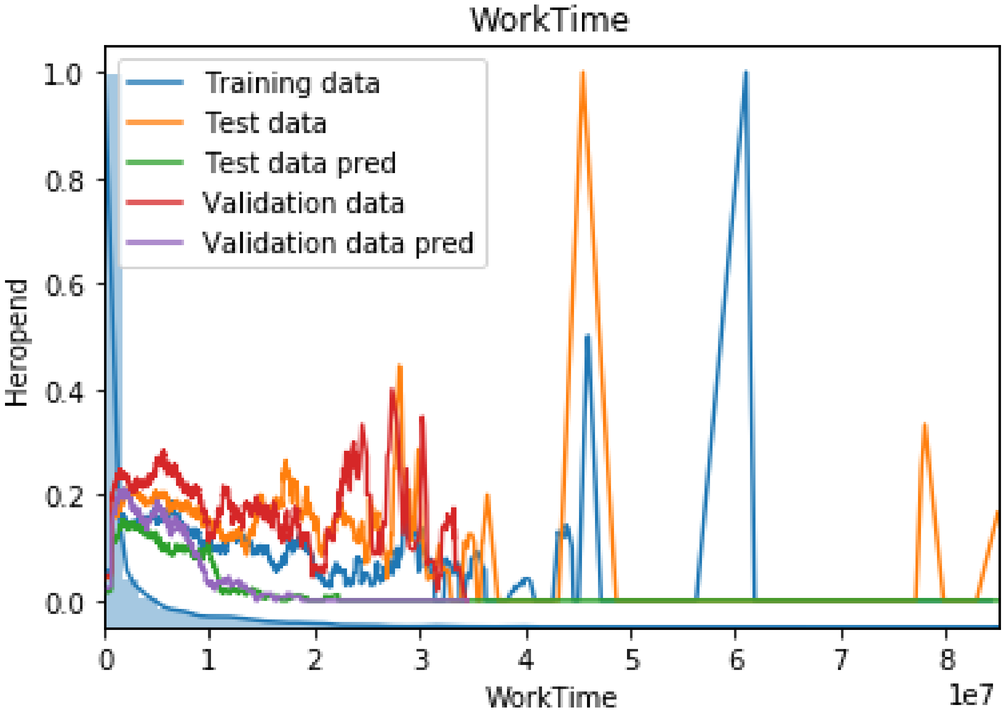
9 Appendix

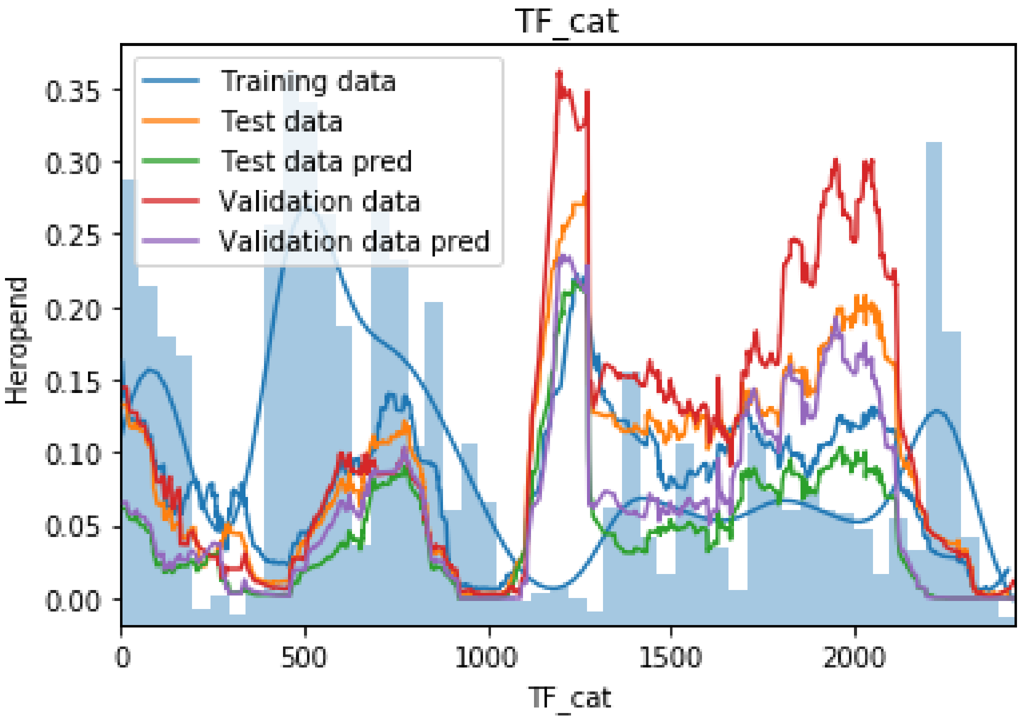
9.1 Predictions per parameter for the random forest maximizing accuracy

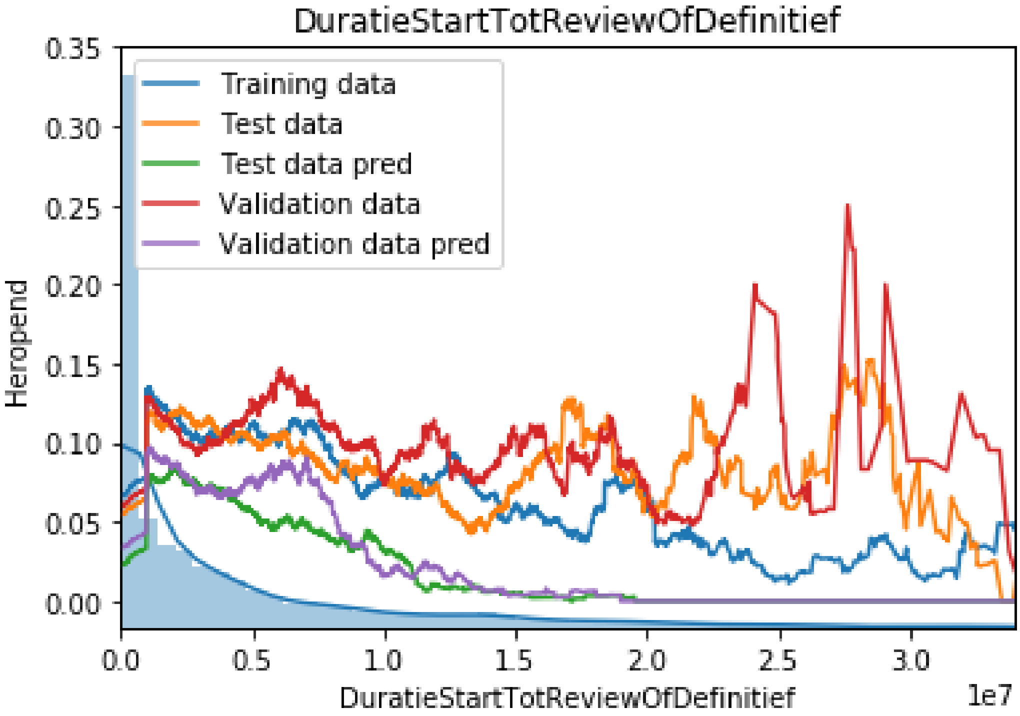


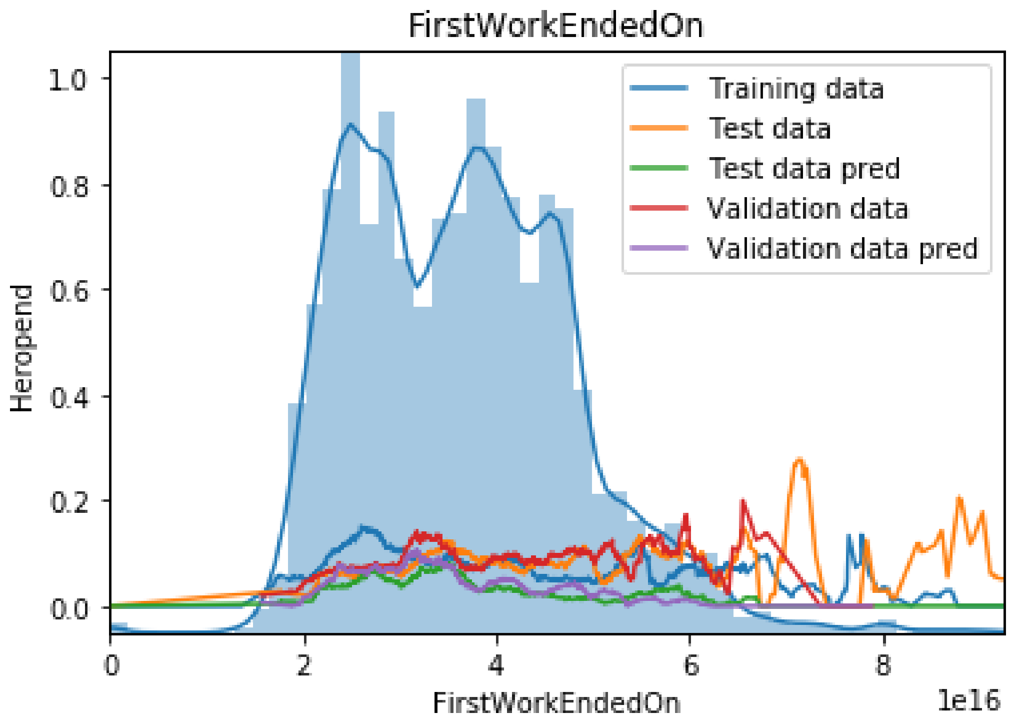


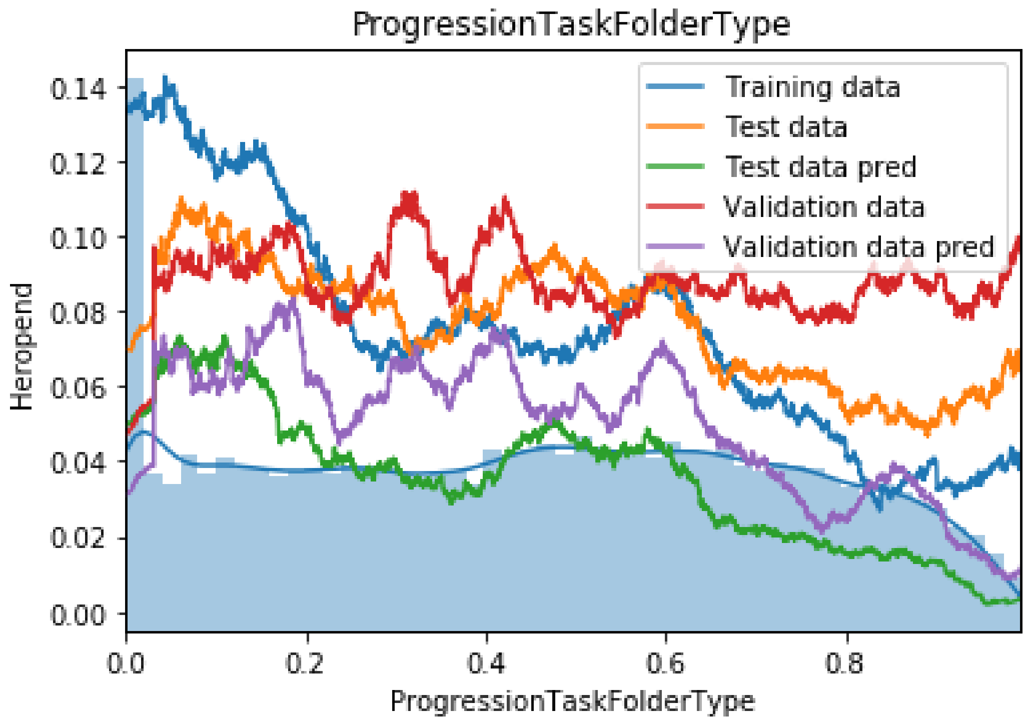

EersteReview
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
2
3
4
5
6
7
8
9
1e16
EersteReview

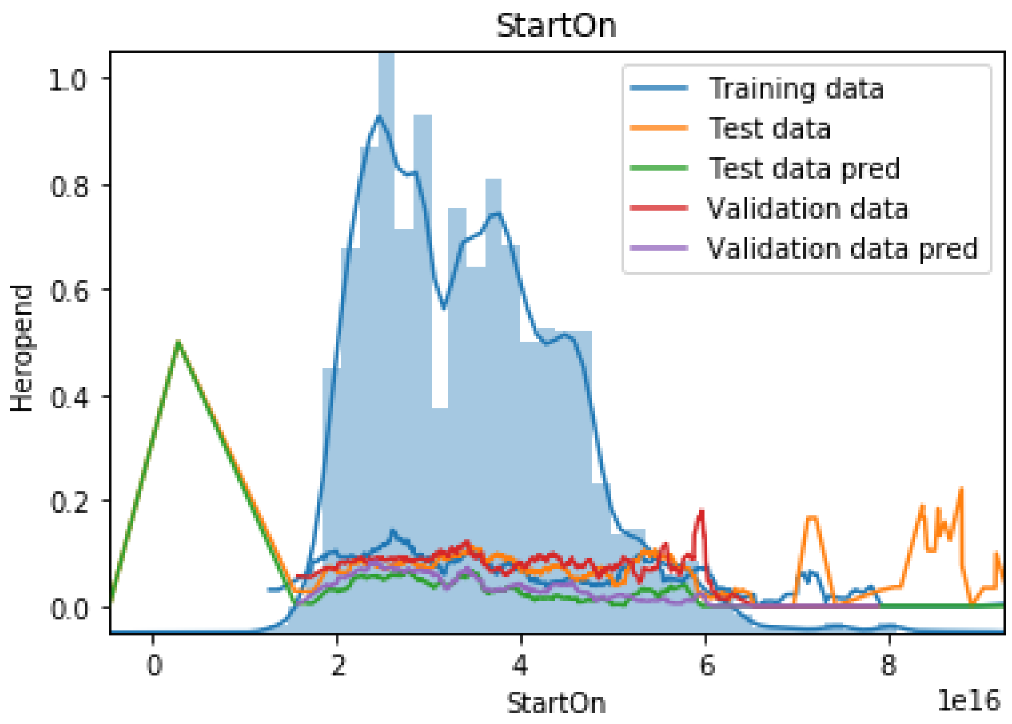

WorkTime
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
1.0
0.8
0.6
0.4
0.2
0.0
0
1
2
3
4
5
6
7
8
1e7
WorkTime

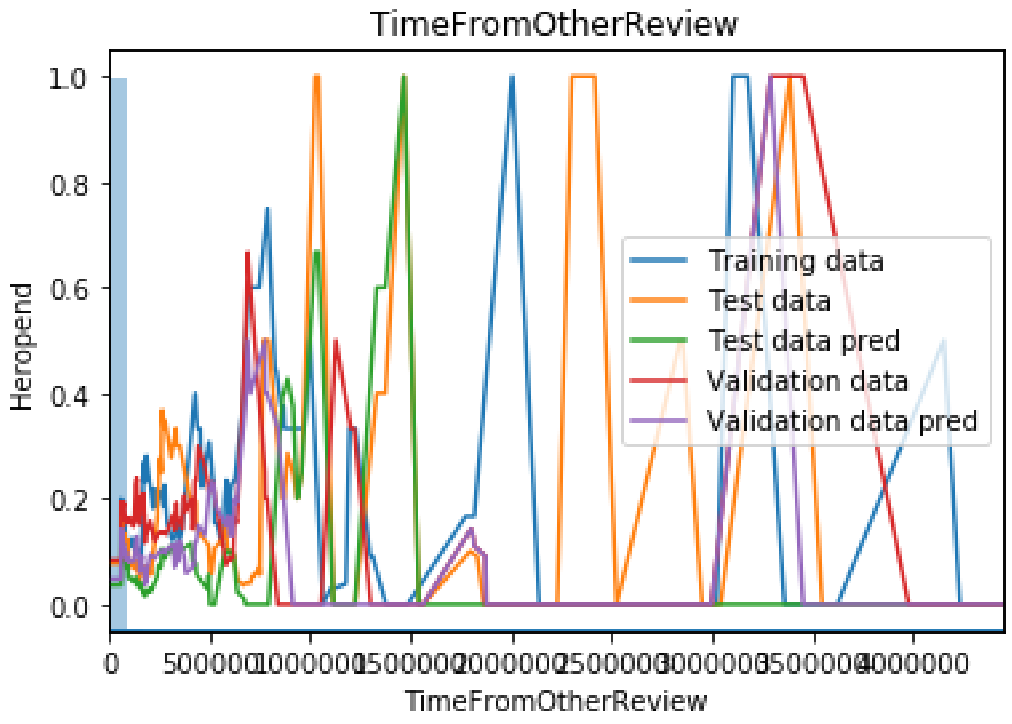

TF_cat
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
500
1000
1500
2000
TF_cat

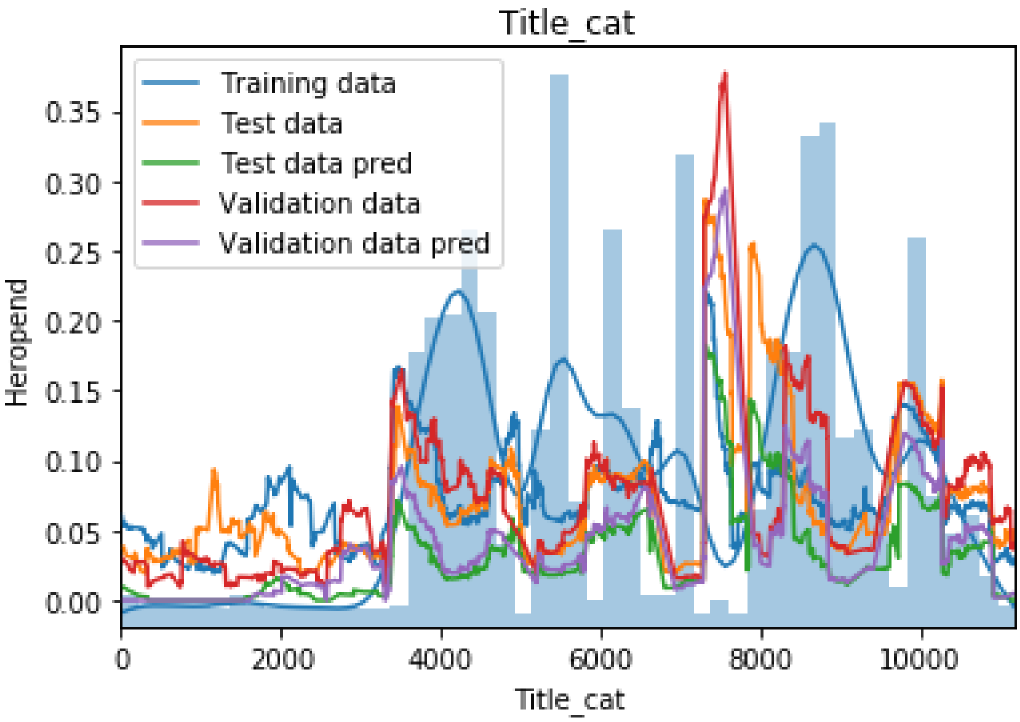

DuratieStartTotReviewOfDefinitief
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0.0
0.5
1.0
1.5
2.0
2.5
3.0
1e7
DuratieStartTotReviewOfDefinitief

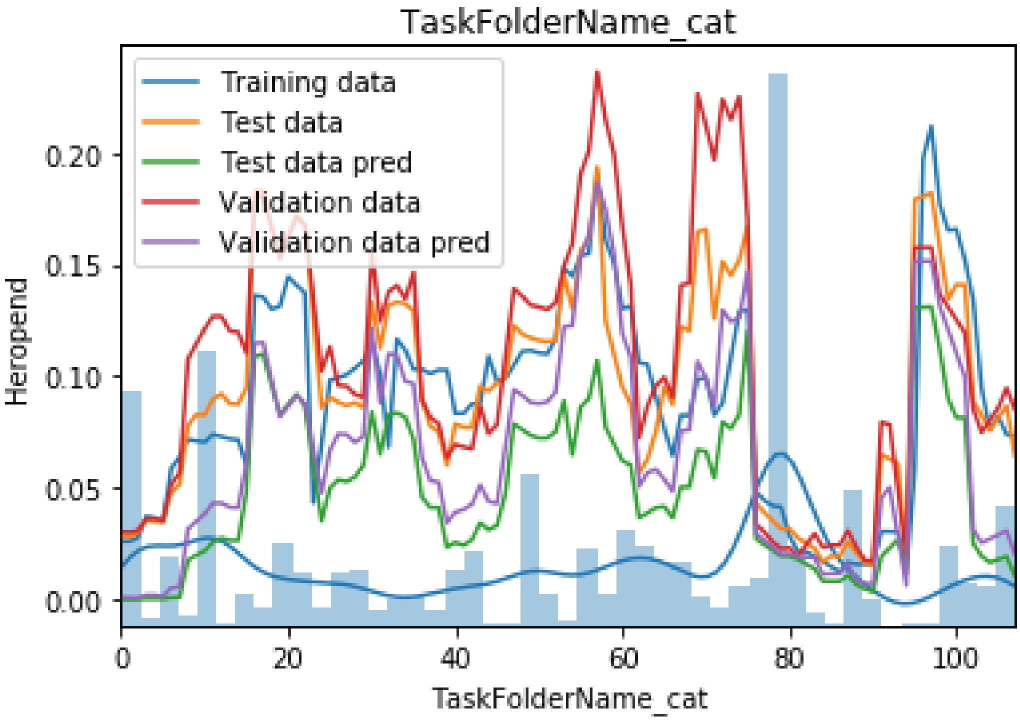

FirstWorkEndedOn
Heropend
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
FirstWorkEndedOn
1e16
Training data
Test data
Test data pred
Validation data
Validation data pred

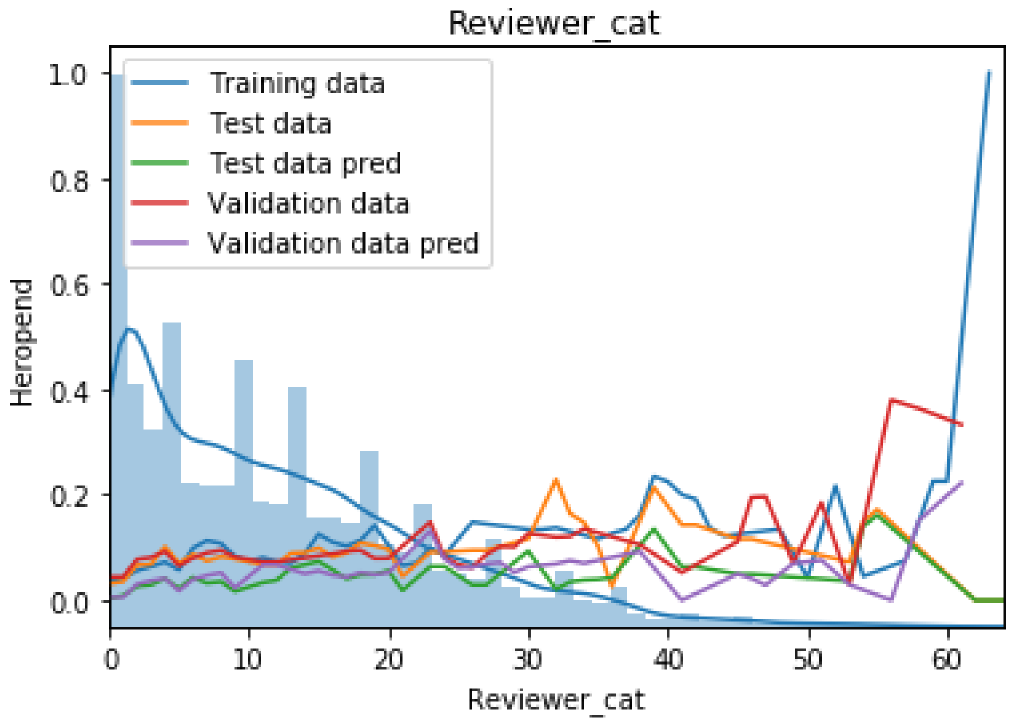

ProgressionTaskFolderType
Heropend
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0.0
0.2
0.4
0.6
0.8
ProgressionTaskFolderType
Training data
Test data
Test data pred
Validation data
Validation data pred

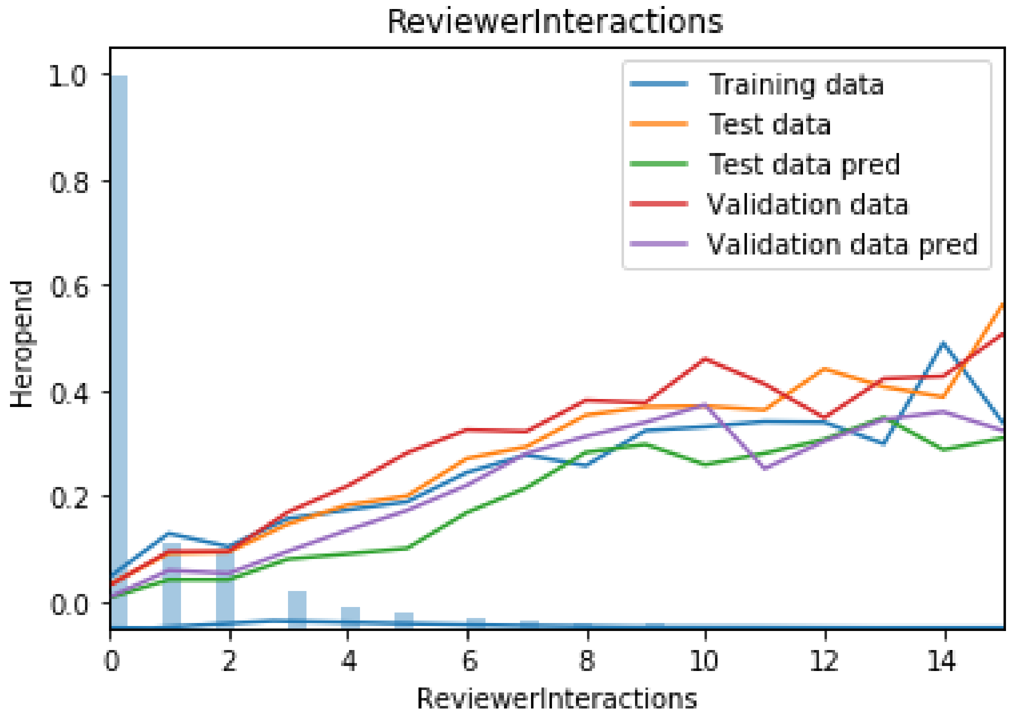

StartOn
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
1e16
StartOn

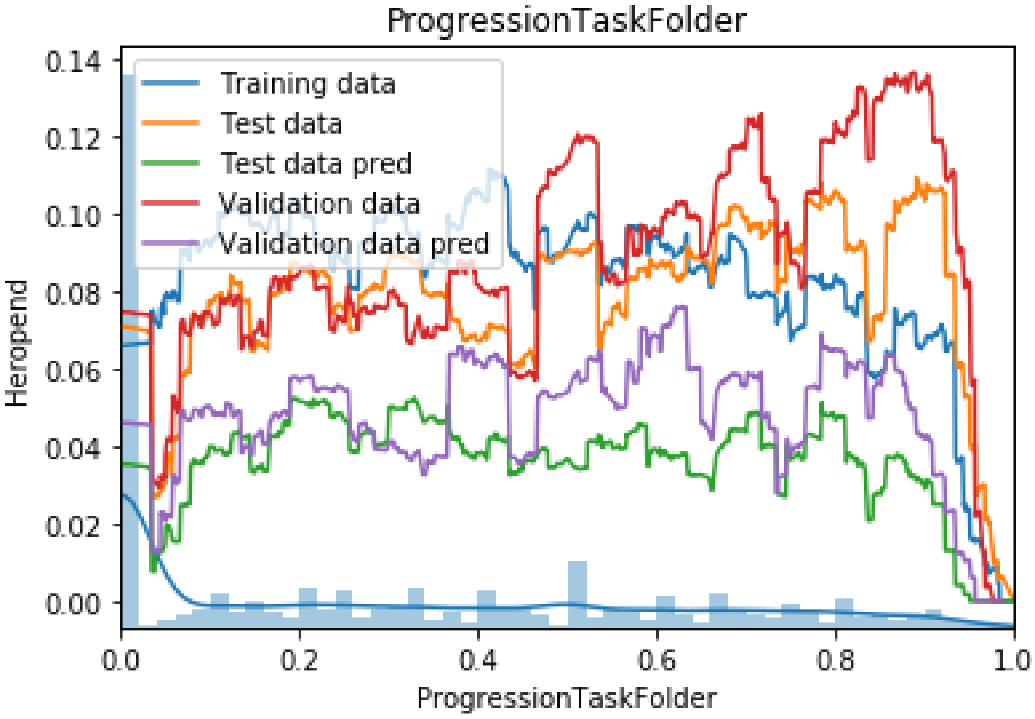

TimeFromOtherReview
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
1.0
0.8
0.6
0.4
0.2
0.0
0
500000
1000000
1500000
2000000
2500000
3000000
3500000
4000000
TimeFromOtherReview

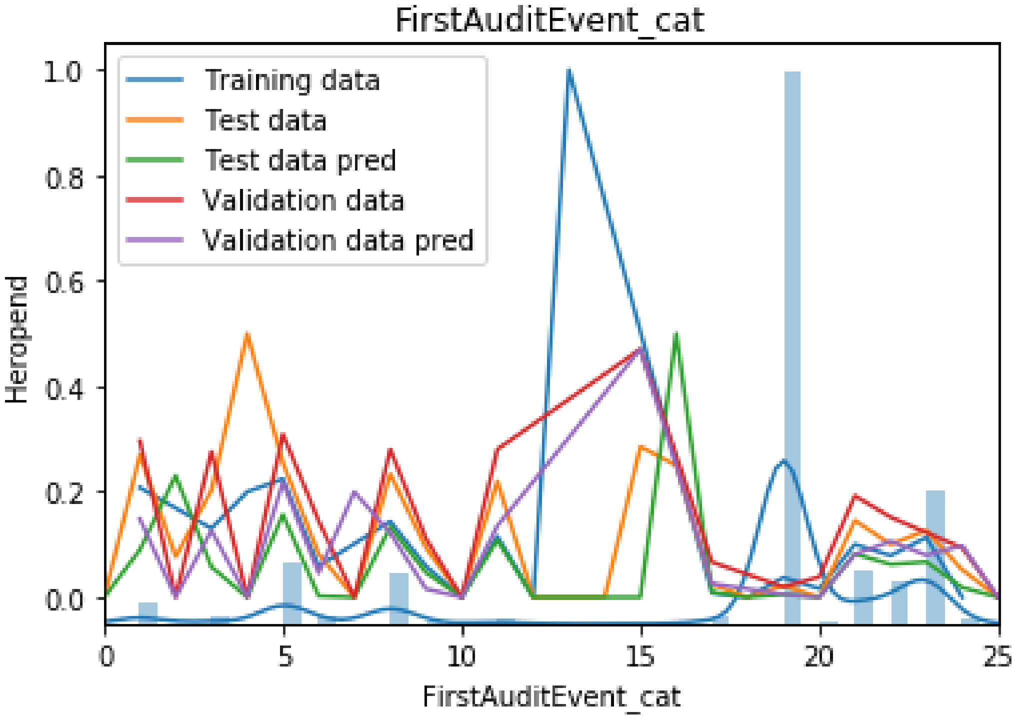

Title_cat
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0
2000
4000
6000
8000
10000
Title_cat

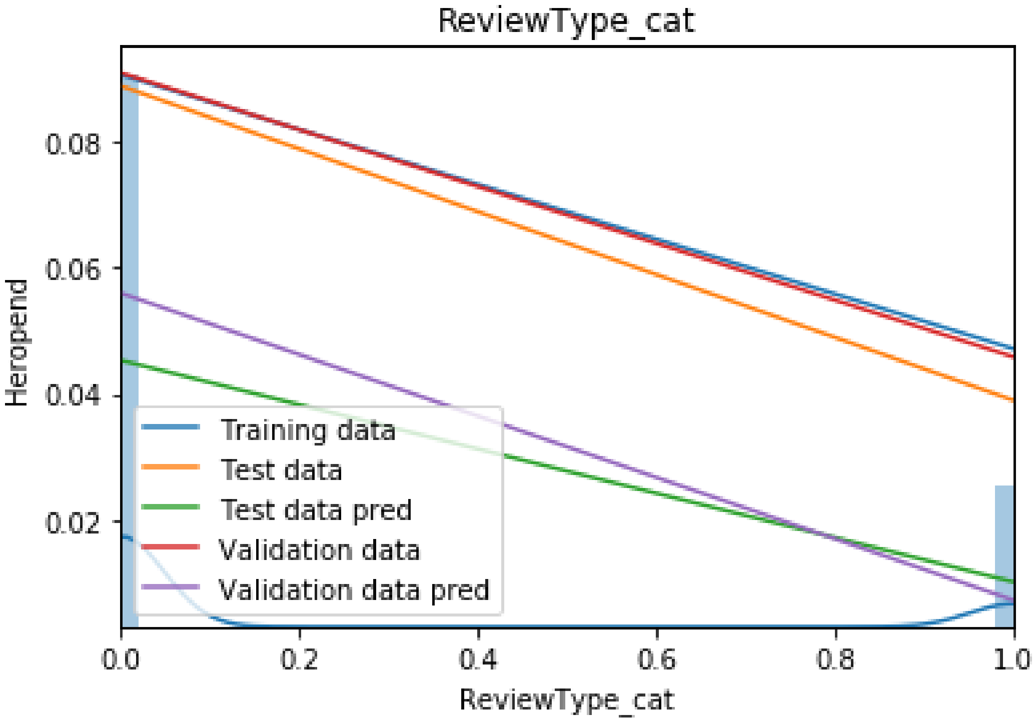

TaskFolderName_cat
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
0.00
0.05
0.10
0.15
0.20
0
20
40
60
80
100
TaskFolderName_cat

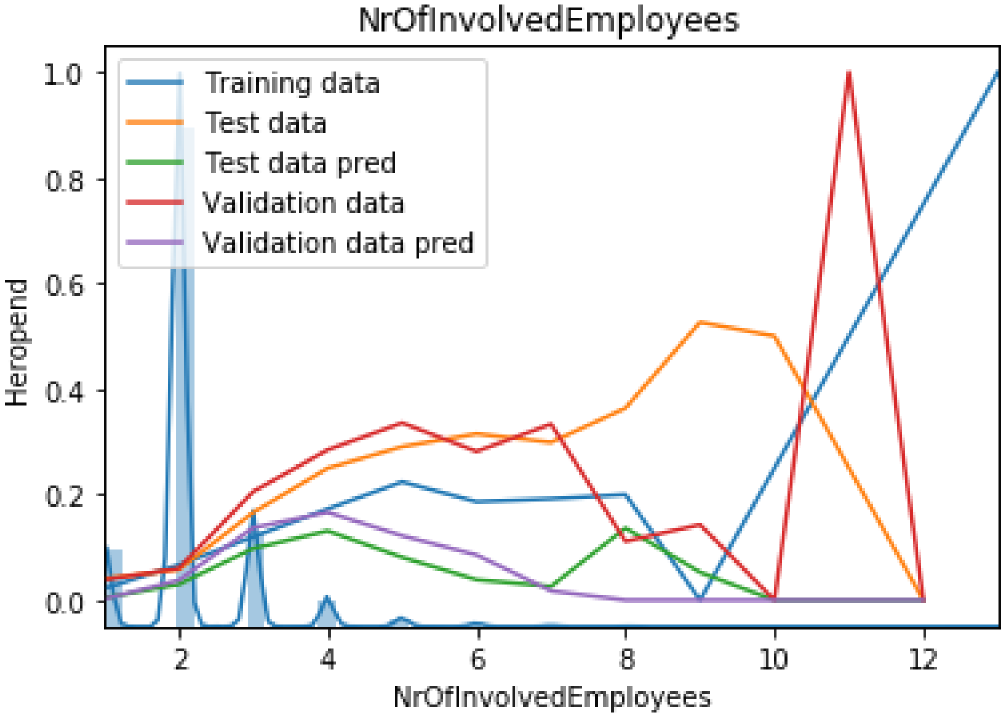

Reviewer_cat
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
50
60
Reviewer_cat

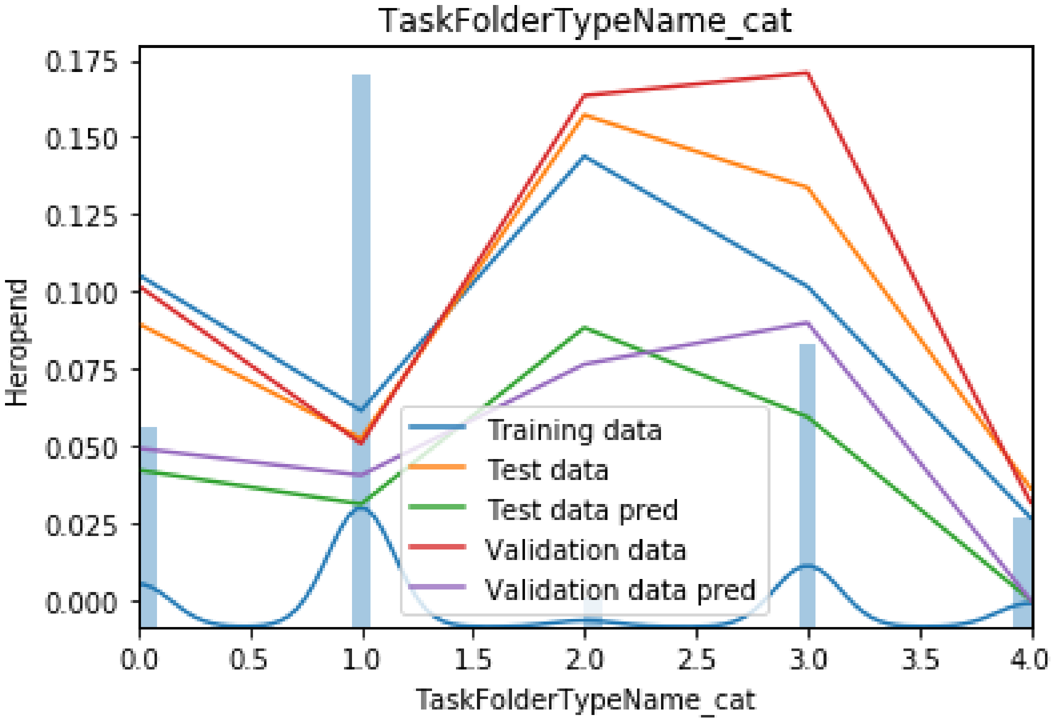

ReviewerInteractions
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10
12
14
ReviewerInteractions

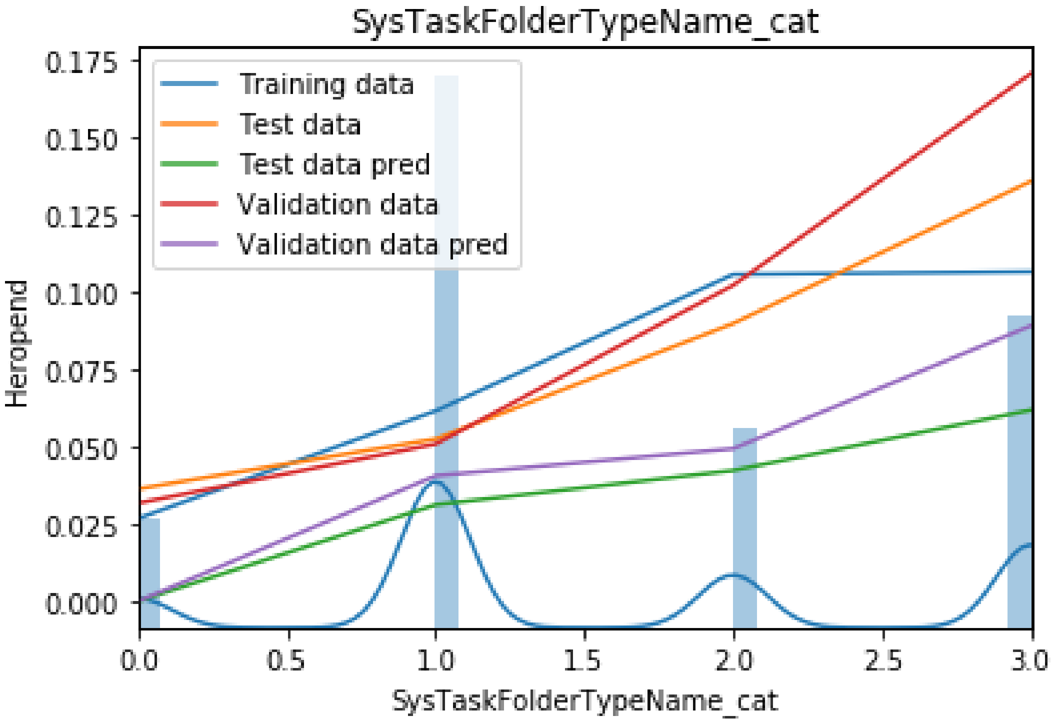

ProgressionTaskFolder
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0.0
0.2
0.4
0.6
0.8
1.0
ProgressionTaskFolder

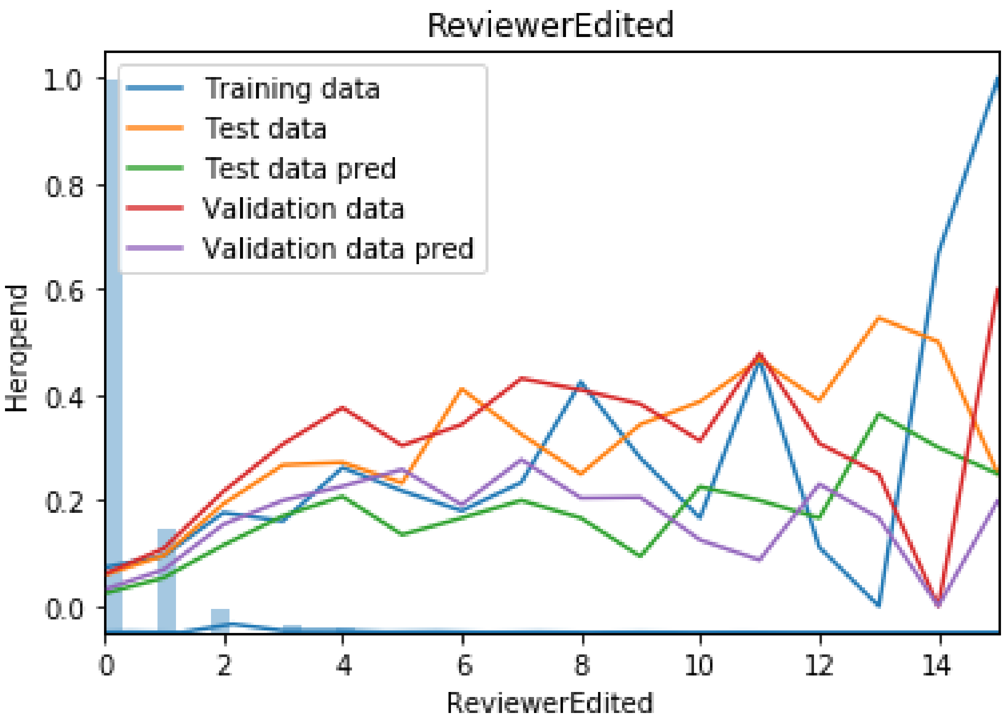

FirstAuditEvent_cat
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25
FirstAuditEvent_cat

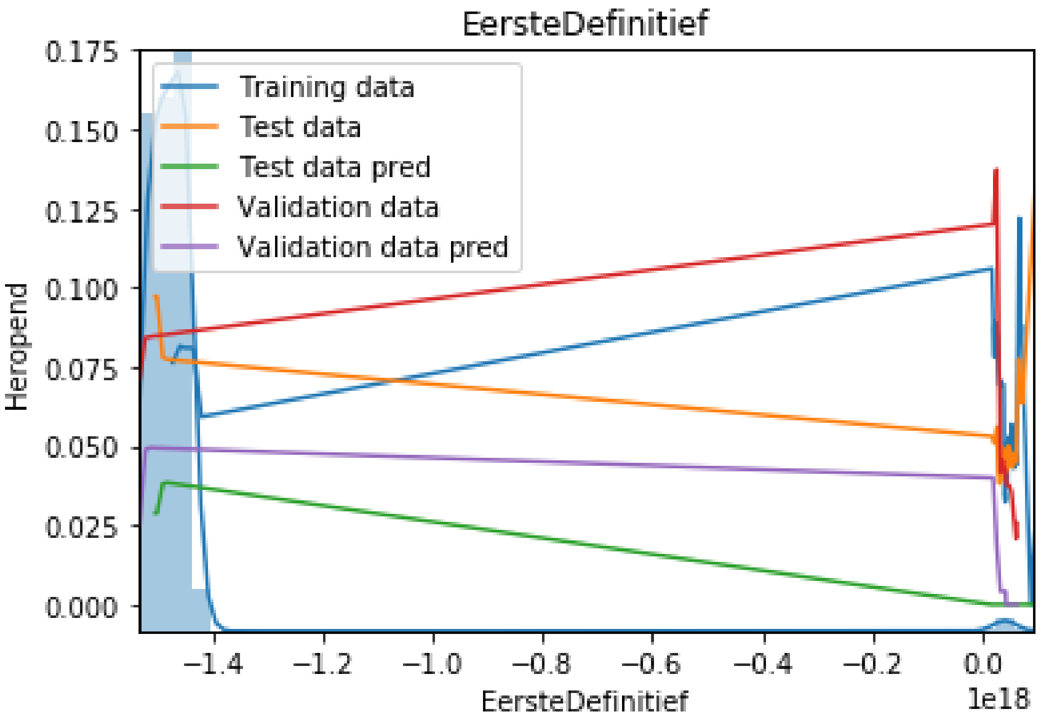

ReviewType_cat
Heropend
0.08
0.06
0.04
0.02
0.0
0.2
0.4
0.6
0.8
1.0
ReviewType_cat
Training data
Test data
Test data pred
Validation data
Validation data pred

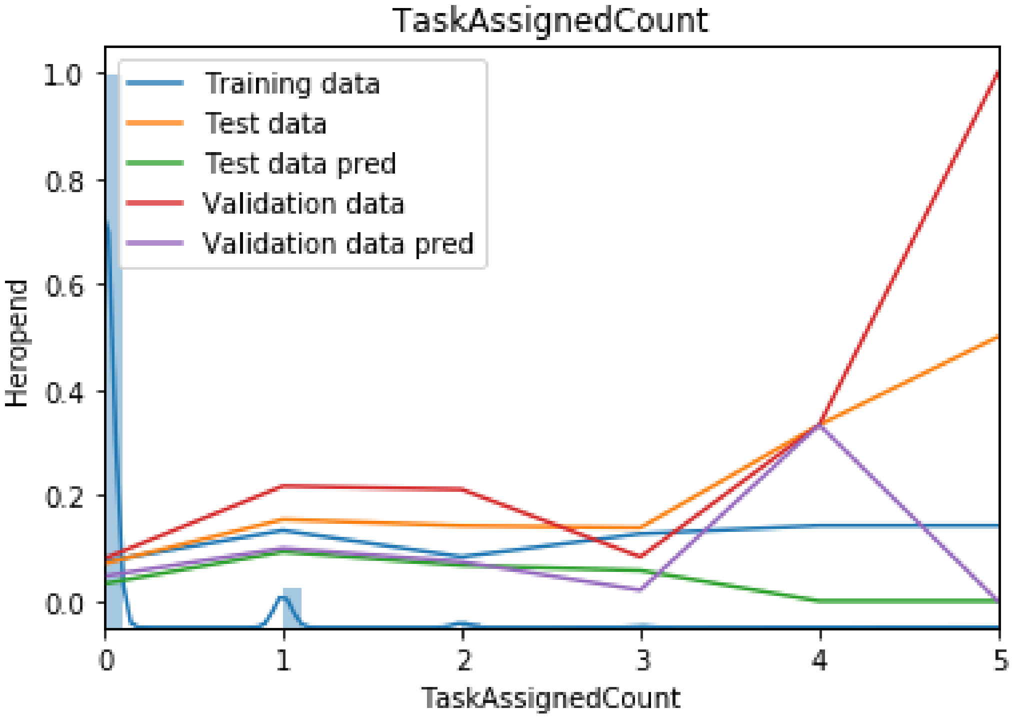

NrOfInvolvedEmployees
Heropend
1.0
0.8
0.6
0.4
0.2
0.0
2
4
6
8
10
12
NrOfInvolvedEmployees
Training data
Test data
Test data pred
Validation data
Validation data pred

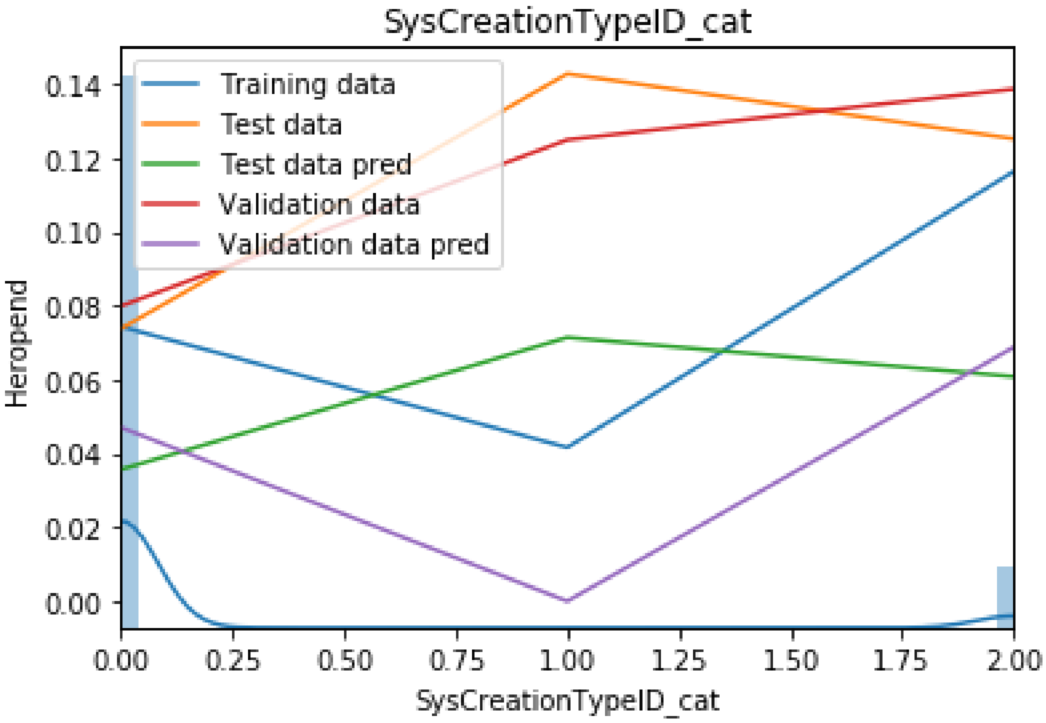

TaskFolderTypeName_cat
Heropend
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
TaskFolderTypeName_cat
Training data
Test data
Test data pred
Validation data
Validation data pred

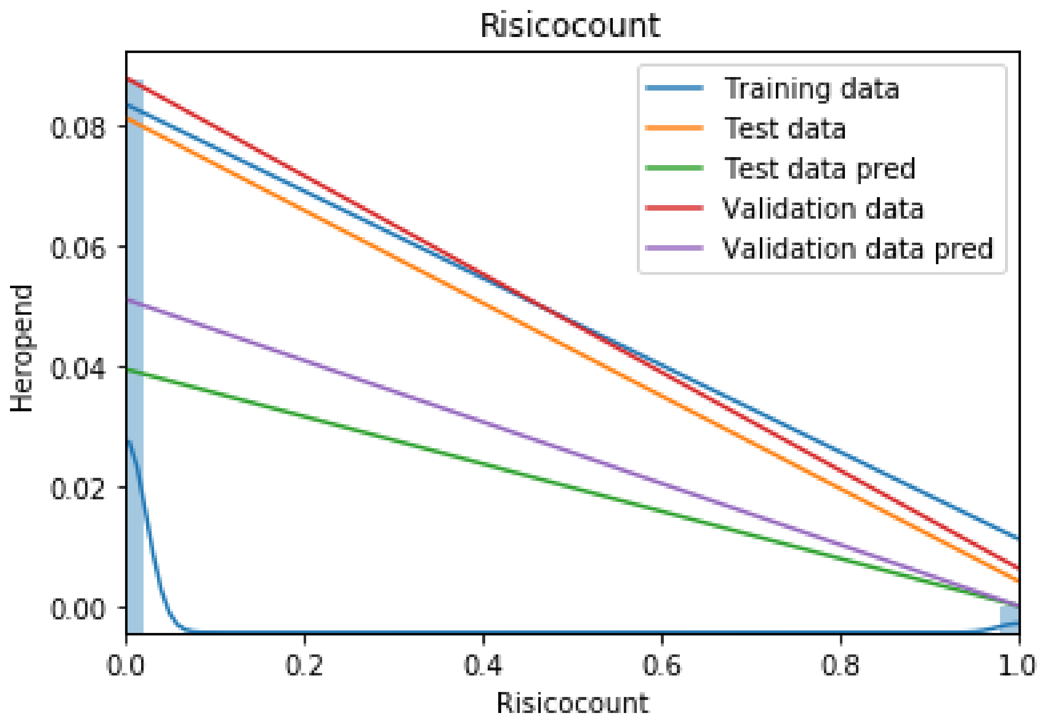

SysTaskFolderTypeName_cat
Heropend
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0.0
0.5
1.0
1.5
2.0
2.5
3.0
SysTaskFolderTypeName_cat
Training data
Test data
Test data pred
Validation data
Validation data pred


ReviewerEdited
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10
12
14
ReviewerEdited

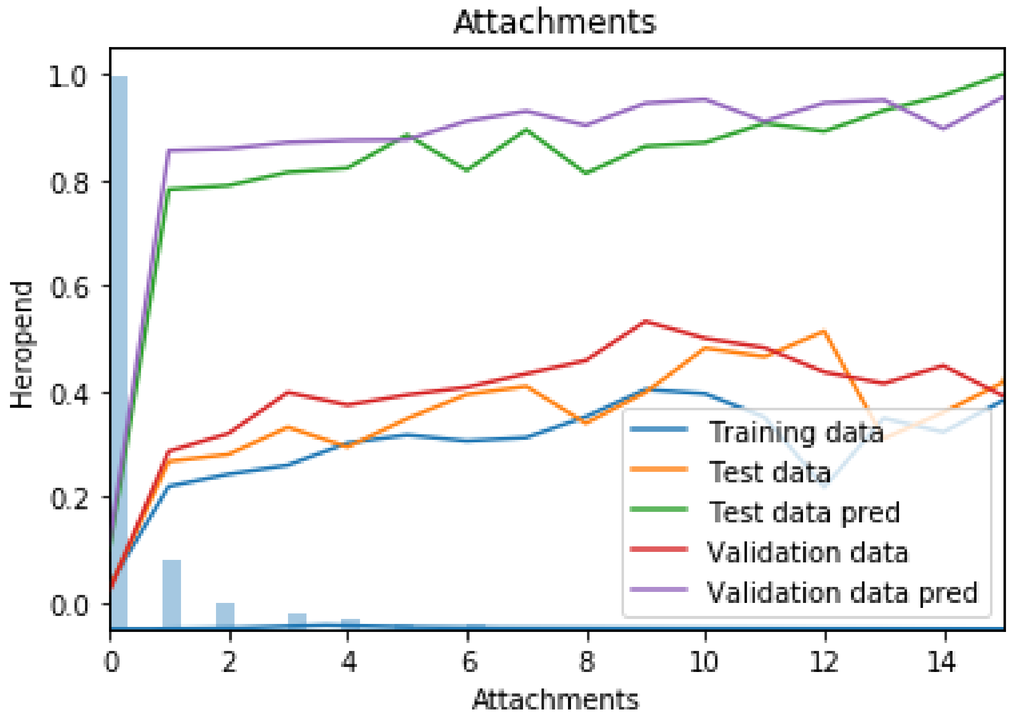

EersteDefinitief
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
−1.4
−1.2
−1.0
−0.8
−0.6
−0.4
−0.2
0.0
1e18
EersteDefinitief

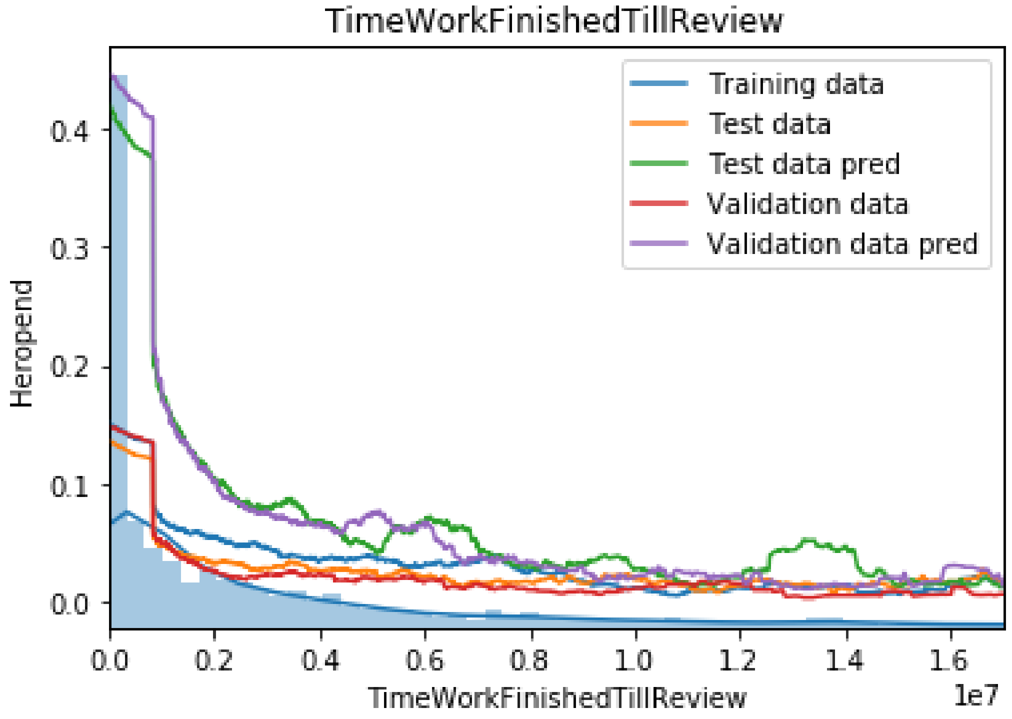

TaskAssignedCount
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
1.0
0.8
0.6
0.4
0.2
0.0
0
1
2
3
4
5
TaskAssignedCount

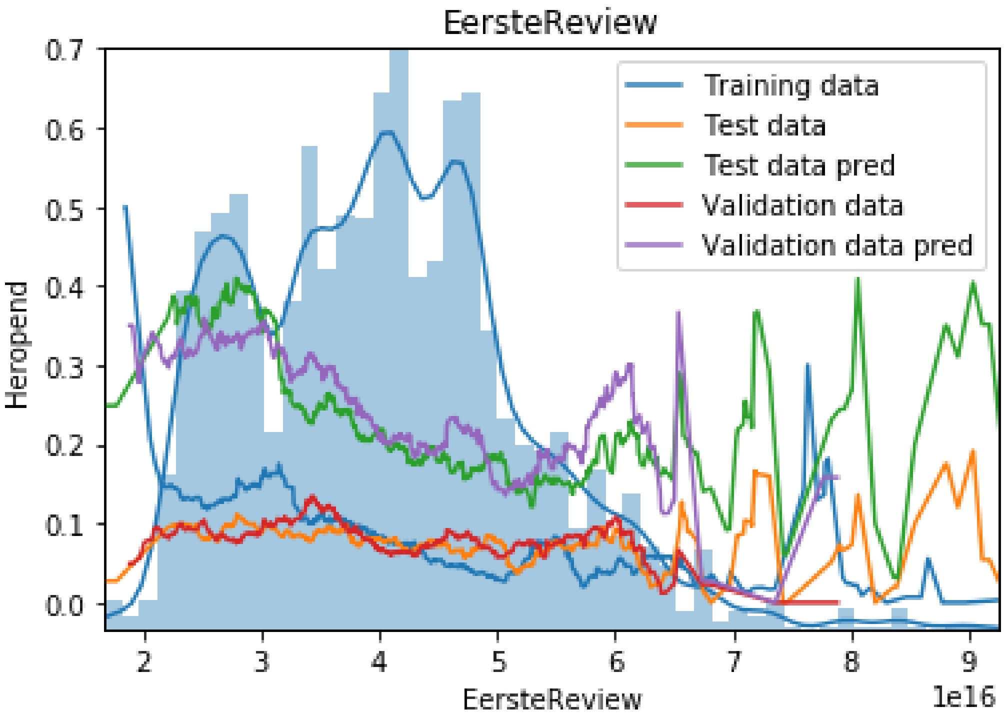

SysCreationTypeID_cat
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
SysCreationTypeID_cat

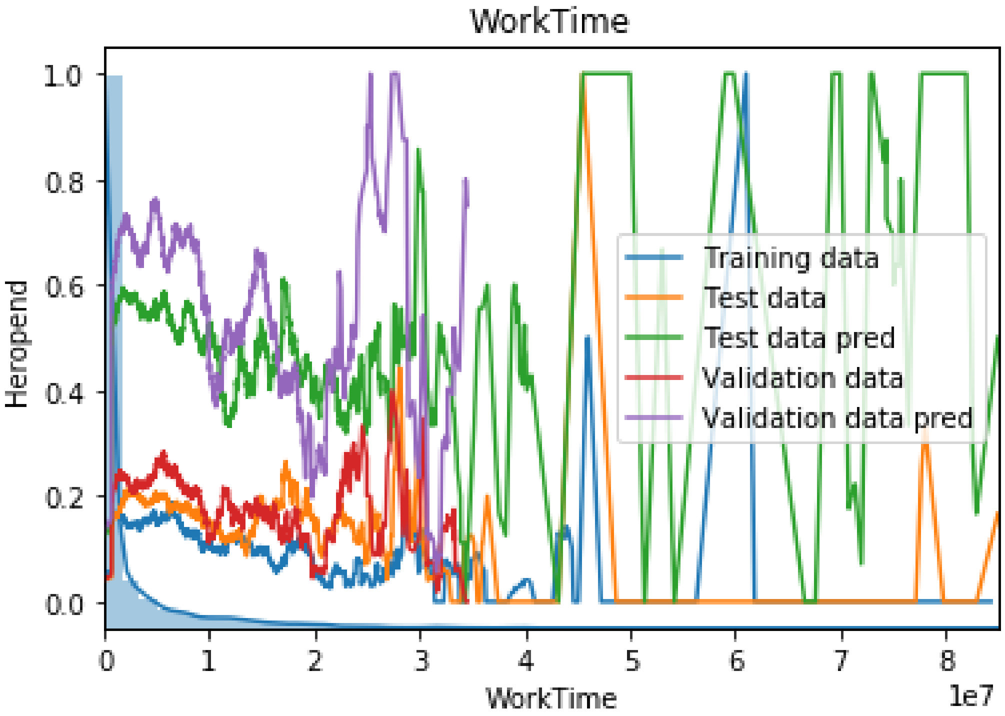

Risicocount
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
0.00
0.02
0.04
0.06
0.08
0.0
0.2
0.4
0.6
0.8
1.0
Risicocount

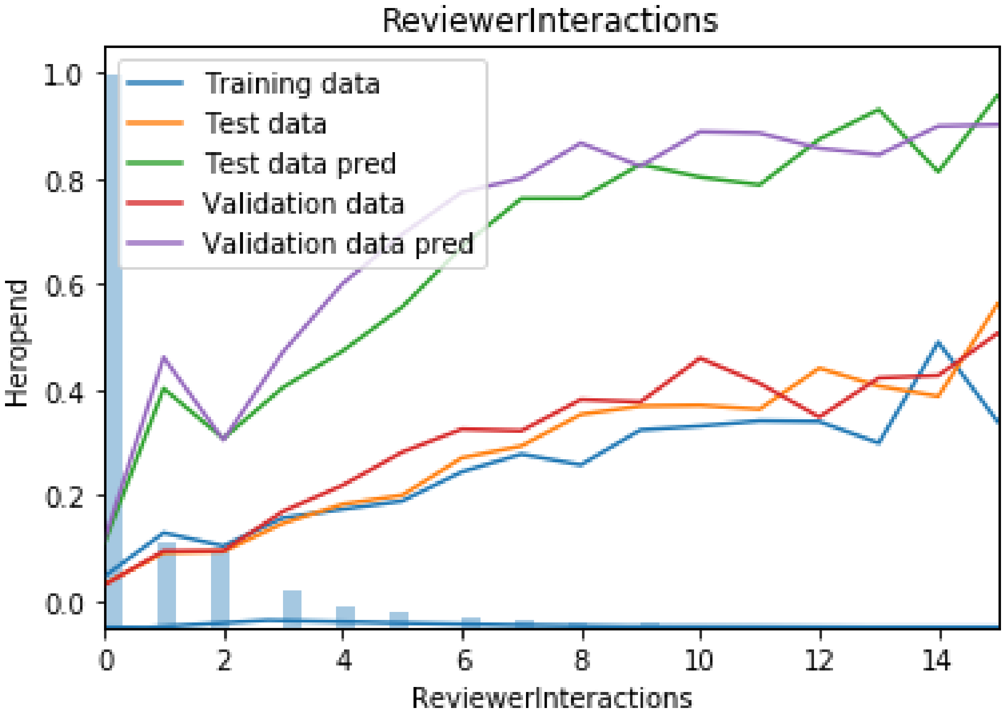
9.2 Predictions per parameter for the random forest minimizing false negatives

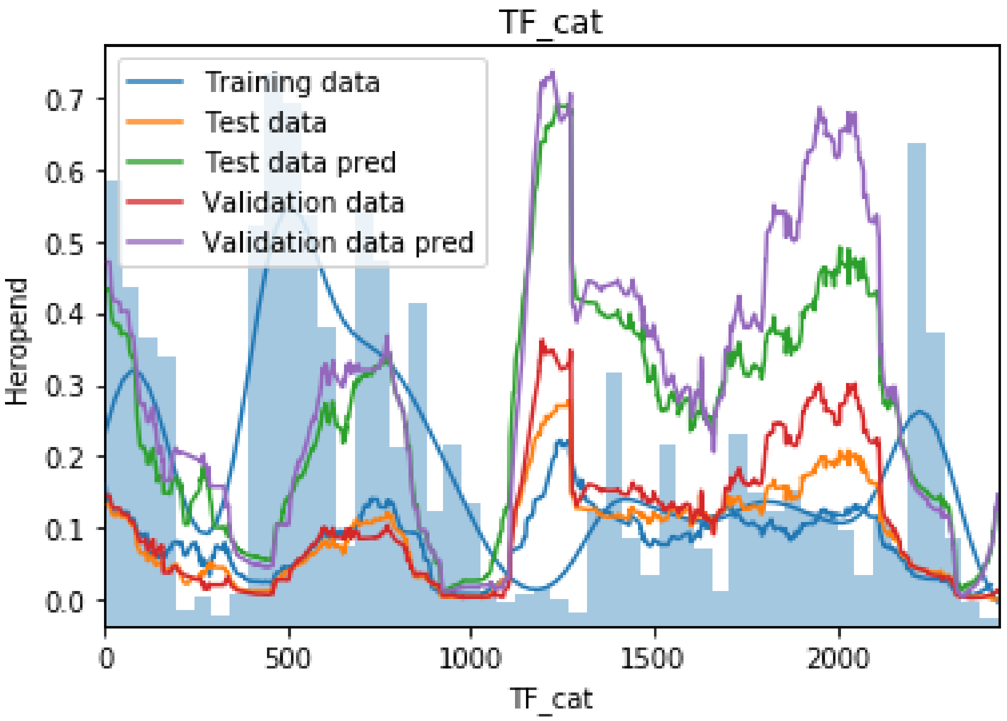


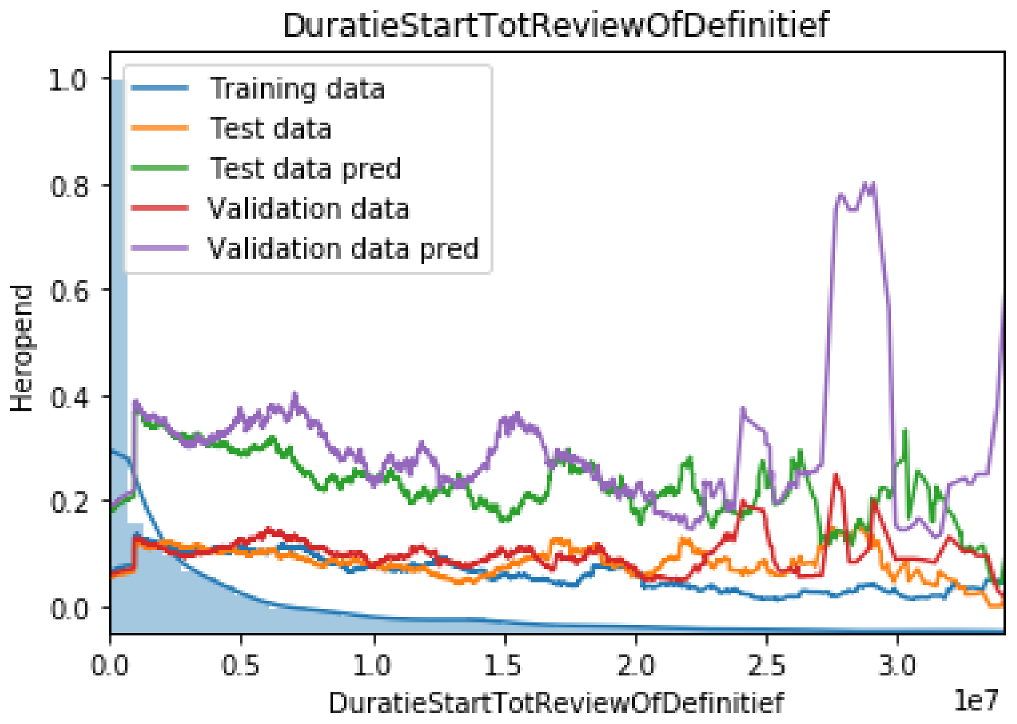


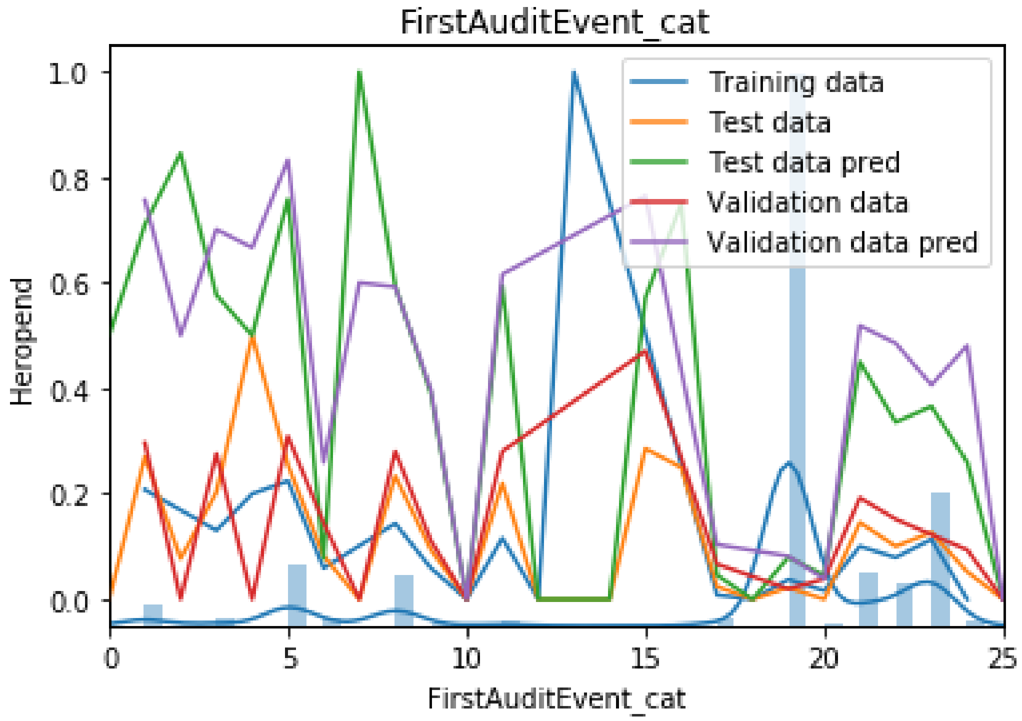

EersteReview
Heropend
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
2
3
4
5
6
7
8
9
1e16
EersteReview
Training data
Test data
Test data pred
Validation data
Validation data pred

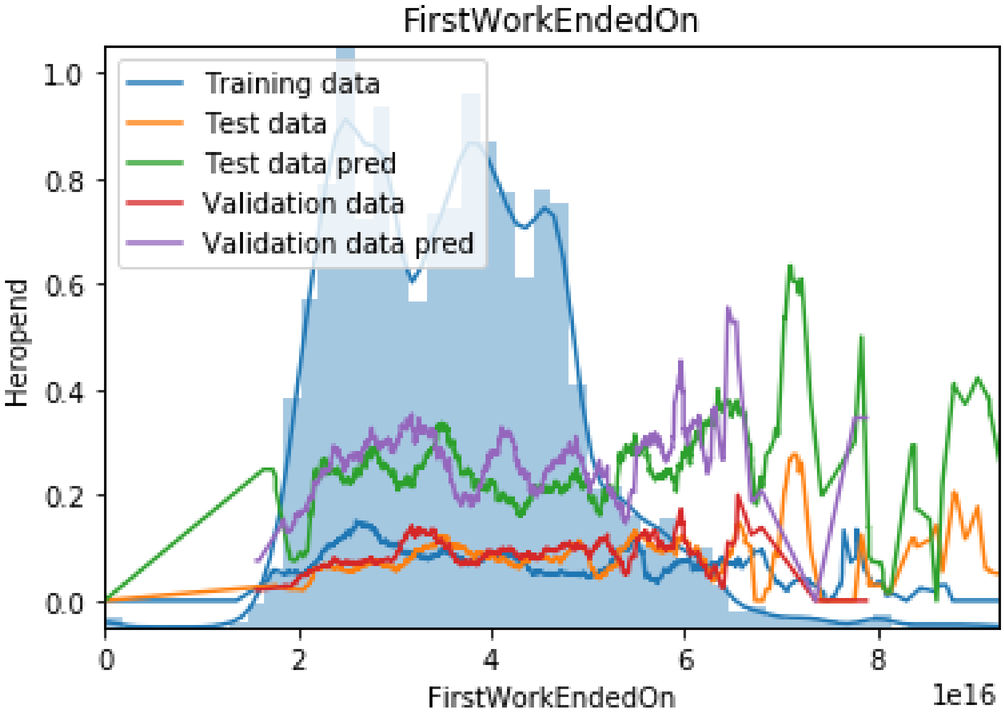

WorkTime
Heropend
1.0
0.8
0.6
0.4
0.2
0.0
0
1
2
3
4
5
6
7
8
1e7
WorkTime
Training data
Test data
Test data pred
Validation data
Validation data pred

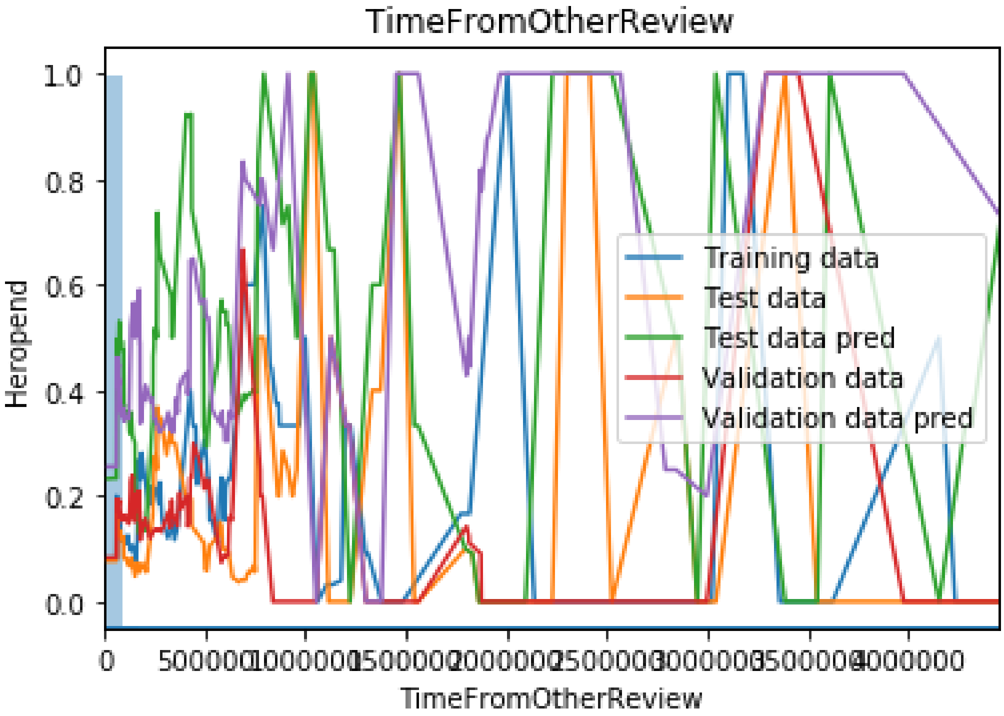

ReviewerInteractions
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10
12
14
ReviewerInteractions

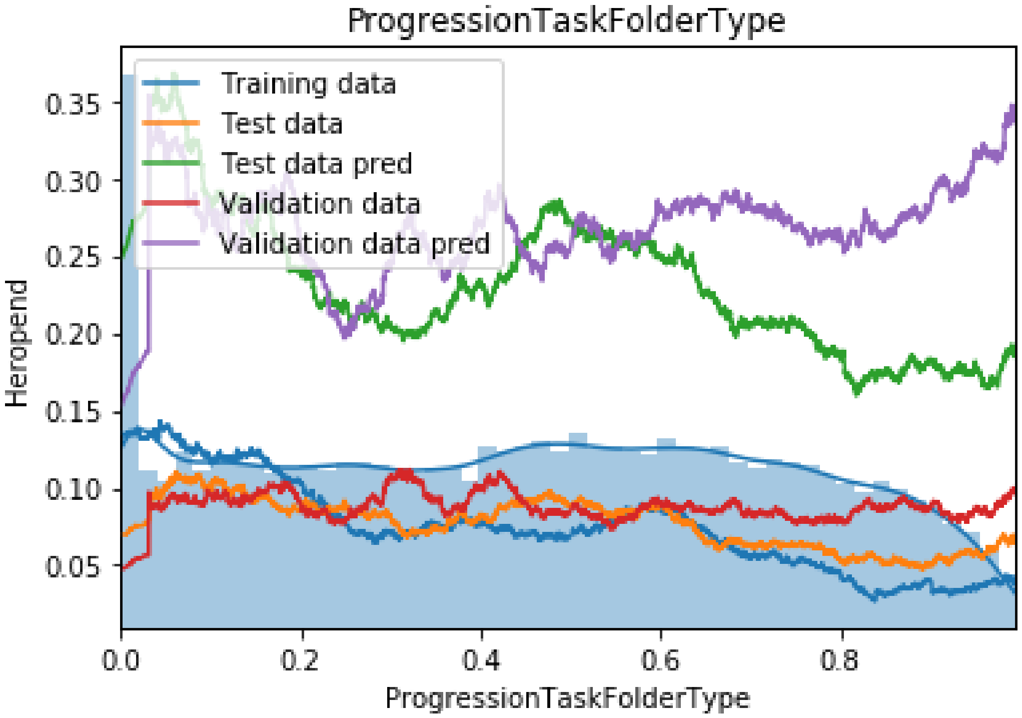

TF_cat
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
500
1000
1500
2000
TF_cat

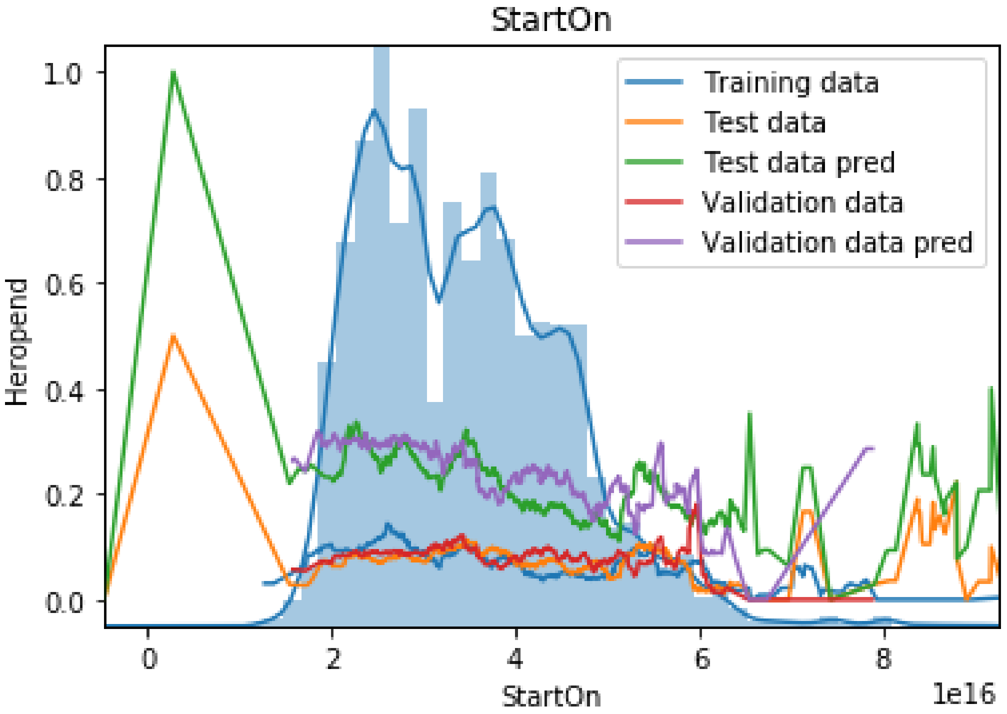

DuratieStartTotReviewOfDefinitief
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
1e7
DuratieStartTotReviewOfDefinitief

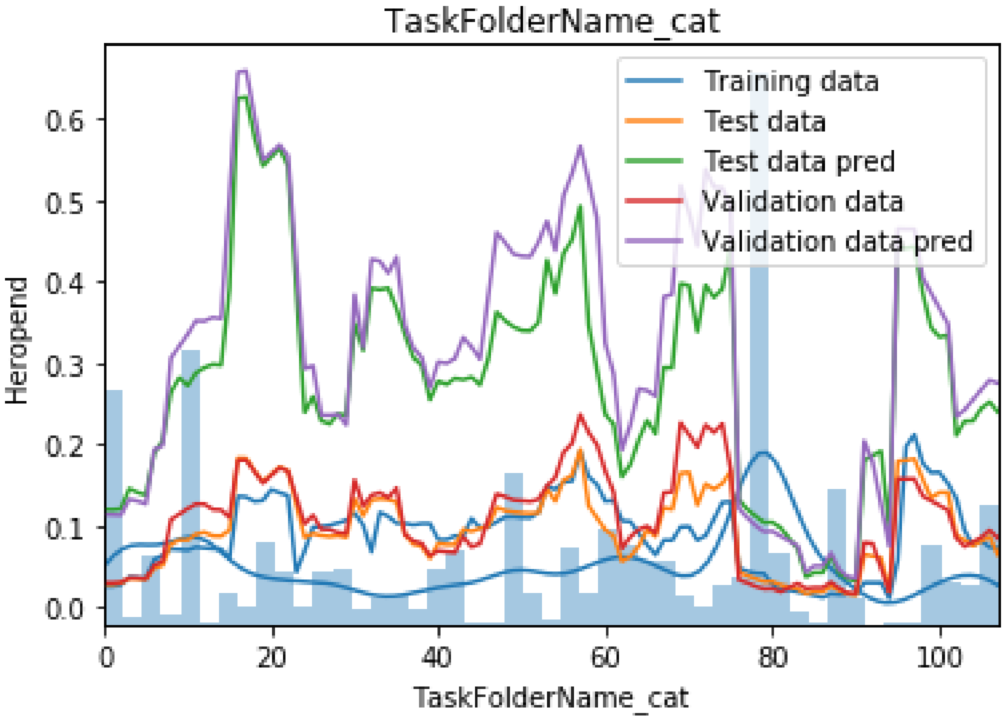

FirstAuditEvent_cat
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
25
FirstAuditEvent_cat

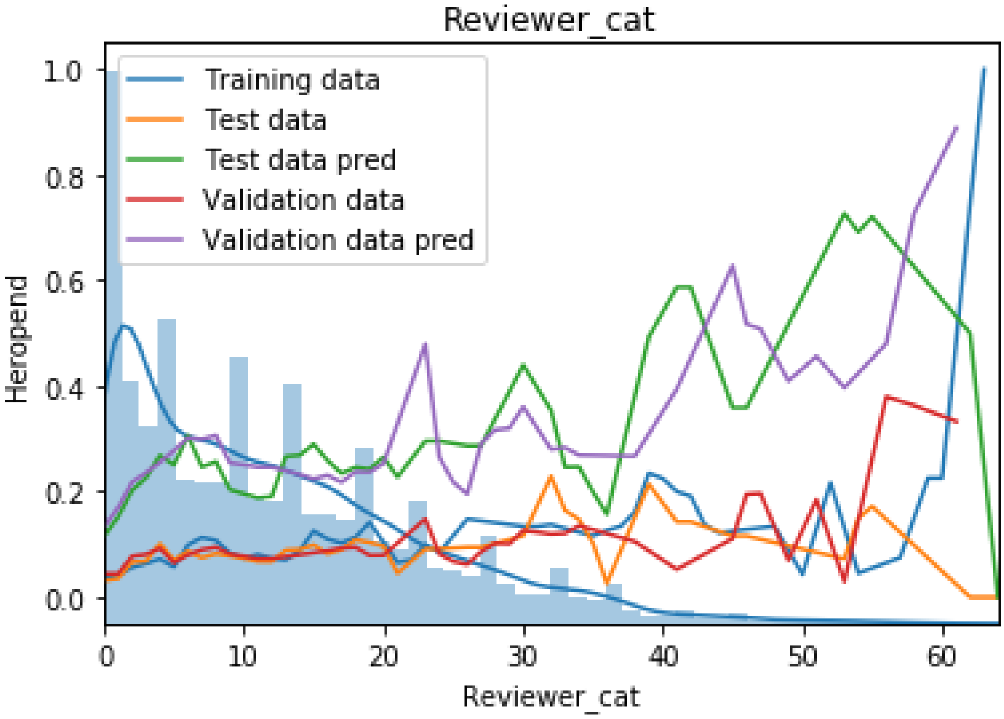

FirstWorkEndedOn
Heropend
1.0
0.8
0.6
0.4
0.2
0.0
Training data
Test data
Test data pred
Validation data
Validation data pred
0
2
4
6
8
FirstWorkEndedOn
1e16

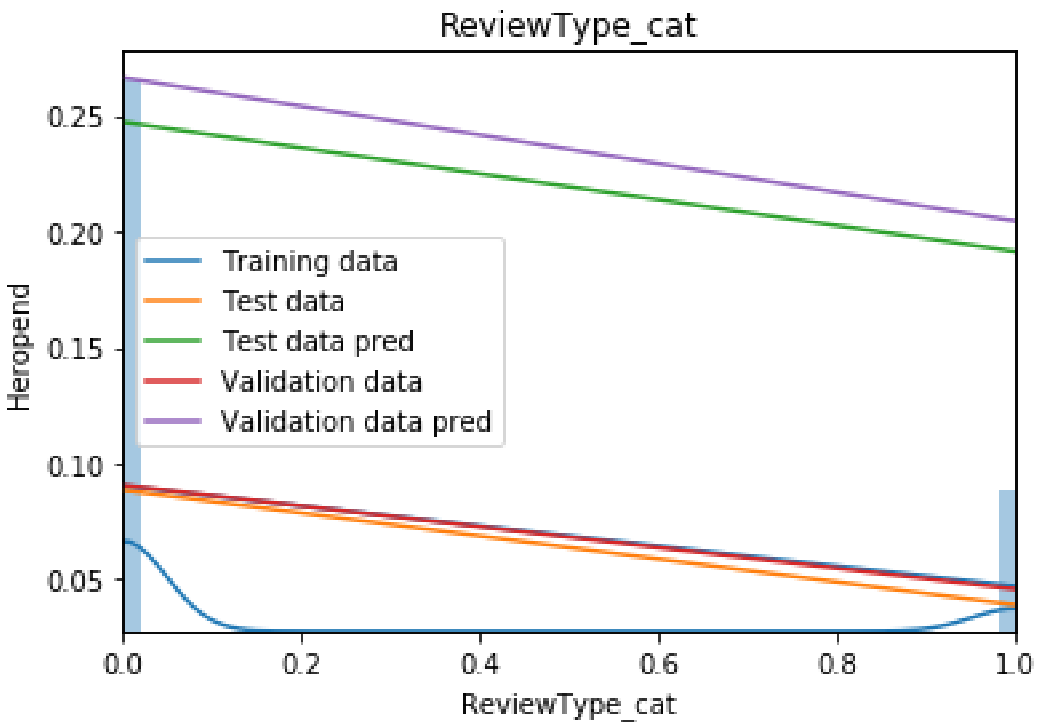

TimeFromOtherReview
Heropend
1.0
0.8
0.6
0.4
0.2
0.0
Training data
Test data
Test data pred
Validation data
Validation data pred
0
500000
1000000
1500000
2000000
2500000
3000000
3500000
4000000
TimeFromOtherReview

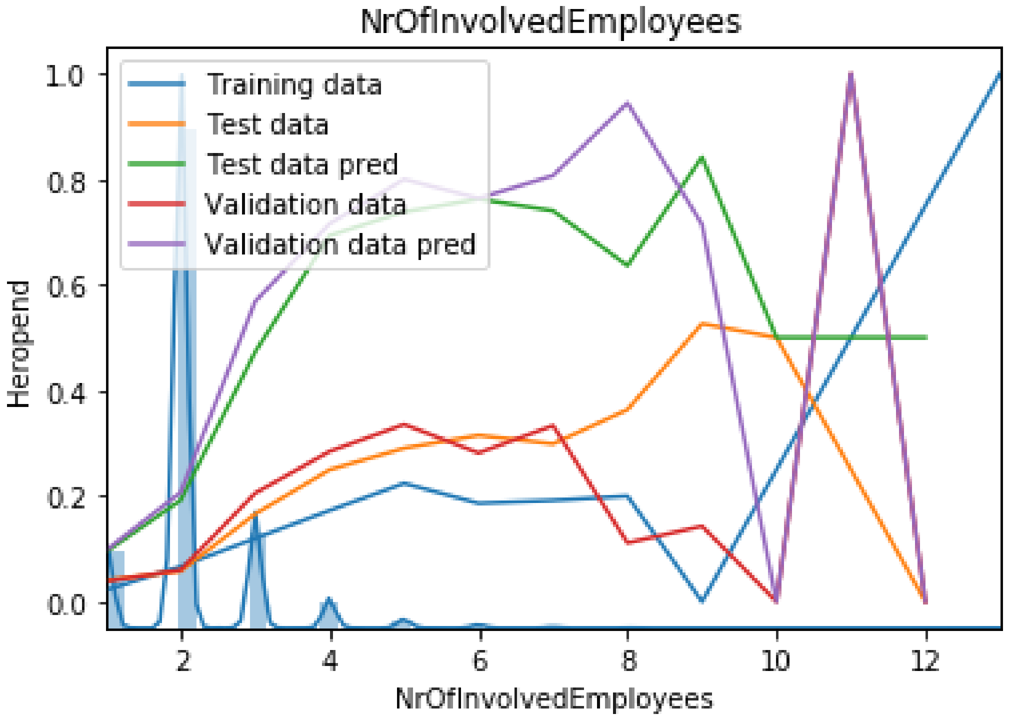

ProgressionTaskFolderType
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.0
0.2
0.4
0.6
0.8
ProgressionTaskFolderType

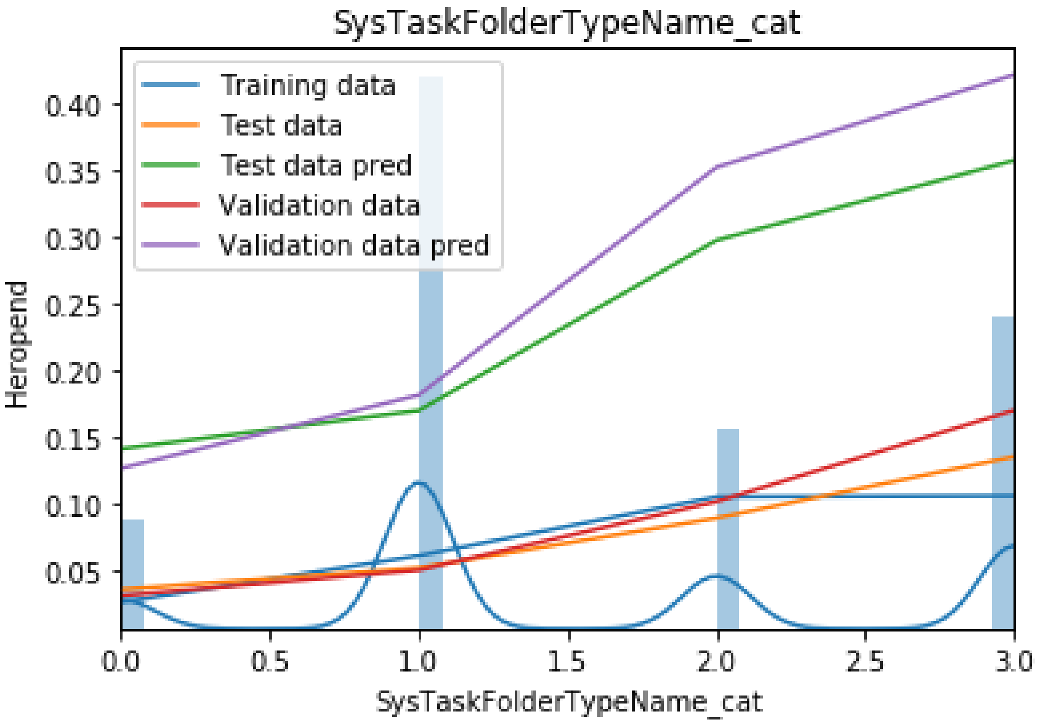

StartOn
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
StartOn
1e16

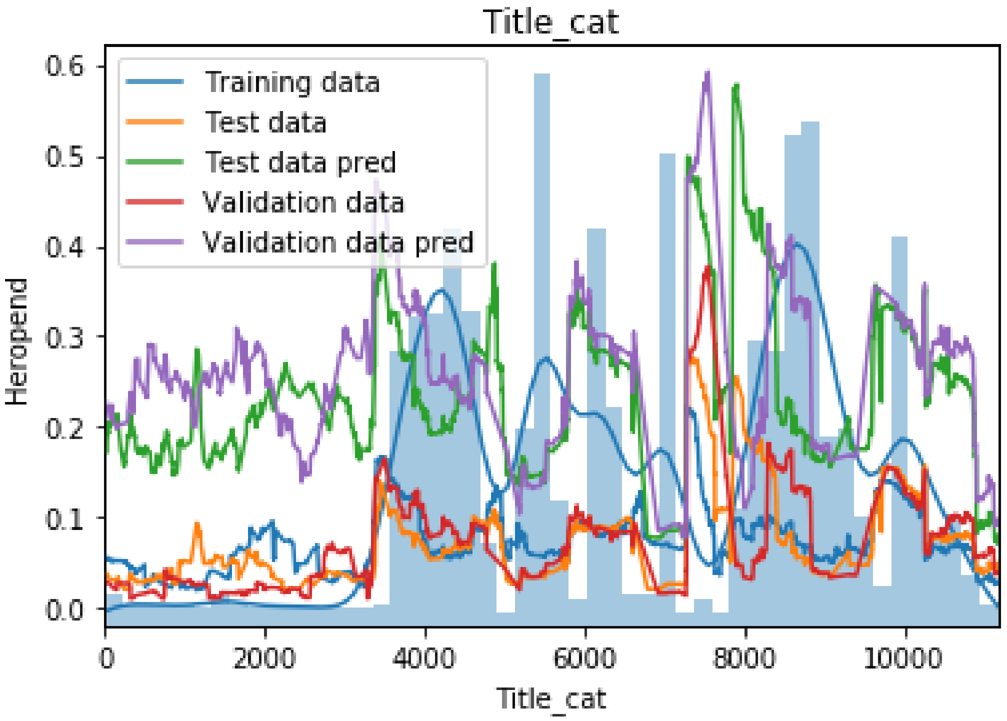

TaskFolderName_cat
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
20
40
60
80
100
TaskFolderName_cat

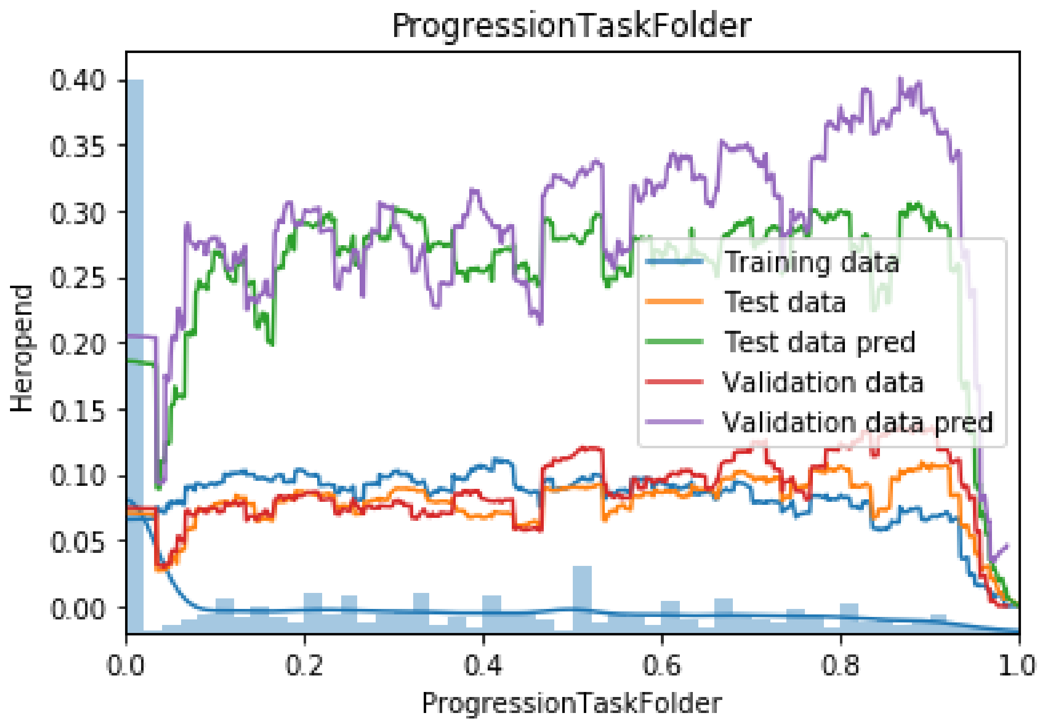

Reviewer_cat
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
0.0
0.2
0.4
0.6
0.8
1.0
0
10
20
30
40
50
60
Reviewer_cat

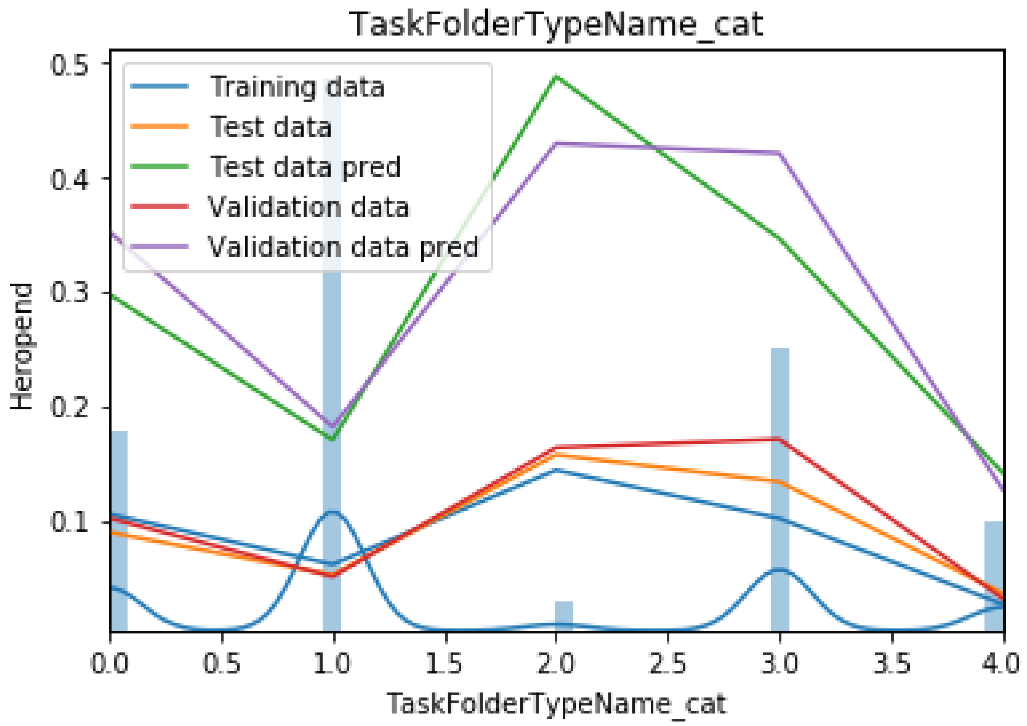

ReviewType_cat
Heropend
0.25
0.20
0.15
0.10
0.05
0.0
0.2
0.4
0.6
0.8
1.0
ReviewType_cat
Training data
Test data
Test data pred
Validation data
Validation data pred

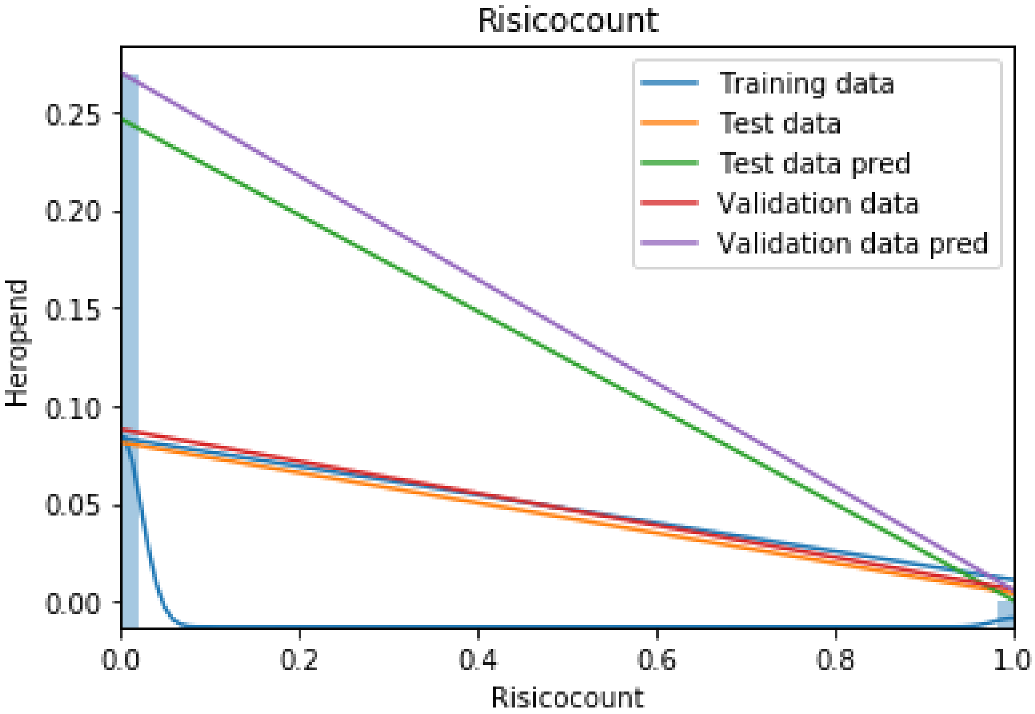

NrOfInvolvedEmployees
Heropend
1.0
0.8
0.6
0.4
0.2
0.0
2
4
6
8
10
12
NrOfInvolvedEmployees
Training data
Test data
Test data pred
Validation data
Validation data pred

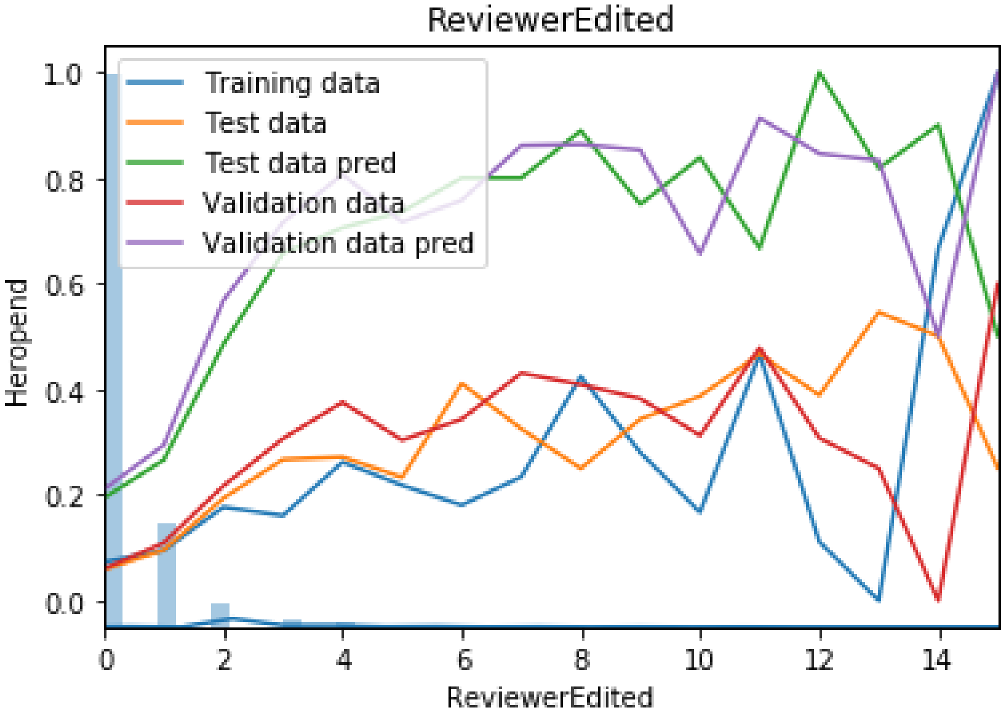

SysTaskFolderTypeName_cat
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.0
0.5
1.0
1.5
2.0
2.5
3.0
SysTaskFolderTypeName_cat

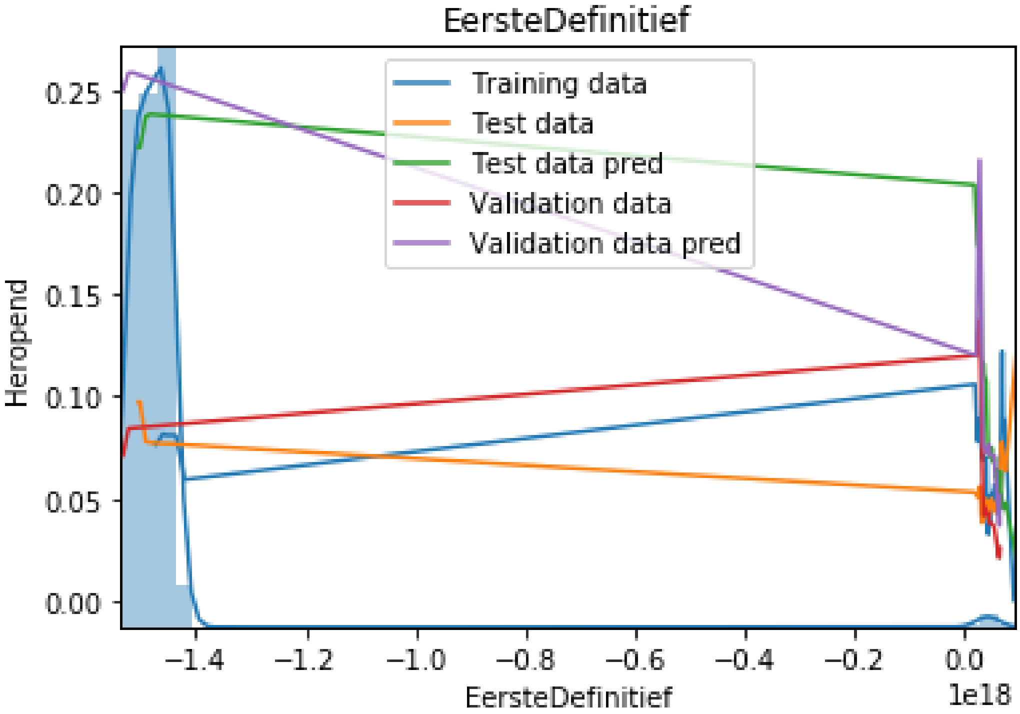

Title_cat
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
2000
4000
6000
8000
10000
Title_cat

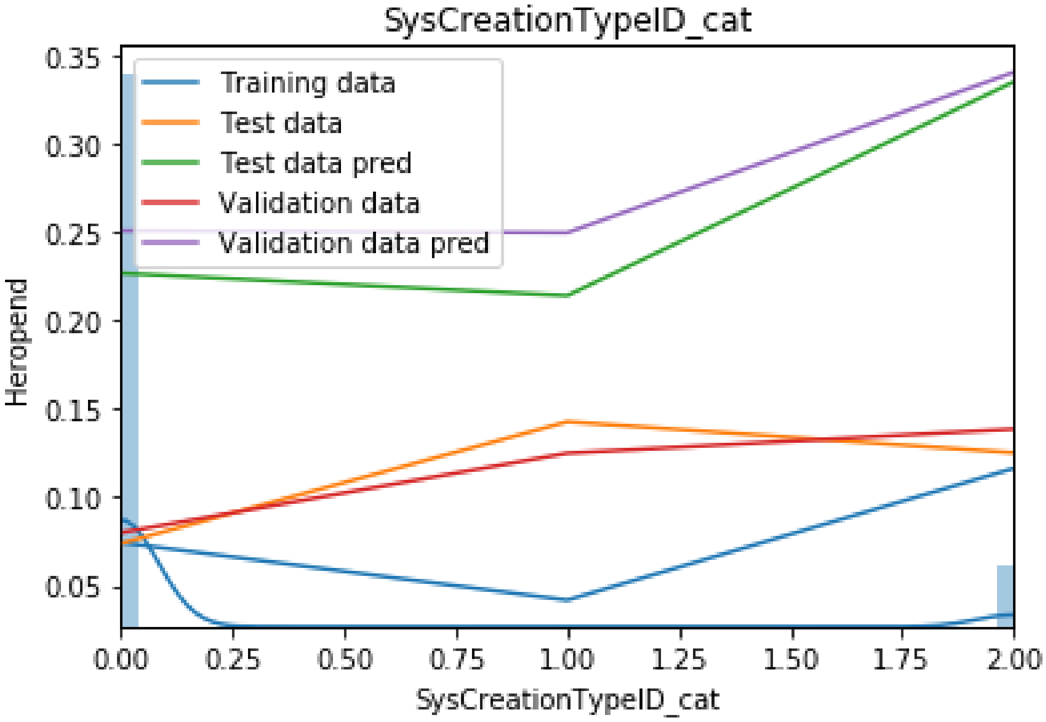

ProgressionTaskFolder
Heropend
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0.0
0.2
0.4
0.6
0.8
1.0
ProgressionTaskFolder
Training data
Test data
Test data pred
Validation data
Validation data pred

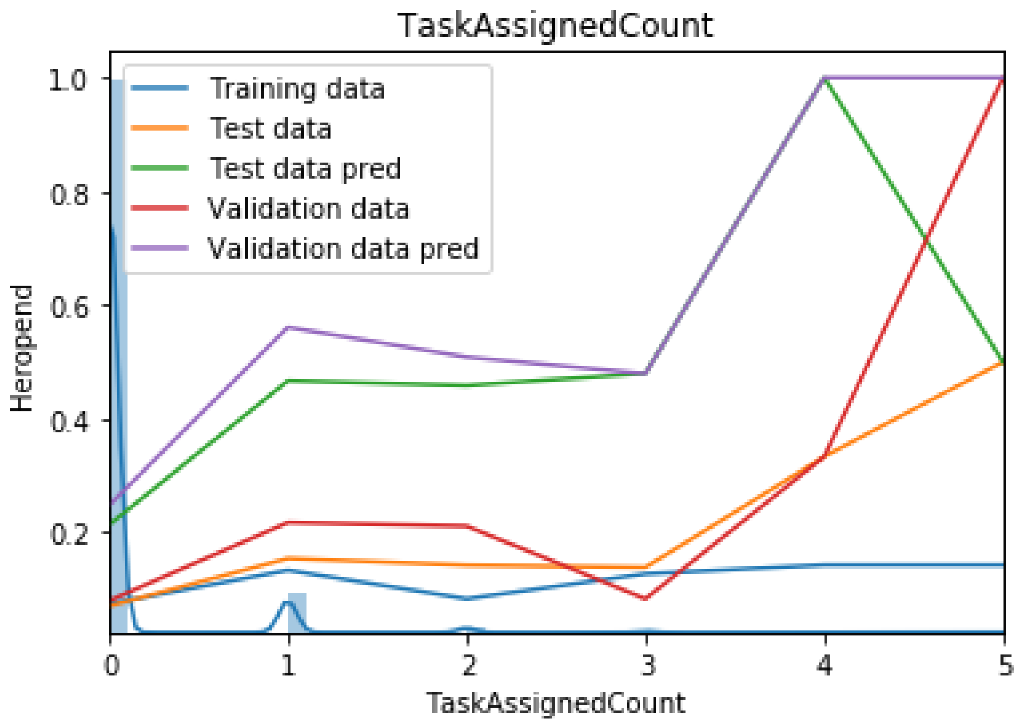

TaskFolderTypeName_cat
Heropend
0.5
0.4
0.3
0.2
0.1
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
TaskFolderTypeName_cat
Training data
Test data
Test data pred
Validation data
Validation data pred


Risicocount
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
0.25
0.20
0.15
0.10
0.05
0.00
0.0
0.2
0.4
0.6
0.8
1.0
Risicocount


ReviewerEdited
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
10
12
14
ReviewerEdited


EersteDefinitief
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
0.25
0.20
0.15
0.10
0.05
0.00
−1.4
−1.2
−1.0
−0.8
−0.6
−0.4
−0.2
0.0
1e18
EersteDefinitief


SysCreationTypeID_cat
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
SysCreationTypeID_cat


TaskAssignedCount
Training data
Test data
Test data pred
Validation data
Validation data pred
Heropend
1.0
0.8
0.6
0.4
0.2
0
1
2
3
4
5
TaskAssignedCount

9.3 SQL query to fetch KPIs

```
SELECT t1.*
        ,DATEDIFF(SECOND, StartOn, MAX(AuditTrail4.DateTime)) AS FirstWorkEndedOn
        ,DATEDIFF(SECOND, StartOn, MAX(AuditTrail4.DateTime)) AS WorkTime
        ,DATEDIFF(SECOND, MAX(AuditTrail4.DateTime), (CASE WHEN t1.EersteReview < t1.EersteDefinitief OR
t1.EersteDefinitief IS NULL THEN t1.EersteReview ELSE t1.EersteDefinitief END)) AS TimeWorkFinishedTillReview
        ,(SELECT TOP 1
                ABS(DATEDIFF(SECOND, at2.DateTime, EersteReview)) AS minDif
                --MIN(CASE WHEN at2.DateTime < EersteReview THEN at2.DateTime - EersteReview ELSE EersteReview
- at2.DateTime END) AS bla
                FROM AuditTrail at2
                WHERE at2.DateTime != EersteReview
                        AND at2.EmployeeID = Reviewer
                        AND at2.SysActionTypeID IN (2,3)
                ORDER BY minDif) AS TimeFromOtherReview
        FROM
        (SELECT DB_NAME() as DBName
                ,Company.CompanyName
                ,Project.ProjectName
                ,Project.DateOpening
                ,Project.DateClosing
                ,SysProjectType.SysProjectTypeName
                ,SysTemplateType.SysTemplateTypeName
                ,Template.TemplateName
                ,Task.TaskID
                ,Task.SysCreationTypeID
                ,Task.Title
                ,Task.SequenceNumber
                ,TaskFolderType.TaskFolderTypeName
                ,TaskFolderType.TaskFolderTypeID
                ,TaskFolderType.SequenceNumber AS TaskFolderTypeSequenceNumber
                ,TaskFolder.TaskFolderName
                ,TaskFolder.Code            AS TF1Code
                ,TaskFolder.SequenceNumber AS TF1SequenceNumber
                ,TF2.Code                                   AS TF2Code
                ,TF2.SequenceNumber                 AS TF2SequenceNumber
                ,TF3.Code                                   AS TF3Code
                ,TF3.SequenceNumber                 AS TF3SequenceNumber
                ,TF4.Code                                   AS TF4Code
                ,TF4.SequenceNumber                 AS TF4SequenceNumber
                ,TF5.Code                                   AS TF5Code
                ,TF5.SequenceNumber                 AS TF5SequenceNumber
                ,SysTaskFolderType.SysTaskFolderTypeID
                ,SysTaskFolderType.SysTaskFolderTypeName
                ,MIN(AuditTrail.DateTime) AS StartOn
                ,AuditTrailReview.DateTime AS EersteReview
                ,AuditTrailReview.EmployeeID AS Reviewer
                ,CASE WHEN AuditTrailReview.SysActionTypeID = 2 THEN 'Partner' ELSE 'Senior' END AS ReviewType
                ,MIN(CASE WHEN AuditTrail.SysActionTypeID = 1 AND AuditTrail.Action = 'Definitief' THEN
AuditTrail.DateTime END) AS EersteDefinitief
                ,DATEDIFF(SECOND, MIN(CASE WHEN AuditTrail.SysActionTypeID <> 1 THEN AuditTrail.DateTime END),
MIN(CASE WHEN AuditTrail.SysActionTypeID IN (2, 3) THEN AuditTrail.DateTime END))
                        AS DuratieStartTotReviewOfDefinitief
                ,COUNT(DISTINCT(AuditTrail.EmployeeID)) AS NrOfInvolvedEmployees
                ,(SELECT COUNT(*) AS nr FROM AuditTrail AuditTrail3 WHERE AuditTrail3.TaskID = Task.TaskID AND
AuditTrail3.SysActionTypeID IN (88, 89)) AS NrOfReviewsOngedaan
                ,(SELECT COUNT(*) AS nr FROM AuditTrail AuditTrail3 WHERE AuditTrail3.TaskID = Task.TaskID AND
AuditTrail3.SysActionTypeID = 90) AS NrOfDefinitiefOngedaan
                ,CASE WHEN (SELECT COUNT(*) AS nr FROM AuditTrail AuditTrail3 WHERE AuditTrail3.TaskID =
Task.TaskID AND AuditTrail3.SysActionTypeID IN (88, 89, 90)) = 0 THEN 0 ELSE 1 END AS Heropend
                ,CASE WHEN (SELECT COUNT(*) AS nr FROM AuditTrail AuditTrail3 WHERE AuditTrail3.TaskID =
Task.TaskID AND AuditTrail3.SysActionTypeID IN (89)) = 0 THEN 0 ELSE 1 END AS HeropendPartner
```

```
            ,CASE WHEN (SELECT COUNT(*) AS nr FROM AuditTrail AuditTrail3 WHERE AuditTrail3.TaskID =
Task.TaskID AND AuditTrail3.SysActionTypeID IN (88)) = 0 THEN 0 ELSE 1 END AS HeropendSenior
            ,(SELECT MAX(minDiff) AS nr FROM (SELECT
                MIN(ABS(DATEDIFF(MINUTE, AuditTrail5.[DateTime], AuditTrail6.[DateTime]))) AS minDiff
                FROM AuditTrail AuditTrail5
                JOIN AuditTrail AuditTrail6 ON AuditTrail5.EmployeeID = AuditTrail6.EmployeeID
                WHERE Audittrail5.TaskID = Task.TaskID
                    AND AuditTrail5.[DateTime] > AuditTrailReview.[DateTime]
                    AND AuditTrail5.[DateTime] > AuditTrail6.[DateTime]
                    AND AuditTrail5.SysActionTypeID IN (1, 10, 11, 13, 54, 55)
                    AND AuditTrail5.TaskID != AuditTrail6.TaskID
                GROUP BY
                    AuditTrail5.AuditTrailID
            ) AS Wijzigingen) AS HeropendGroteWijziging
            ,(SELECT COUNT(*)
                FROM AuditTrail Audittrail5
                WHERE Audittrail5.TaskID = Task.TaskID
                    AND Audittrail5.SysActionTypeID  IN (10, 11)) AS HeropendAangepast
            ,(SELECT TOP(1) SysActionTypeTmp.SysActionTypeName FROM AuditTrail AuditTrailTmp JOIN
SysActionType SysActionTypeTmp ON SysActionTypeTmp.SysActionTypeID = AuditTrailTmp.SysActionTypeID WHERE
AuditTrailTmp.TaskID = Task.TaskID ORDER BY AuditTrailTmp.DateTime) AS FirstAuditEvent
            ,COUNT(DISTINCT RiskProfile.RiskProfileID) AS Risicocount
            ,RiskProfile.RiskPreInterim AS RiskPreInterim
            ,RiskProfile.RiskPostAudit AS RiskPostAudit
            ,Employee.Code AS EmployeeCode
            ,Employee.EmployeeName
            ,(SELECT
                COUNT(CASE WHEN t5.IsReviewed = 1 THEN 1 END)*1./COUNT(*) AS
ProgressionTaskFolderType
                --COUNT(CASE WHEN t5.IsReviewed = 1 THEN 1 END) AS Progression1
                --COUNT(DISTINCT t5.TaskID) AS Progression
                FROM
                    (SELECT
                         CASE WHEN (COUNT(CASE WHEN at3.SysActionTypeID IN (2,3) THEN 1
END) - COUNT(CASE WHEN at3.SysActionTypeID IN (88,89) THEN 1 END)) >= 1 THEN 1 ELSE 0 END AS IsReviewed
                        ,t2.TaskId
                    FROM Task t2
                        JOIN TaskFolder tf2 ON tf2.TaskFolderID = t2.TaskFolderID
                        JOIN AuditTrail at3 ON at3.TaskID = t2.TaskID
                    WHERE tf2.TaskFolderTypeID = TaskFolderType.TaskFolderTypeID
                        AND t2.Hidden = 0 AND t2.Deleted = 0
                        AND tf2.Hidden = 0 AND tf2.Deleted = 0
                        AND at3.DateTime < AuditTrailReview.DateTime
                    GROUP BY t2.TaskID) AS t5) AS ProgressionTaskFolderType
            ,(SELECT
                COUNT(CASE WHEN t5.IsReviewed = 1 THEN 1 END)*1./COUNT(*) AS ProgressionTaskFolder
                --COUNT(CASE WHEN t5.IsReviewed = 1 THEN 1 END) AS Progression1
                --COUNT(DISTINCT t5.TaskID) AS Progression
                FROM
                    (SELECT
                         CASE WHEN (COUNT(CASE WHEN at3.SysActionTypeID IN (2,3) THEN 1
END) - COUNT(CASE WHEN at3.SysActionTypeID IN (88,89) THEN 1 END)) >= 1 THEN 1 ELSE 0 END AS IsReviewed
                        ,t2.TaskId
                    FROM Task t2
                        JOIN TaskFolder tf2 ON tf2.TaskFolderID = t2.TaskFolderID
                        JOIN AuditTrail at3 ON at3.TaskID = t2.TaskID
                    WHERE tf2.TaskFolderID = TaskFolder.TaskFolderID
                        AND t2.Hidden = 0 AND t2.Deleted = 0
                        AND tf2.Hidden = 0 AND tf2.Deleted = 0
                        AND at3.DateTime < AuditTrailReview.DateTime
                    GROUP BY t2.TaskID) AS t5) AS ProgressionTaskFolder
            ,(SELECT COUNT(*)
                FROM AuditTrail at2
                WHERE at2.EmployeeID = AuditTrailReview.EmployeeID
                    AND at2.DateTime < AuditTrailReview.DateTime
                    AND at2.TaskID = Task.TaskID
                ) AS ReviewerInteractions
            ,(SELECT
                (COUNT(CASE WHEN at2.SysActionTypeID = 10 THEN 1 END) - COUNT(CASE WHEN
at2.SysActionTypeID = 11 THEN 1 END)) AS AttachmentCount
                FROM AuditTrail at2
                WHERE at2.DateTime < AuditTrailReview.DateTime
                    AND at2.TaskID = Task.TaskID
                ) AS Attachments
            ,(SELECT
                COUNT(*)
                FROM AuditTrail at2
                WHERE at2.DateTime < AuditTrailReview.DateTime
                    AND at2.TaskID = Task.TaskID
                    AND at2.SysActionTypeID IN (1, 10, 13, 41)
                    AND at2.EmployeeID = AuditTrailReview.EmployeeID
                ) AS ReviewerEdited
            ,(SELECT
                COUNT(*)
                FROM AuditTrail at2
                WHERE at2.DateTime < AuditTrailReview.DateTime
                    AND at2.TaskID = Task.TaskID
                    AND at2.SysActionTypeID = 93
                ) AS TaskAssignedCount
            ,COUNT(DISTINCT(NoteThread.NoteThreadID)) NoteThreadCount
            ,COUNT(DISTINCT(Note.NoteID)) NoteCount
            FROM Task
                JOIN TaskFolder                         ON TaskFolder.TaskFolderID
        = Task.TaskFolderID
                JOIN TaskFolderType                     ON TaskFolderType.TaskFolderTypeID  =
TaskFolder.TaskFolderTypeID
                JOIN Template                           ON TaskFolderType.TemplateID
        = Template.TemplateID
                JOIN Project                            ON Project.TemplateID
            = Template.TemplateID
                JOIN SysTemplateType                ON SysTemplateType.SysTemplateTypeID=
Template.SysTemplateTypeID
                JOIN SysProjectType                     ON SysProjectType.SysProjectTypeID   =
SysTemplateType.SysProjectTypeID
                JOIN SysTaskFolderType              ON SysTaskFolderType.SysTaskFolderTypeID       =
TaskFolderType.SysTaskFolderTypeID
                JOIN Company                            ON Company.CompanyID
            = Project.CompanyID
                LEFT JOIN AuditTrail                ON AuditTrail.TaskID
        = Task.TaskID                  AND      AuditTrail.SysActionTypeID NOT IN (88, 89, 90)
                LEFT JOIN RiskProfile               ON RiskProfile.TaskID
        = Task.TaskID
                LEFT JOIN NoteThread                ON NoteThread.TaskID
        = Task.TaskID
                LEFT JOIN Note                          ON Note.NoteThreadID
            = NoteThread.NoteThreadID
                LEFT JOIN TaskFolder AS TF2  ON TF2.TaskFolderID
        = TaskFolder.ParentTaskFolderID
                LEFT JOIN TaskFolder AS TF3 ON TF3.TaskFolderID
        = TF2.ParentTaskFolderID
                LEFT JOIN TaskFolder AS TF4 ON TF4.TaskFolderID
        = TF3.ParentTaskFolderID
                LEFT JOIN TaskFolder AS TF5 ON TF5.TaskFolderID
        = TF4.ParentTaskFolderID
                CROSS APPLY (
                    SELECT Top 1 *
```

```
                    FROM AuditTrail AuditTrailReviews
                    WHERE AuditTrailReviews.SysActionTypeID IN (2, 3)
                        AND AuditTrailReviews.TaskID = Task.TaskID
                    ORDER BY AuditTrailReviews.DateTime
                ) AuditTrailReview
                LEFT JOIN Employee                      ON AuditTrailReview.EmployeeID
    = Employee.EmployeeID
        WHERE
                    Task.Hidden = 0
                AND Task.Deleted = 0
                AND TaskFolder.Hidden = 0
                AND TaskFolder.Deleted = 0
                AND TaskFolderType.Deleted = 0
                AND Project.AuditfileClosed = 1
                AND (AuditTrail.DateTime IS NULL
                    OR (AuditTrail.DateTime <= COALESCE((SELECT TOP 1 DateTime FROM AuditTrail
AuditTrail2 WHERE AuditTrail2.TaskID = Task.TaskId AND AuditTrail2.SysActionTypeID IN (2, 3) ORDER BY DateTime),
CURRENT_TIMESTAMP)))
                AND SysProjectType.SysProjectTypeName IN ('Controle', 'Audit')
        GROUP BY Task.TaskID
                ,Project.ProjectName
                ,Company.CompanyName
                ,Project.DateOpening
                ,Project.DateClosing
                ,SysProjectType.SysProjectTypeName
                ,SysTemplateType.SysTemplateTypeName
                ,Template.TemplateName
                ,Task.SysCreationTypeID
                ,Task.Title
                ,Task.SequenceNumber
                ,TaskFolderType.SysTaskFolderTypeID
                ,TaskFolderType.TaskFolderTypeName
                ,TaskFolderType.TaskFolderTypeID
                ,TaskFolderType.SequenceNumber
                ,TaskFolder.TaskFolderName
                ,TaskFolder.TaskFolderID
                ,TaskFolder.Code
                ,TaskFolder.SequenceNumber
                ,TF2.Code
                ,TF2.SequenceNumber
                ,TF3.Code
                ,TF3.SequenceNumber
                ,TF4.Code
                ,TF4.SequenceNumber
                ,TF5.Code
                ,TF5.SequenceNumber
                ,SysTaskFolderType.SysTaskFolderTypeID
                ,SysTaskFolderType.SysTaskFolderTypeName
                ,AuditTrailReview.SysActionTypeID
                ,AuditTrailReview.DateTime
                ,AuditTrailReview.EmployeeID
                ,Employee.Code
                ,Employee.EmployeeName
                ,RiskPreInterim
                ,RiskPostAudit) AS t1
    LEFT JOIN AuditTrail AS AuditTrail4 ON AuditTrail4.TaskID = t1.TaskID AND AuditTrail4.DateTime < (SELECT CASE
WHEN t1.EersteReview < t1.EersteDefinitief OR t1.EersteDefinitief IS NULL THEN t1.EersteReview ELSE t1.EersteDefinitief
END)
    WHERE StartOn IS NOT NULL
    GROUP BY
        DBName
        ,CompanyName
        ,ProjectName
```

```
,DateOpening
,DateClosing
,SysProjectTypeName
,SysTemplateTypeName
,TemplateName
,t1.TaskID
,SysCreationTypeID
,Title
,SequenceNumber
,TaskFolderTypeName
,TaskFolderTypeSequenceNumber
,TaskFolderName
,TaskFolderTypeID
,TF1Code
,TF1SequenceNumber
,TF2Code
,TF2SequenceNumber
,TF3Code
,TF3SequenceNumber
,TF4Code
,TF4SequenceNumber
,TF5Code
,TF5SequenceNumber
,SysTaskFolderTypeID
,SysTaskFolderTypeName
,StartOn
,EersteReview
,EersteDefinitief
,DuratieStartTotReviewOfDefinitief
,NrOfInvolvedEmployees
,NrOfReviewsOngedaan
,NrOfDefinitiefOngedaan
,Heropend
,HeropendPartner
,HeropendSenior
,HeropendGroteWijziging
,HeropendAangepast
,FirstAuditEvent
,ReviewType
,Risicocount
,ReviewerInteractions
,ProgressionTaskFolderType
,ProgressionTaskFolder
,Attachments
,ReviewerEdited
,TaskAssignedCount
,Reviewer
,NoteThreadCount
,NoteCount
,RiskPreInterim
,RiskPostAudit
,EmployeeCode
,EmployeeName
```