Eindhoven University of Technology

MASTER

Predicting employee turnover and reducing turnover costs using machine learning techniques

Klop, G.

Award date:
2021

Link to publication

Disclaimer
This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 04. Apr. 2025

EINDHOVEN UNIVERSITY OF TECHNOLOGY

Industrial Engineering & Innovation Sciences
Operations Management and Logistics Group

Predicting employee turnover and reducing turnover costs using machine learning techniques

Master Thesis

In partial fulfillment of the requirements for the degree of
Master of Science
In Operations Management and Logistics

G. Klop (Guido)

TU/e Supervisors:
Dr. M. Mirzaei (Masoud)
Dr. Z. Bukhsh (Zaharah)
Dr.ir. N.P. Dellaert (Nico)

Company Supervisors:
S.F.J. de Vries (Sjoerd)
R. van der Pool (Richard)
G.A. Sannes (Greta)

Redacted Version

Eindhoven, August 2021

Abstract

This master thesis deals with the prediction and interpretation of voluntary employee turnover using machine learning and relates those findings to a cost reduction. In order to provide a theoretical basis, different employee turnover costing frameworks, machine learning methods and feature importances from literature are evaluated. Based on these findings, insights into current turnover and retention costs for the company are obtained. Furthermore, historic employee turnover data is used to train and optimize six state-of-the-art machine learning models, which classify employees who are at risk of leaving. Ultimately, an optimal version of the best performing model is proposed and related to tangible insights. Individual classifications of this model are interpreted so that the insights can be used for global and individual retention strategies. A novel technique named SHAP is used to for this interpretation. Additionally, guidelines are provided on how to obtain these insights in novel situations when the model is used in practice. Lastly, the final model performance and interpretations are related to the turnover and retention costs, providing an indication of a possible cost reduction that can be achieved.

Executive Summary

Introduction

Employee turnover is the departure of people, and thus intellectual capital, from a company (Punnoose & Ajit, 2016). Employee turnover can be costly for companies, as they need to replace the employees that leave. To elaborate, turnover incurs separation costs, replacement costs and placing costs estimated at several thousand dollars per employee (McKinney et al., 2007).

For many companies this kind of employee behavior is a black box, due to the wide variety of reasons that can drive an employee to leave a company to work elsewhere. This thesis opens up this black box by using historical employee turnover data to train a model that can classify employees who are at risk of turnover. Additionally, drivers for turnover are identified, which can be used for global and individual interventions.

Turnover can be split into two different subtypes, voluntary turnover and involuntary turnover. When voluntary turnover occurs, the decision to leave the company is made by the employee without any interference from the company. Involuntary turnover on the other hand occurs when the decision for the employee to depart the company is made by the company instead of the employee (Patel et al., 2020). The focus of this thesis is on the problem of voluntary turnover.

Problem Statement

Increasing retention is important as previous research shows current turnover rates in various sectors to be varying between 8.00% and 28.34% (Fallucchi et al., 2020; Patel et al., 2020; Zhao et al., 2019). Additionally, the costs of turnover can be significant, ranging from ~$2,600 to ~$23,000 per employee, depending on the department and job position of the employee (McKinney et al., 2007). Apart from direct costs such as administrative costs, training costs and screening costs, indirect cost which are less apparent are incurred as well. To illustrate, employees leaving the company take a vast amount of knowledge with them, leaving the company with less intellectual capital (Punnoose & Ajit, 2016). In addition to the reduction of intellectual capital, the team that this employee used to work in is (temporarily) a person short which decreases morale, increases overwork and decreases production (Tziner & Birati, 1996).

The main research question associated with the employee turnover problem is formulated as follows:

How to gain insights into future turnover and apply these insights to reduce employee turnover costs using historical employee turnover data?

In order to answer this main research question, four sub-research questions are defined:

1. What are the current costs associated with employee turnover and employee retention?

2. How to identify future turnover and its causes using machine learning techniques?

3. How do models trained on pandemic data and non-pandemic data compare?

4. How can supervisors and the HR-department interpret the output of the model, such that it can be used for intervention strategies?

Methodology

To answer the research questions in a structured manner, the CRISP-DM framework is applied. This framework divides the problem into six main phases, each associated with a different part of the research problem.

The *business understanding* phase investigates the costs associated with employee turnover and retention. The *Data Understanding & Data Preparation* phase elaborates on the steps that are taken to prepare the company data for machine learning. Subsequently, the *Modeling & Evaluation* phase discusses the use of this dataset for the purpose of machine learning, optimizing a machine learning model and evaluating its performance. Lastly, the *Deployment* phase is partly completed by describing the interpretation of the optimal machine learning model.

Business Understanding

This section has been removed for confidentiality reasons.

Data Understanding & Data Preparation

In order to use the data for machine learning, some preprocessing steps are taken. The dataset used for this research is a product of an integration of data distributed amongst a CRM- and HR-system, one dedicated to hour registration and the other dedicated to employee properties registration. Each entry in the dataset is a unique combination of an employee ID and a period number together with 39 properties of an employee in this specific period. Errors such as missing values, duplicate entries and entry errors are present in the dataset and are remedied. Moreover, entries containing non-temporary employees and involuntary turnover are removed from the dataset. Features containing information that cannot be used directly by a machine learning model are used to create new features or are transformed, resulting in 78 different features. Furthermore, the dataset is split into entries during the pandemic and outside of the pandemic, so that differences in model performance between these datasets can be evaluated. Lastly, the two split datasets are both aggregated from single- into multi-period entries, since turnover intentions develop over a longer period of time instead of one period. Two aggregation approaches are demonstrated: a rolling window and a last window approach, generating multiple versions of a source dataset with different window sizes that can be evaluated.

Modeling & Evaluation

After generating multiple versions of the source dataset, the datasets generated in the preproccessing phase are evaluated. First the different aggregation approaches are evaluated after which the difference between pandemic and non-pandemic data is evaluated. The last windows aggregation leads to better model performance than the rolling windows aggregation for both the pandemic and non-pandemic dataset. Moreover, splitting the dataset into pandemic and non-pandemic dataset, leads to reduced model performance when compared to the full dataset. Therefore, using the full dataset aggregating with a last window approach is preferred.

As a next step, feature selection is applied to the dataset, as a dataset with lower dimensions leads to lower model training times and potentially a higher model performance (Cunningham, 2008). Applying feature selection using RFECV results in an improved model performance and 49 features that should be included in the final dataset. Thereafter, six machine learning models (*Decision Tree*, *Random Forest*, *Gradient Boosting Tree*, *Extreme Gradient Boosting*, *AdaBoost* and *Multilayer Perceptron*) are trained on the dataset and optimized using hyperparameter tuning with random search. The *Extreme Gradient Boosting* model has a superior performance when compared to the other models and is selected for the next modeling step. In this step, the model threshold is optimized using the optimal F1 score and using a custom F_β score based on the costs of false positives (retention costs) and false negatives (turnover costs). Using the F1 score a superior model performance is found with a threshold of 0.283, whereas the custom F_β yields a model that cannot be used in practice.

Interpretation

The resulting optimal model can be interpreted on a global and individual level using SHAP. When interpreting the optimal model on a global level, interesting insights on global retention trends can be obtained. Some important insights on global trends are the following: features describing how the employee evaluates the company are important indicators for turnover, as a bad score is related to a positive contribution to turnover. Moreover, when an employee did not receive a salary increase in the last 12 months, a large contribution to turnover is found. Furthermore, employees with a high salary level are more inclined to remain with the company. A high salary level is often associated with employees in a senior position, due to this position these employees are often more committed to the company.

The insights can also be used for interpretation on an individual level. When an employee is classified as at risk of turnover, individual features contributing to this prediction can be investigated. The HR-department and supervisors can use these insights for individually targeted intervention strategies.

Furthermore, the SHAP values indicating to the importance of each feature are additive. This property makes it possible to add the SHAP values of individuals in different subgroups, enabling scalable interpretations from a a single employee all the way up to a global level.

When interpreting the model, it is important to consider whether a feature can be influenced directly and thus be used for intervention strategies.

Conclusion

This thesis gives important insights into the current costs of turnover and retention. Furthermore, different strategies for aggregating and splitting the dataset are evaluated. Moreover, a model is optimized which classifies a large number of employees at risk of turnover correctly and gives insights into the reasons for this classification. These reasons can help the HR-department and supervisors to develop global and individual retention strategies.

Implementation and interpreting this model could lead to a potential turnover cost reduction and an increase in retention costs. On an individual level, the costs of turnover are much higher than the costs of retention. Therefore, accurately targeting employees at risk using this model should lead to an overall cost reduction. In order to accurately estimate the total cost reduction that can be achieved using this model, more research needs to be conducted into the effect of different retention budgets, their strategies and effects on conversion rates.

Preface

This thesis is the final product of my seven month graduation project and concludes my master Operations Management and Logistics. In these seven months I have grown on a professional, academic and personal level.

The company in which I conducted my thesis helped me grow on a professional level. They have supported me greatly in my professional growth by providing a thought-provoking environment with interesting and involved colleagues. I want to thank Sjoerd de Vries in particular for his never-ending support during this project, he was closely involved in the data gathering process and made sure I was supported in all possible ways. Furthermore, I want to thank Richard van der Pool for his help in this project and the many inspiring conversations we shared. I also want to thank Greta Sannes for her help on the HR related parts of this thesis. I want to thank the data management team for their support during this project and the fun Thursdays when I got to work on location. Lastly, I want to express my gratitude to the company for believing in me and providing me with a job opportunity.

Moreover, my supervisors within the Technical University of Eindhoven helped me grow on an academic level. I want to thank my first supervisor Dr. Masoud Mirzaei for supervising me in the development of this thesis. I also want to thank him for his help, insights, feedback and advice during this project. Moreover, I want to thank my second supervisor Dr. Zaharah Bukhsh for her helpful insights in the area of machine learning.

Lastly, my girlfriend, friends and family helped me grow on a personal level. I want to thank my girlfriend, Dana, in particular. She always supported me during this process, gave valuable advice and took great care of me in busy times. I also want to thank Sil and the rest of my friends for their important feedback and great advice during this project. Lastly, I want to thank my family for continuous support.

Thank you.

Guido Klop
12-08-2021

List of Abbreviations

HR	Human resources
CRM	Customer relationship management
AUC_{Pr}	Area under the precision recall curve
SHAP	SHapley Additive exPlanations
DT	Decision Tree
RF	Random Forest
GBT	Gradient Boosting Tree
XGB	Extreme Gradient Boosting
ADA	AdaBoost
MLP	Multilayer Perceptron
RW	Rolling Window
CRISP-DM	Cross Industry Standard Process for Data Mining
RFECV	Recursive feature elimination with cross validation
LIME	Local Interpretable Model-Agnostic Explanations
TH	Total hours
NA	Number of assignments
SI-L3M	Short illness in the last 3 months
SI-L6M	Short illness in the last 6 months
SI-L12M	Short illness in the last 12 months
LI-L6M	Long illness in the last 6 months
LI-L12M	Long illness in the last 12 months
CS-L3M	Conducted study in the last 3 months
CS-L6M	Conducted study in the last 6 months
V-L3M	Paid leave in the last 3 month
UL-L12M	Unpaid leave in the last 12 months
OW-L3M	Overwork in the last 3 months
OW-L6M	Overwork in the last 6 months
OW-L12M	Overwork in the last 12 months
FG	Salary tier
SP	Salary position
SALI-L12M	Salary increase in the last 12 month
PM-L12M	Promotion in the last 12 months
TT	Travel time
JFR	Job freedom & responsibility
CDL	Culture diversity & leadership
COSA	Company satisfaction
JS	Job searching
ms-G	Marital status: Married
ms-O	Marital status: Unmarried
ms-S	Marital status: Living together
LoS	Line of service
Loc	Location

List of Figures

3.1 Visual representation of a basic decision tree 8
3.2 The schematic of a multilayer perceptron 11
3.3 Confusion matrix 12
3.4 A general example of a precision recall curve 14

5.1 Illustration of the six iterative phases of the CRISP-DM framework 22

7.1 Illustration of the rolling window principle applied to the employee entries 34

8.1 Results of applying the RFECV method for feature reduction to the full dataset . 41
8.2 Illustration of the results of an individual iteration of the threshold tuning process 43

9.1 Overview of the SHAP values when interpreting the test data 47
9.2 Example of the individual feature interpretation for an employee that does not leave the company 49
9.3 Example of the individual feature interpretation for an employee that does leave the company 50

List of Tables

4.1 Results of the quantitative literature review on model performance 18
4.2 Results of the quantitative literature review on feature performance 19

7.1 Example of the structure of the dataset provided by the company 28
7.2 Features included in the original dataset together with their type and description . 29
7.3 Features introduced, transformed and dropped during the feature creation and transformation process 33

8.1 Top 10 performing datasets for the non-pandemic dataset selection 38
8.2 Results for undersampling the best performing non-pandemic rolling window aggregation 39
8.3 Top 10 performing datasets for the pandemic dataset selection 39
8.4 Results for undersampling the best performing pandemic rolling window aggregation 40
8.5 Model performance on the full dataset 40
8.6 Model performance validation after RFECV 42
8.7 Performance scores of the best performing models after hyperparameter tuning . . 43
8.8 Results for threshold tuning 44
8.9 Results for threshold tuning that balances the cost of misclassifications 44

B.1 Full results for the dataset selection non-pandemic 64
B.2 Results for undersampling the best performing non-pandemic rolling window aggregation 65
B.3 Full results for the dataset selection pandemic 67
B.4 Results for undersampling the best performing pandemic rolling window aggregation 68
B.5 Full results for the full dataset 69
B.6 Full model performance validation after RFECV 70
B.7 Resulting feature set from RFECV 70
B.8 Top 50 hyperparameters for hyperparameter tuning with a Decision Tree 72
B.9 Top 50 results for hyperparameter tuning with a Decision Tree 73
B.10 Top 50 hyperparameters for hyperparameter tuning with a Random Forest 74
B.11 Top 50 results for hyperparameter tuning with a Random Forest 75
B.12 Top 50 hyperparameters for hyperparameter tuning with a Gradient Boosting Tree 76
B.13 Top 50 results for hyperparameter tuning with a Gradient Boosting Tree 77
B.14 Top 50 hyperparameters for hyperparameter tuning with Extreme Gradient Boosting 78
B.15 Top 50 results for hyperparameter tuning with Extreme Gradient Boosting 79
B.16 Top 50 hyperparameters for hyperparameter tuning with AdaBoost 80
B.17 Top 50 results for hyperparameter tuning with AdaBoost 81
B.18 Top 50 hyperparameters for hyperparameter tuning with a Multilayer Perceptron . 82
B.19 Top 50 results for hyperparameter tuning with a Multilayer Perceptron 83
B.20 Full results for threshold tuning 84
B.21 Full results for threshold tuning that balances the cost of misclassifications 84

Contents

Abstract iii

Executive Summary vi

Preface vii

List of Abbreviations viii

List of Figures ix

List of Tables x

Contents xi

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Problem Introduction . . . 1
- 1.3 Company Description . . . 2
- 1.4 Contribution . . . 2

2 Problem Statement 3
- 2.1 Detailed Problem Description . . . 3
- 2.2 Research Goal . . . 3
- 2.3 Main Research Question . . . 4
- 2.4 Sub-research Questions . . . 4
  - 2.4.1 Sub-research Question 1 . . . 4
  - 2.4.2 Sub-research Question 2 . . . 4
  - 2.4.3 Sub-research Question 3 . . . 5
  - 2.4.4 Sub-research Question 4 . . . 5

3 Theoretical Background 6
- 3.1 Employee Turnover . . . 6
- 3.2 Machine Learning . . . 7
  - 3.2.1 Models . . . 7
  - 3.2.2 Evaluation . . . 12
- 3.3 Model Interpretation . . . 15

4 Literature Review 16
- 4.1 Employee Turnover Cost . . . 16
- 4.2 Model & Feature Performance . . . 17
  - 4.2.1 Quantitative Review . . . 17
  - 4.2.2 Qualitative Review . . . 20
- 4.3 Literature Gaps . . . 21

5 Methodology **22**
5.1 Framework . . . 22
5.2 Business Understanding . . . 23
5.3 Data Understanding . . . 23
5.4 Data Preparation . . . 23
5.5 Modeling & Evaluation . . . 24
5.6 Deployment . . . 25

6 Business Understanding **26**
6.1 Employee Turnover Costs . . . 26
6.2 Employee Retention Costs . . . 26

7 Data Understanding & Data Preparation **27**
7.1 Data Description . . . 27
7.2 Data Validation . . . 30
7.3 Data Cleaning . . . 30
7.4 Feature Creation & Transformation . . . 31
7.5 Dataset Aggregation . . . 33
7.5.1 Rolling Window Aggregation . . . 34
7.5.2 Last Window Aggregation . . . 35

8 Modeling & Evaluation **37**
8.1 Dataset Selection . . . 37
8.1.1 Non-pandemic Datasets . . . 37
8.1.2 Pandemic Datasets . . . 38
8.1.3 Full Dataset Evaluation . . . 40
8.2 Feature Selection . . . 41
8.3 Hyperparameter Tuning & Model Selection . . . 42
8.4 Threshold Tuning . . . 43
8.5 Relating Model Performance to Turnover . . . 45

9 Interpretation **46**
9.1 Global Interpretation . . . 46
9.2 Individual Interpretation . . . 49

10 Conclusion **52**
10.1 Key Findings . . . 52
10.2 Relevance . . . 54
10.2.1 Scientific Contribution . . . 54
10.2.2 Company Relevance . . . 54
10.3 Ethical Considerations . . . 55
10.4 Limitations . . . 55
10.5 Future Research . . . 56

Bibliography **58**

Appendix A Questionnaire **61**
A.1 Questionnaire Turnover Cost . . . 61
A.2 Original Questionnaire Turnover Cost . . . 62

Appendix B Model Performance Benchmarks **63**
B.1 Non-pandemic dataset full results 63
B.2 Pandemic dataset full results 66
B.3 Full dataset full results 69
B.4 Feature selection full results 70
B.5 Hyperparameter tuning full results 71
B.6 Threshold tuning full results 84

Chapter 1

Introduction

This thesis studies the cost of employee turnover and proposes a machine learning model to predict it. The purpose of this section is to provide a general scope of the research problem. Specifically, section 1.1 gives a motivation for the thesis, section 1.2 introduces the problem, section 1.3 gives a short company description, and lastly, section 1.4 identifies the contribution to the theoretical landscape.

1.1 Motivation

Employee turnover is the departure of people, and thus intellectual capital, from a company (Punnoose & Ajit, 2016). Employee turnover can be costly for companies, as they need to replace the employees that leave. To elaborate, turnover incurs separation costs, replacement costs and placing costs estimated at several thousand dollars per employee (McKinney et al., 2007).

For many companies this kind of employee behavior is a black box, due to the wide variety of reasons that can drive an employee to leave a company to work elsewhere. This thesis opens up this black box by using historical employee turnover data to train a model that can classify employees who are at risk of turnover based on their features. In addition to this model, employees' most important drivers for turnover are identified. With this information, the HR-department can intervene on a per employee basis, with a tailored intervention based on the results of the model. These interventions can lead to a healthier work environment in which employees are more satisfied and subsequently retained longer. Additionally, intellectual capital is retained and turnover costs are reduced.

1.2 Problem Introduction

Most companies are comprised of a large group of different people working together. Within this group, employees can decide to leave a company for a variety of reasons. This departure of intellectual capital is defined in the literature as employee turnover (Punnoose & Ajit, 2016), employee churn (Ma et al., 2019) and employee attrition (Fallucchi et al., 2020), with each definition roughly describing the same phenomenon. The difference between these three definitions is that turnover is defined as an employee leaving a company, and thereafter being replaced by another, as opposed to attrition and churn, that only describe the departure of an employee (Sisodia et al., 2017). Turnover, (as well as churn and attrition) can be split into two different subtypes, voluntary turnover and involuntary turnover. When voluntary turnover occurs, the decision to leave the company is made by the employee without any interference from the company. Involuntary turnover on the other hand occurs when the decision for the employee to depart the company is made by the company instead of the employee (Patel et al., 2020).

The focus of this thesis is on the problem of voluntary turnover. This decision is made for the following two reasons. Firstly, whenever an employee decides to leave the company a position

within a team becomes vacant. Generally, this vacant position needs to be filled with a new employee in order to maintain the size of the workforce, making this problem a turnover problem. Secondly, when an employee is fired from the company (involuntary turnover) the company is aware that this decision is going to be made. Therefore, they can take precautionary measures, limiting downtime, or decide not to rehire in the case of downsizing. Whenever an employee decides to leave by themselves (voluntary turnover), the company might be surprised and thus unprepared. In conclusion, the problem that is worthwhile to predict, is the problem of voluntary turnover. Therefore, in the following chapters, the usage of the terms 'turnover' and 'employee turnover' refer to voluntary employee turnover.

1.3 Company Description

The company is a large financial service provider operating worldwide, offering a wide variety of financial services. This company wishes to remain anonymous due to the sensitivity of the data and insights gained in this thesis. Therefore, the description of the company is kept as general as possible, without compromising the context for the thesis and the dataset.

The company wants to reduce their current turnover numbers and retain employees longer. To achieve this reduction, a project was formulated in which data from their CRM- and HR-system will be made available for the purpose of training and optimizing a machine learning model. Moreover, the HR- and IT-departments are closely involved with this project, since the project uses data and expertise from both departments. The project and research are conducted within the IT-department, as this department is tasked with the implementation of data driven innovations. Furthermore, supervisors from all departments within the company are available for questioning with surveys or interviews when necessary.

1.4 Contribution

In order to correctly position this thesis and its contents, it is important to consider its contributions to the literature. To start, this thesis has the following practical contributions:

Firstly, many of the recently published papers on this topic are based on similar public datasets as elaborated in section 4.2, as it is hard to obtain employee data due to its sensitivity. The advantage of this thesis is that it will provide new insights into model performance and feature importance due to its unique dataset.

Secondly, very little research on this topic has been done in the financial sector, a sector that employs many people around the world. Therefore, it is important that the employee turnover characteristics in this sector are explored. This thesis aims to enhance the current state of the art by providing new insights into turnover model performance and feature importance in this sector.

Lastly, as of last year the world has been in a state of global pandemic. The pandemic has changed the way companies operate, since working from home has become the standard. Very little research on turnover classification during a global pandemic and its influence on model performance and feature importance has been done. This thesis aims to compare a model trained on pandemic data and non-pandemic data, with as a goal to provide practical insights on difference in model performances.

In addition to these practical contributions, this thesis also has some theoretical contributions to the literature. Currently, papers written on employee turnover focus on a part of the problem. They discuss the costs associated with employee turnover or discuss suitable models and predictive features. Meaning that, these papers do not tackle the turnover problem as a whole. These literature gaps are discussed in more detail in section 4.3. This thesis aims to unify the cost, modeling and interpretation aspects into one holistic approach, thereby contributing a new unified methodology to the existing literature. Section 10.2.1 elaborates on this theoretical contribution. Moreover, the insights gained by the model are related to a cost reduction by looking at potential employee retention and cost savings related to that.

Chapter 2

Problem Statement

This chapter describes and elaborates on the research problem. Specifically, section 2.1 describes the problem in more detail, section 2.2 elaborates on the research goal, and lastly, section 2.3 states the main research question and section 2.4 its sub-research questions.

2.1 Detailed Problem Description

As discussed in section 1.2, voluntary employee turnover occurs when an employee decides to leave the company and thereafter has to be replaced. This departure could be caused by a wide variety of factors, such as outside job offers, location changes and working environment (Ma et al., 2019). Due to the great number of factors that can lead to turnover, it is important for a company to gain insight into who is at risk of turnover and what their reasons are. These insights can help the company to create individually tailored interventions for the employees at risk, consequently increasing retention.

Increasing retention is important as previous research shows current turnover rates (equation 3.1) in various sectors to be varying between 8.00% and 28.34% (Fallucchi et al., 2020; Patel et al., 2020; Zhao et al., 2019). Additionally, the costs of turnover can be significant, ranging from ~$2,600 to ~$23,000 per employee, depending on the department and job position of the employee (McKinney et al., 2007). For example, the total annual cost of turnover of a company with of an annual average workforce of 1,000 employees and turnover rate and cost of 8.00% and $2,600 can be $208,000.

Direct costs are not the only type of costs associated with employee turnover, indirect costs are also a part of this problem (Tziner & Birati, 1996). Employees leaving the company take a vast amount of knowledge with them, leaving the company with less intellectual capital (Punnoose & Ajit, 2016). In addition to the reduction of intellectual capital, the team that this employee used to work in is (temporarily) a person short which decreases morale, increases overwork and decreases production (Tziner & Birati, 1996). These indirect costs cannot exactly be measured, they can however, be estimated by supervisors that work closely with the team as demonstrated by McKinney et al. (2007).

2.2 Research Goal

The goal of this thesis is to provide the company with a model that can assist the HR-department in identifying employees at risk. This model is trained on historic employee data and is retrained when new data becomes available. Whenever an employee at risk of leaving the company is identified by the model, the HR-department and/or supervisor of this employee should be notified of this employee. In order to assist the HR-department, the model should also include the most likely reasons for this employee to leave the company. Subsequently, these reasons can be used to

create a tailored intervention that fits the case that has been identified. To illustrate, if overwork is identified as a driver for turnover, the HR-department/supervisor can look into the causes for this overwork and, subsequently, try to mitigate these causes. In summary, the most important features leading to the turnover classification should be identified and ranked by the model. These features can be used by the HR-department/supervisor as guidelines for where their potential intervention should be targeted.

2.3 Main Research Question

Three important components are identified with respect to the research problem stated as discussed in section 2.1. Firstly, turnover can be expensive and have a major financial impact on the company, therefore it is important to investigate the cost aspect of this problem. Secondly, voluntary turnover occurs unexpectedly, investigating the predictability of this phenomenon in practice can help the company to mitigate part of this problem. To illustrate, a predictive model can help the company to identify the employees that are at risk of leaving. Lastly, reasons for voluntary turnover are generally a black box for the company. Investigating the individual drivers for employees to leave the company identified by a classification model can help the company to react to the potential issues that these employees are having. These insights should help the company in carrying out a targeted intervention and mitigating a share of the voluntary turnover within the company. Therefore, the main research question is:

How to gain insights into future turnover and apply these insights to reduce employee turnover costs using historical employee turnover data?

2.4 Sub-research Questions

In order to answer this main research question several sub-questions need to be formulated and answered.

2.4.1 Sub-research Question 1

Every time an employee leaves, there is a certain cost involved, as the company has made some time and monetary investments into this employee. These costs can differ depending on the characteristics of the employee that is leaving. For example, an employee that is with the company a long time, will often have a high position in the hierarchy with a large sum of sunk costs incurred. Moreover, the company also spends money on retaining employees in order to prevent turnover. Thus, the first sub-research question is:

What are the current costs associated with employee turnover and employee retention?

2.4.2 Sub-research Question 2

Within the machine learning domain a wide variety of models exist for classification. In addition to these models, employees can be defined by a wide variety of features. In order to arrive at a viable scope for the models and features used in this thesis, it is important to review what other researchers have concluded in this area of research. Based on the findings of this literature review, the features and models need to be evaluated on their performance in the context of this problem, in order to select the model and features with the best performance. Subsequently, these models need to be interpreted, as it is important to identify individual drivers for turnover. Therefore, the second sub-research question is:

How to identify future turnover and its causes using machine learning techniques?

2.4.3 Sub-research Question 3

The world is currently in a global pandemic, meaning that the last year has been different from other years for most companies. A consequence of this pandemic could be that characteristics of the dataset have changed. Therefore, it is important to compare a baseline (non-pandemic) dataset to a non-baseline (pandemic) dataset to determine if there is a difference in model performance. Thus, the third sub-research question is:

How do models trained on pandemic data and non-pandemic data compare?

2.4.4 Sub-research Question 4

Identifying an employee that is at risk is important, but the identification will not stop the employee from leaving. Therefore it is important for the HR-department/supervisor to be able to intervene. This intervention has the highest chance of success when it is directly aimed at the main driver for turnover. Therefore, the gap between the model output and the people using this output needs to be bridged. Consequently, the fourth sub-research question is:

How can supervisors and the HR-department interpret the output of the model, such that it can be used for intervention strategies?

Chapter 3

Theoretical Background

This chapter provides a theoretical background that supports the concepts used in this thesis. Section 3.1 discusses the general concepts related to employee turnover, and section 3.2 introduces the general concept of machine learning. Moreover, section 3.2.1 describes each machine learning model and its inner working, and section 3.2.2 elaborates on and compares methods for the evaluation of these models. Lastly, section 3.3 discusses the concept of machine learning model interpretation and explains the inner workings of SHAP.

3.1 Employee Turnover

The basic concepts related to employee turnover have already been discussed in sections 1.2 & 2.1. This section describes and highlights some aspects of this concept that have not been discussed yet.

To start, it is important to highlight that voluntary employee turnover does not happen instantaneously. To elaborate, prior to leaving a company, an employee develops turnover intentions, defined by Tett and Meyer (1993) as an employee developing a conscious and deliberate willingness to leave a company. Moreover, Boswell et al. (2008) explain these intentions to be proximal and preceding to the turnover, implying that turnover is not instant but rather something that happens over time.

Furthermore, an important metric to measure employee turnover is the employee turnover rate. The employee turnover rate is defined as the percentage of turnover with respect to the entire average workforce in a year (equation 3.1). This metric can be used for different timespans than a year and different employees can be excluded or included. For this thesis, only non-temporary employees are considered for the average workforce and only voluntary turnover within this workforce is considered for the turnover figure.

$$Employee\ Turnover\ Rate = \frac{Voluntary\ turnover}{Workforce_{end\ year} - Workforce_{start\ year}} \cdot 100\% \tag{3.1}$$

Lastly, the polar opposite of employee turnover is employee retention, which is defined in the literature as the encouragement of employees to remain with the company for a maximum period of time (Das & Baruah, 2013). This definition implies that an active effort is made by the company to keep an employee within the company. This concept is important for this thesis, as the insights gained into employee turnover can actively be used for retention strategies and interventions on a global and individual level.

3.2 Machine Learning

Machine learning is the practice of using mathematical algorithms (models) to make sense of large datasets. These models are trained to find patterns in the datasets they are applied to and, subsequently, use these discovered patterns for predictions on new samples presented to the model. These algorithms find these patterns by learning rules from the data, as opposed to conventional techniques, that have certain user defined rules on which decisions are made. Because of this property, machine learning models can make sense of large noisy datasets, which have no clear patterns on which rules can be defined (Witten et al., 2011). The employee turnover dataset provided by the company for this thesis shares these properties, as the dataset has thousands of entries, is partly made up from manual input data, and might contain complex interactions between features. These properties make machine learning methods suitable for the analysis of this dataset. Additionally, machine learning has been applied successfully to this problem in past research by, amongst others, Punnoose and Ajit (2016), Zhao et al. (2019) and El-Rayes et al. (2020).

Several different machine learning tasks exist, such as supervised learning, unsupervised learning and reinforced learning. Specifically, supervised machine learning is a machine learning technique where the model is tasked with relating several input features to one or more target feature(s), which is the goal of this thesis. Moreover, the nature of these predictions can be a quantity (regression) or a label (classification). Since the goal of this thesis is labeling employees on whether they will (1) or will not (0) leave the company, the machine learning task discussed in this thesis is a binary classification problem (Gao et al., 2019).

3.2.1 Models

In recent years, numerous machine learning models have been developed, adapted and used in practice. In order to narrow the scope of the models that are appropriate to use for the business case presented in this thesis, various machine learning models and their merits are evaluated and discussed in the literature review in section 4.2. The theoretical framework on which these methods are based is presented in this section.

Decision Tree

A very basic classifying algorithm is the decision tree, a simple and easy to interpret algorithm, that forms the basis for many advanced machine learning models. The ideas presented in this section are based on research by Rokach and Maimon (2005).

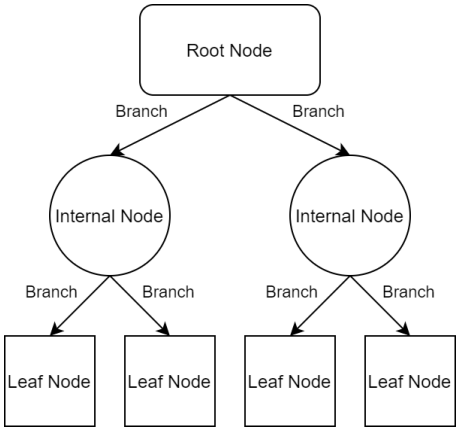
A decision tree is characterised by a number of nodes with branches between them. Specifically, a root node at the top of the tree, several internal nodes below the root node and, lastly, the leaves at the bottom of the tree. A visual representation of a basic decision tree is shown in Figure 3.1.

A decision tree is automatically constructed from a dataset in the following manner: First, the root node is constructed by calculating the split that results in the lowest impurity within each split. Specifically, the impurity in the resulting two nodes is low, when the parent node creates a clean split between the classes that have to be classified by the model. Several different impurity measures can be used to compute the impurity, such as the Gini index (Breiman et al., 1984). The general equation for the Gini Index, also known as the Gini Impurity, is shown in equation 3.2 below. In this equation, C is the set of classes in the classification problem (0 and 1 for this thesis) and P_i is the probability of classifying a sample within this class.

$$Gini\ Index = 1 - \sum_{i=0}^{C} (P_i)^2 \tag{3.2}$$

Since the Gini Index is calculated for both sides of the split, the Gini Gain is calculated to find the total impurity for making this split. This calculation is done by taking the weighted Gini Index of both sides of the split, as shown in equation 3.3. In this equation S_{side} is the total number

Figure 3.1
Visual representation of a basic decision tree



Note. Adapted from Sá et al. (2016)

of samples in the left (L) and right (R) side of the split, and GI_{side} is the Gini Index in the left (L) and right (R) side of the split.

$$Gini\ Gain = \frac{S_L}{S_L + S_R} * GI_L + \frac{S_R}{S_L + S_R} * GI_R \tag{3.3}$$

The feature that results in the lowest impurity is chosen as the root node (first split). Subsequently, this process is repeated for each internal node in the tree with the remaining features. A branch in the tree is terminated whenever both splits have a higher impurity than the parent node. Consequently, this internal node is not split further and becomes a leaf node, the last node in a certain branch in the tree. This process is repeated until no more splits can be made without increasing the impurity for the next split. Ultimately, the termination of this algorithm means that the final version of the decision tree has been constructed.

Random Forest

An often used technique in machine learning, is the practice of using ensemble models. These models make use of multiple (weak) machine learning algorithms of which the outputs are aggregated into a final prediction. The random forest model is one of these ensemble models, introduced in a paper written by Breiman (2001) and used as a basis for the ideas presented in this section. A random forest is based on an ensemble of many decision trees that are trained and combined as follows:

As a first step, samples are taken from the dataset that is provided as train data for the random forest model. Each sample is chosen at random and with replacement, meaning that one sample can be chosen multiple times. This process of sampling is called bootstrapping, consequently, creating a bootstrapped dataset. Thereafter, n (n is a user defined parameter) features are selected at random from the bootstrapped dataset, which are considered for the root node of the decision tree. Subsequently, the feature with the lowest impurity is chosen, and a second set of n features is selected from the remaining features for the following split. This process is repeated until a final decision tree is built on the bootstrapped dataset. Because this is an ensemble method, the process of building a decision tree on a bootstrapped dataset is repeated, until a user specified number of trees is reached. Lastly, when making predictions, each tree votes on the class that they predict the sample to belong in. The final predicted class is determined with a majority vote from the decision trees that have been created in the modeling process, also known as bagging.

Gradient Boosting Tree

Bagging is not the only ensemble method that combines multiple weak learners into one aggregated prediction. Another method that is often used is the boosting principle. Whereas in the aforementioned bagging ensemble method the models are used in parallel (all models vote together on the final output), in boosting, the weak learners are used in a sequential manner. Specifically, each tree used to make classifications is weighed and its output is used as a basis for the next tree that is created. One of the models using this boosting principle is the gradient boosting tree model, introduced by Friedman (2001) in his paper and used as a basis for this section. The gradient boosting tree algorithm works as follows:

As a starting point, the algorithm is initialized by calculating the average value of the target variable, which is used as an initial prediction for each entry in the training dataset. Subsequently, the residuals between the individual entries in the dataset and the initial prediction are calculated. Thereafter, a decision tree is built that tries to predict these residuals for each data point in the training data. These predictions are then multiplied with a user defined learning rate, after which they are added to the initial average prediction, resulting in a new prediction for each sample. Using these new predictions, the new residuals are calculated, a tree is built on those residuals and the new outputs are multiplied with the learning rate and added to the previous prediction. This process repeats itself until either a user specified number of trees has been reached, or the residuals show no signs of improvement in new iterations of the algorithm.

Extreme Gradient Boosting

An algorithm that is built further on the ideas presented in the previously described gradient boosting tree is the extreme gradient boosting model. Specifically, this model uses the boosting technique to sequentially construct and weigh decision trees, aggregating them into a final prediction. The difference between this model and the gradient boosting tree model mainly manifests itself in the way that the individual decision trees are constructed. Chen and Guestrin (2016) explain the inner workings of this model in detail, the ideas presented in their paper are used as a basis for this section. Specifically, the algorithm is implemented as follows:

Just as the gradient boosting tree algorithm, this model needs a starting point for each sample from which it starts working towards the final classification. Whereas the gradient boosting tree model uses the average of the target feature in the training dataset, the extreme gradient boosting model initializes its predictions at 0.5 for each individual prediction. Subsequently, the residuals between this prediction and the target feature are calculated for each sample in the training dataset. Similar to the gradient boosting model, a decision tree is built, which tries to predict the residuals.

However, the manner in which this decision tree is constructed is different from the conventional way of constructing a decision tree. Specifically, as a first step, all residuals are pooled together in the root node of the decision tree. Thereafter, a quality score describing the similarity of the classes within the node is calculated. In order to create more branches within this decision tree, quality scores for each feature and threshold on which a possible split can be made are calculated. Based on the quality scores of the left and right side of a split, a gain score is calculated, a score that indicates the potential gain that is achieved when adding this branch. Out of all feature-threshold combinations, the combination with the highest gain score is selected as the next branch. This process is repeated until no more splits can be made, or when a user set cover parameter is reached and thus the final decision tree is constructed. The cover parameter determines the minimum number of data points that should end up in a leaf. Ultimately, when the tree is complete, the leafs are pruned (removed) based on their gain score. Whenever a node splitting the data has a gain score below a user defined threshold, it is pruned from the decision tree.

After the construction of the decision tree, the algorithm is similar to the gradient boosting tree algorithm again. New predictions based on the previously constructed decision tree are obtained, multiplied with a user defined learning rate, and added to the initial prediction. Subsequently, the new residuals are calculated, and a new tree is constructed on these residuals. Thereafter,

new predictions are obtained from the decision tree, multiplied with the learning rate and, lastly, added to the prediction from the previous iteration. This process is repeated until either a user specified number of trees has been reached, or the residuals show no signs of improvement in new iterations of the algorithm.

AdaBoost

A different approach to boosting is the AdaBoost algorithm proposed by Freund and Schapire (1999) on which this section is based. AdaBoost uses multiple weak learners in the form of decision stumps. Decision stumps are a simplified version of a decision tree, which have one node and two leaves. This algorithm works as follows:

The algorithm is initialized by giving each sample a weight that shows how important it is to correctly classify this sample. In the first iteration of the algorithm, each sample gets the same weight, which is calculated by dividing one by the total number of samples in the training data. Subsequently, stumps are created for each feature and threshold combination in the dataset. Thereafter, the Gini Indexes (equation 3.2) and Gini Gains (equation 3.3) for each stump are calculated. From this collection of stumps, the stump with the lowest impurity (Gini Gain) is selected. Since a stump is a weak learner, which generally does not classify all samples correctly, its error and thus importance in the final classification (α_t) needs to be evaluated. The error of a stump is equal to the sum of the weights (ϵ_t) of the samples that have been wrongly classified by the stump (equation 3.4).

$$\alpha_t = \frac{1}{2} \ln(\frac{1 - \epsilon_t}{\epsilon_t}) \tag{3.4}$$

As a next step, the weights of the samples in the dataset are updated accordingly. Correctly classified samples are scaled by multiplying the original weight with $e^{-\alpha_t}$ and incorrectly classified samples are scaled by multiplying the original weight with e^{α_t}. Thereafter, the weights are normalized so that they add up to one, thus, representing the new weight of the samples. Subsequently, new stumps are created and evaluated in a slightly modified way from the first iteration of the model. The stumps can be evaluated in two ways. Firstly, they can be evaluated on a weighted version of the Gini Gain, where the previously calculated weight is taken into account in the Gain Index. Secondly, they can be evaluated on a newly sampled dataset, where samples are taken from the original dataset and each sample has a chance to be sampled equal to its weight. Based on either of the two methods, the best performing stump is selected and its importance is calculated again. Thereafter, the weights are updated and normalized again and new stumps are evaluated and chosen. This process is repeated until a user defined number of stumps is reached.

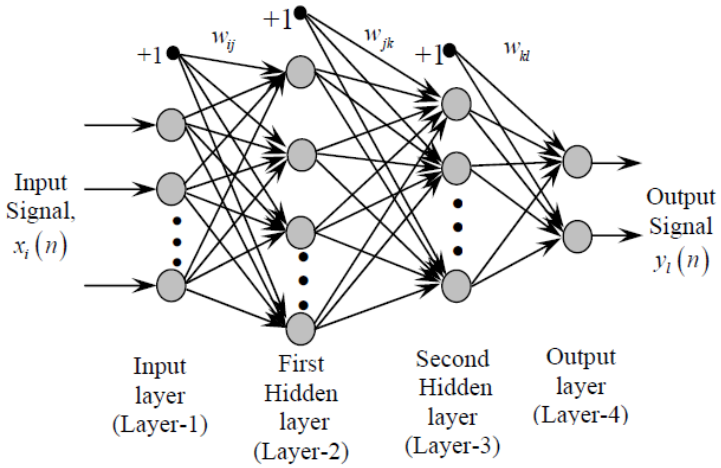
After using the algorithm to create a forest of stumps, the final classifications are made by evaluating the classifications made by the individual stumps for each sample. Specifically, the importances of the individual stumps that predict the same class for the sample are added up. The class with the highest summed importance, is the class that the sample is classified in.

Multilayer Perceptron

The models discussed in the previous sections all make use of some adaptation of a decision tree to compute their final classifications. However, machine learning models based on different underlying mechanisms exist as well. One of these models is the multilayer perceptron, which makes use of simple linear interactions between nodes in a network, combined with an activation function. Witten et al. (2011) describe this model in detail in their book, on which the ideas presented in this section are based. The multilayer perceptron algorithm works as follows:

A multilayer perceptron is a network of nodes that are interconnected, a visual representation of a general version of this network is shown in Figure 3.2. This network takes the values of different input signals, such as the values of features in a dataset, and applies various mathematical manipulations to them to arrive at a final prediction. The first manipulation in the network is a multiplication of the input values with different weights, indicated in Figure 3.2 with w_{ij}, and

Figure 3.2
The schematic of a multilayer perceptron



Note. Image taken from Mishra (2015)

the addition of a bias, indicated in Figure 3.2 with +1. Thereafter, the input is manipulated in a hidden layer, which applies an activation function (equations 3.5 to 3.8) to the aforementioned manipulated input values. Depending on the network, the resulting values are manipulated for a user defined number of times in a similar fashion, after which they are added into a number of resulting output values. For this thesis, that deals with a binary classification problem, there are two output values.

$$Identity: \quad f(x) = x \tag{3.5}$$

$$Sigmoid: \quad f(x) = \frac{1}{1+e^{-x}} \tag{3.6}$$

$$Tanh: \quad f(x) = tanh(x) \tag{3.7}$$

$$ReLU: \quad f(x) = max(0, x) \tag{3.8}$$

Now that the means of prediction of a multilayer perceptron is clear, a closer look is taken at the methods it uses to approximate the desired output values as closely as possible. To start, the network is initialized with random weights, indicated in Figure 3.2 with w_{ij}, w_{jk} and w_{kl}. Thereafter, the network makes its initial predictions as described in the previous section, after which backpropagation is applied to the model. Backpropagation is a means of evaluating the prediction errors between the desired and the actual output of the model. These errors are evaluated from the output node(s) (the back of the network) back towards the input node(s) (front of the network). An optimization algorithm often used in backpropagation is gradient descent. Specifically, gradient descent calculates the gradient of the error function at each manipulation in the network. Since this method makes use of backpropagation, it starts at the output of the network and works its way up to the input of the network. These gradients of the error function indicate in which way the weights need to be adjusted in order to lower the error. A positive gradient indicates a negative adjustment of the weight, whereas a negative gradient indicates a positive adjustment of the weights. In the next step, each weight is updated by, adding to or subtracting of, a user defined learning rate to the current weight based on the aforementioned sign of the gradient. This process is repeated until a user defined number of iteration is reached, or the error shows no more sign of improvement.

3.2.2 Evaluation

Due to the different inner workings of the models described in the previous section, each model has a different performance in different situations. One model might work better for a certain kind of dataset, whereas another model might excel when another dataset is used. Consequently, model performance differs depending on the problem they are applied to. The merits of various models used for this problem are discussed in section 4.2, however, multiple models are found to be suitable for the problem of employee turnover. When comparing the performance of these models in this specific case, some performance benchmarks need to be defined. The specific performance benchmarks chosen to evaluate these models are discussed in this section.

Confusion Matrix

Evaluating the performance of a binary classification model can be done with several performance measures. However, before these measures can be calculated, a confusion matrix needs to be constructed. A general representation of the confusion matrix is shown in Figure 3.3.

Figure 3.3
Confusion matrix

		Predicted class	
		Negative	**Positive**
Actual class	**Negative**	True negatives (TN)	False positives (FP)
	Positive	False negatives (FN)	True positives (TP)

This matrix is built up as follows. Whenever the model correctly classifies an entry in the negative class, this is registered in the matrix as a true negative (TN). Additionally, whenever the model correctly classifies an entry in the positive class, it is registered as a true positive (TP). Moreover, whenever the model makes a wrong classification in either of the classes, these are registered as false predictions in the matrix. Specifically, whenever the model incorrectly classifies an entry in the positive class, it is registered as a false positive (FP), whereas an incorrectly classified entry in the negative class is identified as a false negative (FN) (Baldi et al., 2000). In the context of this thesis these classifications can be summarized as follows:

- TN: The total number of employees that stay with the company (0), that are identified as a non-turnover (0) by the classification model.
- TP: The total number of employees that leave the company (1), that are identified as a turnover (1) by the classification model.
- FP: The total number of employees that stay with the company (0), that are identified as a turnover (1) by the classification model.
- FN: The total number of employees that leave the company (1), that are identified as a non-turnover (0) by the classification model.

With just these four different scoring categories, various performance measures can be calculated. The performance measures used in this thesis are the *accuracy*, *precision*, *recall*, *f1 score*

and the *area under the precision recall curve*. These performance measures are explained in detail and the usage thereof is justified in the following sections.

Accuracy

The first benchmark that can be derived from the confusion matrix is the accuracy. This metric is defined as the correctly classified samples as a fraction of the total number of samples presented to the model. The mathematical representation of this metric is shown in equation 3.9.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.9}$$

The accuracy performance benchmark is included for completeness, as this is, generally, an often used metric in machine learning research (Baldi et al., 2000). However, not much importance is placed into the accuracy score, as accuracy generally misrepresents the intricate nature of an imbalanced dataset (He & Garcia, 2009). To illustrate, when a dataset has a ratio of 99 negative cases to 1 positive case, a naive classifier always predicting the negative cases will achieve an accuracy of 99% while providing no practical insights. The interesting positive cases are never identified and, therefore, the model has no predictive power while the accuracy is nearly perfect.

Precision

The second benchmark that can be derived from the confusion matrix, is the precision. This metric is defined as the fraction of correctly classified positive (1) cases out of the total number of positive cases presented to the model. Specifically, the precision gives an insight into the model performance with regard to its false positives, meaning that a low precision indicates that the model tends to wrongly classify negative (0) samples as positive (1). The mathematical representation of this metric is shown in equation 3.10.

$$Precision = \frac{TP}{TP + FP} \tag{3.10}$$

The precision performance benchmark is important, as it partly shows how the model performs in the classification of an imbalanced dataset (He & Garcia, 2009).

Recall

The third benchmark that can be derived from the confusion matrix, is the recall. This metric is defined as the fraction of correctly classified positive samples (1) out of all positive (1) samples presented to the model. In other words, this metric shows how good the model is at identifying the positive cases in the dataset. The mathematical representation of this metric is show in equation 3.11.

$$Recall = \frac{TP}{TP + FN} \tag{3.11}$$

Consequently, the recall metric is important in evaluating model performance on imbalanced data, and shares an inverse relationship with the precision score (He & Garcia, 2009).

F1 score

The fourth benchmark is a benchmark that takes the values of the precision and recall into account. Where the precision and recall scores only show part of the performance of the model on an imbalanced dataset, the F1 score aggregates these two metrics into one. The mathematical representation of this metric is shown in equation 3.12.

$$F1\ score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{3.12}$$

The idea behind the F1 score is that it solves the problem of only relying on the precision or recall score, as it accurately captures the trade off between these scores (Chawla, 2010). Consequently, the F1 score is an important metric when evaluating model performance on imbalanced data.

A generalized version of the F1 score is the F_β score, defined in equation 3.13. This generalized version uses a factor β that defines the importance of recall in terms of precision. To illustrate, if recall is deemed 5 times as important as the precision, $\beta = 5$. Note that $\beta = 1$, returns the normal F1 score.

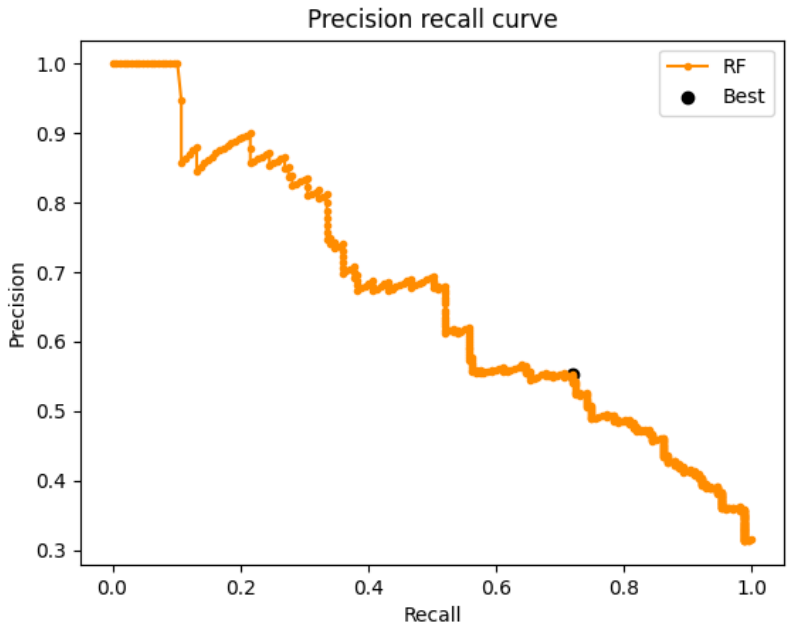
$$F_\beta \; score = (1 + \beta^2) \cdot \frac{Precision \cdot Recall}{(\beta^2 \cdot Precision) + Recall} \tag{3.13}$$

Area under the precision recall curve

The last benchmark used to evaluate model performance is the area under the precision recall curve. As discussed in the previous sections, precision and recall share an inverse relationship.

Classification models generally make predictions in the form of a probability. For binary classification problems, this probability describes the chance that a sample belongs to the positive class as opposed to the negative. This probability is used in combination with a threshold as follows. Whenever the probability of a sample belonging to the positive class is higher than the threshold, the sample is classified into the positive class. Conversely, when the probability is lower than the threshold it is classified into the negative class. Consequently, shifting the threshold from zero all the way up to one, will give different precision and recall values for each threshold. As shown in Figure 3.4, shifting the threshold results in a line representing the different trade-offs that can be made between the precision and the recall of the model. The orange line in the figure represents the different precision recall trade-offs.

Figure 3.4
A general example of a precision recall curve



Note. Precision-recall curve generated using a Random Forest.

Using the precision recall curve two important metrics can be derived. Firstly, the area under this curve (AUC_{Pr}) can be computed. A greater area under the precision recall curve indicates

that the model (after shifting the threshold) has a greater potential performance than another (He & Garcia, 2009). Secondly, the precision recall curve can be used to find the optimal threshold of a model by calculating the F1 score for each point on the curve. The threshold with the optimal trade-off is associated with the highest F1 score, indicated by the black dot in Figure 3.4.

A metric that is very similar to the precision recall curve is the Receiver Operating Characteristic Curve (ROC curve) combined with the area under ROC curve. The ROC is different than the precision recall curve because for the computation of this curve, precision and recall are replaced by sensitivity and specificity (Baldi et al., 2000). The ROC curve is not included as a performance metric in the model evaluation, because it is insensitive to imbalanced data (Saito & Rehmsmeier, 2015).

3.3 Model Interpretation

In certain situations, such as the one presented in this thesis, it is important to understand why a model makes certain predictions. This concept, defined as model interpretability, is straightforward when using basic models such as decision trees, where each decision is clearly represented by the split rules of the tree. However, when models capture more complex interactions between features, interpretation becomes harder and more advanced interpretation methods are needed. One of these advanced methods is SHAP (SHapley Additive exPlanations), an interpretation method proposed by Lundberg and Lee (2017) on which the explanations provided in this section are based. An argumentation for the usage of SHAP is given in section 5.6.

To start, SHAP starts out with a certain base rate, which is the expected value for the model output. SHAP explains the gap between this base rate and the final classification of a single sample in terms of feature contribution. The explanation is obtained as follows:

In order to determine the contribution of each individual feature, SHAP starts by using a single feature for classification. For this feature, the contribution to the base rate is found by evaluating the model output. To illustrate, consider a model with a base rate of 0.5. SHAP starts by including the feature *tenure*, which returns an output of 0.6 for this individual. This output implies that *tenure* contributes +0.1 to turnover for this single individual. Thereafter, a new feature is added to the classification, again evaluating the impact of this feature on the final output of the classification. For example, adding the *gender* feature changes the output of the classification to 0.45, implying that *gender* contributes -0.15 to the final prediction. This process is repeated by SHAP, adding one feature at a time, subsequently, evaluating its impact on the final classification. Note that the features and the corresponding numbers mentioned in this paragraph are fictitious and merely used to illustrate the concept.

In the aforementioned example, all features are added in a single (random) order, which poses problems when features have some form of interaction. To elaborate, when features interact, features that are introduced earlier in the process are given more credit to the final output of the model than their counterparts. In order to distribute the interaction effects fairly among all features, a game theoretical approach is applied. Specifically, a fair distribution of credit is achieved by averaging the contribution of each individual feature over all possible orderings of the features. Thus, the process described in the previous paragraph is repeated for each possible feature ordering after which the individual contributions are averaged.

The process described in this section obtains the feature contributions to the classification of a single sample. This process is repeated for each individual classification in the dataset, such that SHAP values are calculated for each feature for all samples. A property of SHAP values is that they are additive, meaning that individual insights can be added to obtain insights on a global or sub-class level. Therefore, the SHAP values obtained for each individual sample, can easily be manipulated for insights on different aggregation levels.

Chapter 4

Literature Review

This chapter discusses the findings of a literature review that is conduced to provide a starting point for this thesis. The literature review is divided into two separate parts. Section 4.1 describes a literature review conducted to find a suitable framework for employee turnover cost calculation. Section 4.2 discusses a literature review conducted to provide a basis on what machine learning models and what features have been found appropriate in previous research. Section 4.3 elaborates on the literature gap that is currently present in the literature.

4.1 Employee Turnover Cost

For the employee turnover cost calculations, various models in the literature are evaluated. Before discussing the various models that are considered, it is important to outline the criteria on which the selection is made.

The goal of the employee turnover cost calculation is to provide the company with an estimate of what the costs are of an employee leaving the company. To determine this estimate, it is important to select a cost framework that suits the business case presented in this thesis. Since, the business case will be judged by different departments with different mathematical backgrounds, the method to arrive at the final calculation should be clear and easy to follow.

Cascio (1991) proposes a model for turnover cost calculation that splits turnover costs into separation costs, replacement costs and training costs. This model has been adapted by Tziner and Birati (1996), as they argued that it is also important to take the indirect costs associated with employee turnover into account. These indirect costs arise in the form of morale loss, excess overtime pay and loss of production. Both models have their strengths and weaknesses, the model proposed by Cascio (1991) is strong in its simplicity but weaker in its completeness. On the other hand, the expanded model proposed by Tziner and Birati (1996) is strong in its completeness, but weaker in the fact that variables such as costs due to loss in morale, are hard to estimate. McKinney et al. (2007) address this problem by integrating the models proposed by Cascio (1991) and Tziner and Birati (1996) into a questionnaire with cost estimates that stakeholders can answer (Appendix A). The fact that this questionnaire is easy to interpret by stakeholders and that the cost figures are tangible, makes this model a suitable candidate for the turnover cost calculation in this thesis.

Tziner and Birati (1996) are not alone in their work on further developing the turnover cost calculation method proposed by Cascio (1991). Pinkovitz et al. (1997) and Hinkin and Tracey (2000) expand upon this method as well. Pinkovitz et al. (1997) apply the three turnover cost aspects (separation, replacement and training) in a calculation template that can be implemented by companies to estimate their cost of turnover. They identify the need for the implementation of indirect costs into the cost framework proposed by Cascio (1991), but, as opposed to Tziner and Birati (1996), conclude that these costs are too hard to estimate and subsequently do no implement this cost item in their calculation template. Hinkin and Tracey (2000) take it one step

further by applying the cost framework from Cascio (1991) to a real world problem, turnover in the hospitality industry. Specifically, they measure the turnover cost in two different hotels using the aforementioned cost framework. Hinkin and Tracey (2000) mention the need for the measurement of indirect costs in their paper and, subsequently, partly account for this by implementing the loss of productivity into their measurements. This estimation focuses on the productivity of the new employee and the fact that they still have to master their craft. Productivity loss for the other team members, due to reduced motivation or morale is not accounted for.

Research conducted by Mitrovska and Eftimov (2016) builds further on the aforementioned framework proposed by Cascio (1991) and further developed framework by Hinkin and Tracey (2000). Specifically, they make use of the three main cost aspects and the loss of productivity caused by having a new and inexperienced employee in the company. In addition to these costs, they identify the same costs that are identified by Tziner and Birati (1996), loss of productivity among other team members due to loss of motivation and morale. With the addition of these costs, the model they use in their research is closely related to the model used by McKinney et al. (2007), since they make use of the same costs types. In doing so, Mitrovska and Eftimov (2016) indirectly validate the research conducted by McKinney et al. (2007).

4.2 Model & Feature Performance

The literature review on model and feature performance is used to create a baseline on model and feature performance in previous research. First, a global overview of model and feature usage and performance is presented, after which a selection of papers is discussed in more detail.

4.2.1 Quantitative Review

To start, relevant articles found in the literature search are encoded to create a broader picture of the current progress in this area of research. Each machine learning model used in the articles is rated in their performance compared to the other models used in the same paper. The encoding categories *Good*, *Average* and *Bad* are used so that models can be compared between papers. Moreover, each performance category is based on the insights provided by the authors of the papers.

After the encoding of all papers, a score is assigned to each aforementioned performance category. *Good* is awarded with three points, *Average* with two points and *Bad* with one point. Those scores are added and, thereafter, divided by the amount of papers that scored this model, providing a list of weighed averaged scores for each model.

The results of model performance in the different papers are presented in Table 4.1. In this Table, the findings with respect to model performance are summarized using a score of G (Good), A (Average), B (Bad) and - (Not used). In the last column, the aggregated scores based on the aforementioned scoring method are shown for each model.

Now that the performance of each model in different papers is known and scored, Table 4.1 can be used for model selection. To start, models with an overall score below 2.0 are not considered, as they perform below average. Models with a score equal to 2.0 and below 2.5, should have appeared in three or more papers, so that the lower score is based on enough data. Lastly, all papers scoring higher than 2.5 are included when they have been discussed in 2 or more papers. Due to the high scores of these models, a more lenient approach is taken towards the number of occurrences in the literature.

Applying the selection based on the aforementioned thresholds results in six models that satisfy all constraints. These models are the *Decision Tree*, *Random Forest*, *Gradient Boosting Tree*, *Extreme Gradient Boosting*, *AdaBoost* and *Multilayer Perceptron* classification models, which are italicized in Table 4.1. Since these results are based on a quantitative analysis of papers, of which the scores are a result of interpretation, the resulting models are verified with a qualitative analysis in section 4.2.2.

Table 4.1
Results of the quantitative literature review on model performance

Model	Punnoose and Ajit (2016)	Zhao et al. (2019)	El-Rayes et al. (2020)	Esmaieeli Sikaroudi et al. (2015)	Khera and Divya (2019)	Sisodia et al. (2017)	Gao et al. (2019)	Monisaa Tharani and Vivek Raj (2020)	Ma et al. (2019)	Fallucchi et al. (2020)	Yadav et al. (2018)	Gabrani and Kwatra (2018)	Vasa and Masrani (2019)	Jain et al. (2020)	Patel et al. (2020)	**Score**
Extreme Gradient Boosting	G	G	-	-	-	-	-	G	-	-	-	-	-	-	-	3.0
Gradient Boosting Tree	-	G	G	-	-	-	-	-	-	-	-	-	-	-	-	3.0
Weighted Quadratic Random Forest	-	-	-	-	-	-	G	-	-	-	-	-	-	-	-	3.0
Alternating Decision Tree	-	-	-	-	-	-	-	-	G	-	-	-	-	-	-	3.0
Random Forest	G	G	G	G	-	G	G	A	G	B	G	G	G	G	G	2.8
Decision Tree	-	A	G	-	-	G	B	-	A	A	G	G	A	A	B	2.2
Multilayer Perceptron	-	A	-	A	-	-	-	A	A	-	-	-	-	-	-	2.0
AdaBoost	-	-	-	-	-	-	-	-	-	-	A	A	A	-	-	2.0
Linear Regression	-	-	A	-	-	-	-	-	-	-	-	-	-	-	-	2.0
Classification and Regression Tree	-	-	-	A	-	-	-	-	A	-	-	-	-	-	-	2.0
Naïve Bayes	A	B	-	G	-	B	B	B	B	G	-	-	-	-	-	1.6
Linear Discriminant Analysis	A	B	-	-	-	-	-	-	-	-	-	-	-	-	-	1.5
Logistic Regression	A	B	B	-	-	-	A	-	-	A	B	B	B	-	-	1.4
K-Nearest Neighbor	B	B	-	B	-	A	-	A	-	B	-	-	-	-	A	1.4
Support Vector Machine	B	B	-	B	A	B	-	A	-	B	A	-	-	B	A	1.4
Probabilistic Neural Network	-	-	-	B	-	-	-	-	-	-	-	-	-	-	-	1.0

The aforementioned literature not only discusses model performance, but also gives an indication on feature performance. Therefore, the findings in these papers with respect to feature performance are encoded in a similar manner as the model performance. Specifically, features are rated as *Good*, *Average*, *Bad* and *Not indicated* and scores are assigned to each of these performance categories. *Good* is awarded with three points, *Average* and *Not indicated* with two points and *Bad* with one point. Those scores are added and, thereafter, divided by the amount of papers that scored this feature, providing a list of weighed averaged scores for each feature.

The results of the feature usage in the different papers are presented in Table 4.2. In this Table, the findings with respect to feature performance of each paper are summarized using the same scores as for the model performance analysis, with the addition of X (Used but no performance indication given). In the last column, the aggregated scores based on the aforementioned scoring method are shown for each feature.

Table 4.2 shows a list of features included in previous research, hierarchically ordered by their scores. This list is used as a suggestion to the company for important features, where the priority of a feature is indicated by its score. The features in Table 4.2 are not exhaustive and extra

Table 4.2
Results of the quantitative literature review on feature performance

Model	Punnoose and Ajit (2016)	Zhao et al. (2019)	El-Rayes et al. (2020)	Esmaieeli Sikaroudi et al. (2015)	Khera and Divya (2019)	Sisodia et al. (2017)	Gao et al. (2019)	Monisaa Tharani and Vivek Raj (2020)	Ma et al. (2019)	Fallucchi et al. (2020)	Yadav et al. (2018)	Gabrani and Kwatra (2018)	Vasa and Masrani (2019)	Jain et al. (2020)	Patel et al. (2020)	**Score**
Last pay raise	-	G	-	-	-	-	-	-	-	-	-	-	-	-	-	3.0
Legal knowledge	-	-	-	G	-	-	-	-	-	-	-	-	-	-	-	3.0
Technical skills	-	-	-	G	-	-	-	-	-	-	-	-	-	-	-	3.0
Alternative Job Opportunity	-	-	-	-	-	-	-	G	-	-	-	-	-	-	-	3.0
Overtime	-	-	-	-	X	-	G	-	-	G	-	-	-	-	-	2.7
Professional Career Length	-	-	-	G	-	-	A	-	-	G	-	-	-	-	-	2.7
Job Satisfaction	X	-	-	G	-	X	B	G	X	X	X	G	X	G	-	2.3
Distance from home	-	-	-	-	X	-	A	-	-	G	-	-	-	-	X	2.3
Tenure	X	G	G	A	X	X	A	-	-	G	X	G	-	A	X	2.3
Salary	X	A	G	-	X	G	G	-	-	G	X	X	X	B	X	2.3
Number of projects worked on	-	-	-	-	-	X	-	-	-	-	X	X	X	G	-	2.2
Age	X	G	-	B	X	-	A	-	-	G	-	-	-	-	X	2.1
Working hours	-	-	-	-	X	G	A	-	-	-	X	X	X	A	-	2.1
Education	X	A	-	A	X	-	B	G	-	X	-	-	-	-	-	2.0
Service Line	-	A	-	-	-	-	-	-	-	-	-	-	-	-	-	2.0
Percentage Salary Increase	-	-	-	-	X	-	A	-	-	A	-	-	-	-	-	2.0
Working Conditions/team	X	A	-	-	-	-	-	-	-	-	-	-	-	-	-	2.0
Employee's perception of fairness	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2.0
Supervision	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2.0
Burnout	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2.0
Time since last promotion	X	-	-	-	X	G	A	-	-	A	X	X	-	B	X	2.0
Specialized Area	-	A	-	-	-	-	-	-	-	-	-	-	-	-	-	2.0
Total industry experience	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	2.0
Amount of training	-	-	-	-	X	-	A	-	-	X	-	-	-	-	X	2.0
Job Stress	-	-	-	-	-	-	-	A	-	-	-	-	-	-	-	2.0
Attitude towards Covid	-	-	-	-	-	-	-	A	-	-	-	-	-	-	-	2.0
Performance Rating	-	-	-	-	X	X	B	-	-	X	X	G	X	A	X	2.0
Marital Status	X	-	-	B	X	-	B	G	-	-	-	-	-	-	-	1.8
Number of companies worked before	-	-	-	G	B	-	B	-	-	X	-	-	-	-	X	1.8
Department	-	A	-	-	X	-	B	-	-	-	X	X	X	B	-	1.7
Job/Management Level	-	B	-	-	X	-	-	-	-	-	-	-	-	-	X	1.7
Ethnicity	X	B	-	-	-	-	-	-	-	-	-	-	-	-	-	1.5
Gender	X	B	-	-	B	-	B	A	-	-	-	-	-	-	-	1.4
Title/Role	-	B	-	-	X	-	B	B	-	X	-	-	-	-	-	1.4
Has children	-	-	-	-	-	-	B	-	-	-	-	-	-	-	-	1.0
Travel distance for business purposes	-	-	-	-	B	-	-	-	-	-	-	-	-	-	-	1.0

features deemed important can be added. Since the review is based on a quantitative analysis of papers, of which the scores are a result of interpretation, a closer look is taken at some features in the qualitative analysis in section 4.2.2.

4.2.2 Qualitative Review

Punnoose and Ajit (2016) demonstrate seven machine learning methods that are trained on an employee turnover dataset, after which their performance is evaluated. They conclude that tree based models are most suitable for this type of problem and, subsequently, identify the *Random Forest* and *Extreme Gradient Boosting* models as best performing models. The features used to train the model are briefly mentioned, however, no specific feature performance is discussed.

El-Rayes et al. (2020) further evaluate the performance of the aforementioned tree based models. Specifically, the authors compare the performance of three tree base models (*Decision Tree*, *Random Forest* and *Gradient Boosting Tree*) against *Linear-* and *Logistic Regression* models. They confirm the superior performance of these tree based models for the turnover classification problem. In addition to the model performance, the features *Tenure* and *Salary* are indicated as good predictors in the evaluated models. These two features are also used by Punnoose and Ajit (2016), where no feature performance indication is given.

Zhao et al. (2019) evaluate another important aspect of analyzing turnover data, which is the size of the dataset. In this paper, two datasets are split into multiple smaller datasets ranging from 50 entries up to and including 9,000 entries. Subsequently, these datasets are used as training data for nine machine learning models. They conclude that for smaller datasets ($\leq$ 100 entries), the best performing model differs per dataset, due to the large variance present in datasets of this size. When datasets become larger ($\geq$ 1,000 entries), the best performing models start to converge towards tree based models (*Decision Tree*, *Random Forest*, *Gradient Boosting Tree* and *Extreme Gradient Boosting*) and the *Multilayer Perceptron* model. Zhao et al. (2019) also show the feature importance in the *Extreme Gradient Boosting* model trained on a dataset of size 1,000. In this model the features *Age*, *Tenure* and *Last Pay Raise* have the greatest importance. *Tenure* is also used by Punnoose and Ajit (2016) and evaluated by El-Rayes et al. (2020), in which it is also found to be a good predictor for turnover. *Age* is only used by Punnoose and Ajit (2016) without indication of importance.

Monisaa Tharani and Vivek Raj (2020) compared the performance of six machine learning models in the IT industry. They conclude that *Extreme Gradient Boosting* is the best performing machine learning model out of the six evaluated. This finding is consistent with the literature, as Punnoose and Ajit (2016) and Zhao et al. (2019) arrive at a similar conclusion in their papers. Monisaa Tharani and Vivek Raj (2020) also rate the *Random Forest*, *Multilayer Perceptron*, *Support Vector Machine* and *K-Nearest Neighbour* as good performing models. These findings are partly consistent with the aforementioned literature, since *Random Fores* and *Multilayer Perceptron* models are evaluated as good performing models by Zhao et al. (2019). However, *Support Vector Machine* and *K-Nearest Neighbour* models are judged as weak performing models by Punnoose and Ajit (2016), Zhao et al. (2019), Sisodia et al. (2017) and Fallucchi et al. (2020). In addition to the machine learning model evaluations, Monisaa Tharani and Vivek Raj (2020) also identify *Education*, *Marital Status*, *Job Satisfaction* and *Alternative Job Opportunity* as important features for the prediction of employee turnover. From these four features, *Education*, *Marital Status*, *Job Satisfaction* are also used in Punnoose and Ajit (2016), who did not indicate their importance.

As mentioned by El-Rayes et al. (2020), tree based models offer the strongest performance when dealing with the turnover prediction problem. In the previously discussed literature, one important tree based model has not yet been discussed, *AdaBoost*. This model is evaluated in research by Gabrani and Kwatra (2018) amongst others, such as the *Decision Tree*, *Random Forest* and *Logistic Regression* models. Gabrani and Kwatra (2018) conclude, that the best performing model is the *Random Forest* model, closely followed by the *Decision Tree* and *AdaBoost* models. *Logistic Regression* is found the be the worst performing model of the four considered models. This finding is consistent with the findings by Zhao et al. (2019) and El-Rayes et al. (2020). Gabrani

and Kwatra (2018) also conclude that the features *Job Satisfaction*, *Tenure* and *Evaluation* are good predictors of employee turnover.

In summary, tree based models are generally found to be the best performing models by a large majority of the papers. Specifically, these models are the *Decision Tree*, *Random Forest*, *Gradient Boosting Tree*, *Extreme Gradient Boosting* and *AdaBoost* classification models. In addition to these tree based models, the *Multilayer Perceptron* classification model is also used with moderate success in the literature. The features with a high predictive power identified in the papers have some overlap between papers, but are dependent on the features that were present in the dataset that was used for the research. *Job Satisfaction* was, for example, not present in every dataset and could therefore not be evaluated in every paper. An overview of the important features, as found by the previously discussed papers, is as follows: *Age*, *Tenure* and *Last Pay Raise Education*, *Marital Status*, *Alternative Job Opportunity*, *Job Satisfaction* and *Evaluation*.

As a final remark, it is important to note that many papers dealing with employee turnover are based on two public datasets. The first one is a dataset provided by Kaggle, as shown in Gabrani and Kwatra (2018), the second one is a synthetic dataset provided by IBM through Kaggle, as shown in Fallucchi et al. (2020). These datasets are used because it is difficult for researchers to obtain third party data due to its sensitivity. In this qualitative literature review papers that are reviewed do not share the same source data, as this could skew the insights to one or two particular models that deal well with those specific datasets.

To conclude, the following six models will be used in the modeling process based on the quantitative and qualitative literature reviews: the *Decision Tree*, *Random Forest*, *Gradient Boosting Tree*, *Extreme Gradient Boosting*, *AdaBoost* and *Multilayer Perceptron* models. Moreover, the features presented in Table 4.2 will be used as a recommendation for features to be included in the final dataset.

4.3 Literature Gaps

When comparing the papers discussed in this chapter, some gaps in the current literature become apparent, which are highlighted in this section.

Firstly, all papers that discuss applying machine learning to the turnover problem, never relate the model to actual turnover costs. The cost aspect of turnover is often mentioned, however, only for motivating the research and placing it in an overall context.

Secondly, papers optimizing machine learning models applied to the employee turnover problem only partly relate the optimal model back to the actual problem. Specifically, tangible benchmarks such as the correctly identified number of employees that are predicted as turnover are mentioned, however, insights on false positives, false negatives and true negatives are often not present. Moreover, the optimal model is never related to insights on an individual level.

Thirdly, researchers never use threshold tuning during model optimization. They often relate the model performance to either the AUC_{Pr} or the ROC-curve, however, no steps are taken to use this curve to arrive at an optimal threshold. Moreover, no papers have used the actual costs of turnover and retention to optimize the prediction threshold of a model.

Fourthly, as mentioned in section 1.4, no papers on employee turnover prediction using machine learning in the financial sector have been published. Moreover, many papers are based on the same public dataset, showing the need for more diversity in industry and source data in this research area.

Lastly, one paper mentions the impact of the pandemic on the employee turnover problem. However, this paper only takes the pandemic into account by making use of a feature that measures the attitude of an employee towards the pandemic. No distinction is made between data obtained during and outside the pandemic.

To conclude, many individual parts of the employee turnover problem have been researched, however, they have never been linked together into one holistic approach. Furthermore, research is currently limited to a handful of datasets and industries, restricting the generalizability of previous findings.

Chapter 5

Methodology

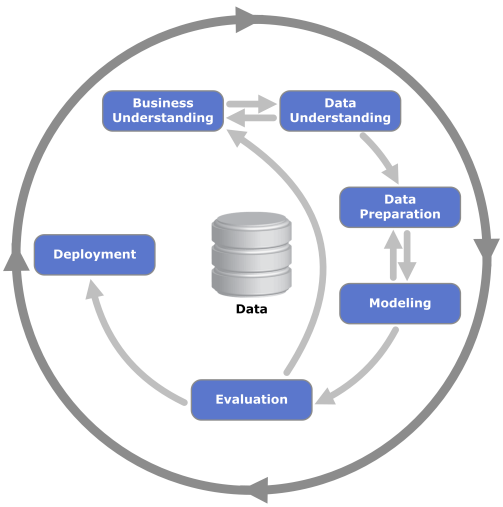
In order to obtain useful insights from data, many steps need to be taken. This chapter elaborates on the methodology of the thesis in a structured manner. Section 5.1 introduces the general framework on which the research is based. Sections 5.2 to 5.6 discuss each step in the framework in detail.

5.1 Framework

The framework adopted for this thesis is a widely used framework for solving data mining problems: the CRISP-DM (Cross Industry Standard Process for Data Mining) framework. This framework provides a structured approach for undertaking data mining problems, which makes it suitable for this thesis.

CRISP-DM divides the data mining problem into the following six main phases: *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modeling*, *Evaluation* and *Deployment* (Wirth & Hipp, 2000). An illustration of the six phases of the CRISP-DM methodology is shown in Figure 5.1 and the details of these phases and their relation to this thesis are outlined in their respective sections below.

Figure 5.1
Illustration of the six iterative phases of the CRISP-DM framework



Note. Source: Wirth and Hipp (2000)

5.2 Business Understanding

The business understanding phase is focused on defining the data mining problem from a business perspective. In this phase, the goals of the thesis are defined and the problem with relation to the company is explored (Wirth & Hipp, 2000). The costs and benefits of solving the employee turnover problem are defined in this phase, partly laying the foundation for answering sub-research question 1 (section 2.4.1).

In order to gain insight into the current turnover costs, the method proposed by McKinney et al. (2007), as discussed in section 4.1, is used. Specifically, a questionnaire is used to measure these costs based on the questionnaire proposed by McKinney et al. (2007) and shown in Appendix A.2. Since their questionnaire is developed for cost estimation within the government, some parts of it are redundant while others are incomplete. Therefore, some modifications are made in consultation with stakeholders within the company, the final questionnaire is shown in Appendix A.1. Within the questionnaire, supervisors are asked to estimate different cost types associated with turnover. Since these are estimates, it is important to gather enough data to reduce the variance in the replies.

The final questionnaire is send to supervisors who experienced voluntary turnover in the last 6 months. A period of 6 months is chosen for two reasons. Firstly, the analysis is powerful when it is built on many samples (McKinney et al., 2007). A shorter period would reduce the sample size, increasing variance. Moreover, the sample size is small even at a cutoff of 6 months, as only a small number of employees left in this period. Secondly, a longer period would result in more noise, as the supervisors' memory of the cases fades when time passes. Subsequently, the estimates in the survey are analysed and reported, resulting in the average cost per employee for the different cost types.

Additionally, a closer look is taken at the company's current expenditure with relation to employee retention. Specifically, costs related to education, promotion and other retention programs are elaborated.

5.3 Data Understanding

In the data understanding phase, the data is collected, described, and validated (Wirth & Hipp, 2000). To start, a relevant dataset is collected in collaboration with the company. Special attention is paid to the fact that the features included in the dataset are based on the important features found in section 4.2. Moreover, the feature set is enriched with features considered important by the HR-department, such as evaluation scores. It is important to note that the data collection process is constrained by the currently available data at the company, as it is not possible to gather new data on employees who already left the company.

Thereafter, a general overview of the dataset is provided, creating better understanding on how the entries and features within the dataset are built up. In this step, features that need to be elaborated on are highlighted and explained, so that a deeper understanding of the dataset and its features is gained. Lastly, an overview of all features within the dataset is given, together with their type and a brief description.

The dataset is a product of two datasets, originating from two different sources, that have been merged by the company. Moreover, some features within the dataset are a product of manual input. Therefore, the dataset needs to be validated. Specifically, the dataset is checked for missing values, empty string values, duplicate entries and outliers. All errors uncovered in this step, are resolved in the next phase of the CRISP-DM cycle.

5.4 Data Preparation

In the data preparation phase, the data is cleaned and features are created and transformed (Wirth & Hipp, 2000). As a first step, all errors identified in the data verification process are cleaned,

meaning that missing values and empty string values are imputed with a correct value or removed. Thereafter, the data is cleaned based on employee properties. Specifically, employees that left the company involuntary, students and contractors are removed from the dataset, as these employees fall outside of the scope of this thesis.

After cleaning the dataset, existing features are transformed and new ones are created from existing features, such that they can be used for machine learning. Two examples are, the transformation of categorical features into a one hot encoding and the creation of years passed from different dates. Moreover, the turnover feature is created determining if an employee is active or has left for each entry in the dataset. Features that are used as basis for the creation of new features are dropped, as they are no longer used in the rest of the process.

The dataset consists of multiple entries per employee, each entry covering roughly a month. Since employee turnover is a process that develops over a longer period of time (section 3.1), multiple periods are aggregated into multi-period entries. To achieve this aggregation, two contrasting approaches are employed, as it is not known which approach leads to the best model performance beforehand. Additionally, for both aggregation methods, the dataset is split into pandemic and non-pandemic data before aggregating the data.

The first method of aggregating the entries of employees, is a rolling window approach. In this approach a window of a certain size is placed over a subset of an employee's entries in the dataset, whereafter the training and target features within this window are calculated. These features are averaged, multiplied, or the last entry within the window is taken, depending on the properties of the individual features. Each window consists of a training window and a target window, of which the sizes can vary. The target window size determines how many steps ahead the forecast is made. The optimal sizes of these windows are unknown, as they depend on the properties of the dataset. Therefore, a method for determining these window sizes, as proposed by Inoue et al. (2017) is adapted. To elaborate, Inoue et al. (2017) introduce a method of comparing the effect of differing rolling window sizes on prediction performance, using linear regression. Instead of using linear regression as in the original method, machine learning is used to make classifications.

The second method of aggregating the entries of employees, is a last window approach. In this approach, only the last entries of each employee are aggregated, as opposed to a window that slides over each entry in the rolling aggregation approach. To illustrate, when forecasting one step ahead, the last 13 periods of data are used in the aggregation. One period for the target window and the other 12 for the training window. Moreover, the features within a window are aggregated differently. In the last window approach, individual feature values are divided into bins or are binary encoded.

To conclude, these two approaches are contrasting in the fact that, the first approach makes use of all entries in the dataset aggregated using the actual values of the features. Whereas, the second approach only makes use of the last entries of each employee, aggregated in an abstract manner.

5.5 Modeling & Evaluation

In the modeling phase, the relevant modeling techniques are selected, after which the models are trained (Wirth & Hipp, 2000). Additionally, in the evaluation phase, the modeling results are assessed and a decision is made on model performance (Wirth & Hipp, 2000). These phases are combined, as they are closely interrelated and build further on one another.

The first two steps in the modeling process, are dataset selection and feature selection. In these two steps, the different datasets generated in the previous phase are evaluated and an optimal subset of features is selected. In the dataset selection step, the effect of each uniquely aggregated dataset on model performance is evaluated. Thereafter, the dataset with the best model performance is chosen for the features selection step. In the feature selection step, different subsets of features are evaluated using Recursive Feature Elimination with Cross Validation (RFECV). To elaborate, RFECV has proven in previous research to reduce the number of features effectively, while improving model performance (Misra & Yadav, 2020). Moreover, this method has been

applied to turnover research with success by Yadav et al. (2018).

Both steps are evaluated as follows: First a random forest is trained on the data, with the same (default) hyperparameters throughout the trials to facilitate a fair comparison. A random forest is chosen for this comparison, as it offers good performance on complex datasets, while maintaining transparency on prediction outcomes (Speiser et al., 2019). Moreover, repeated stratified K-fold cross-validation is used to validate model performance, as this form of cross validation is proven to be a reliable method to asses the models without overfitting (Krstajic et al., 2014). This version of cross validation keeps the same imbalance within samples, so that folds accurately represent the full dataset. In both steps, the model with the highest AUC_{Pr} is considered the best model, as elaborated in section 3.2.2.

The dataset with the optimal set of features found with RFECV is used as a basis for selecting the best performing machine learning model. Specifically, the models found in the literature review (section 4.2) are evaluated in this step. Moreover, each model that is trained has a certain set of unique hyperparameters which determine the learning properties of the model. These hyperparameters affect the model performance, therefore it is important to optimize them in order to arrive at the most suitable version of the model. In general, three methods for determining the hyperparameters are often used, Grid Search, Manual Search and Random Search. Random Search is used for this thesis, due to its superior performance in comparison with the other two methods (Bergstra & Bengio, 2012). Additionally, all results are validated using repeated stratified K-fold cross-validation and the model with the highest AUC_{Pr} is chosen, as in the previous steps.

Lastly, for the best performing model, the optimal classification threshold is determined using the precision recall curve. The optimal threshold is the point in the precision recall curve where the F1 score (or F_β score) is the highest, which is again evaluated using repeated stratified K-fold cross-validation. At this point, the AUC_{Pr} is no longer relevant, as it is the same for each point on the precision recall curve.

In this phase, the answers to sub-research questions 2 (section 2.4.2) and 3 (section 2.4.3) are found.

5.6 Deployment

In the deployment phase, a deployment plan is given based on the findings in the previous phases (Wirth & Hipp, 2000). In this phase, the final model is interpreted so that it can be used by supervisors and the HR-department for actual retention strategies and interventions. Two state-of-the-art methods for model interpretation are LIME (Local Interpretable Model-Agnostic Explanations) (Ribeiro et al., 2016) and SHAP (Lundberg & Lee, 2017). These methods are popular since their interpretations are additive. To elaborate, SHAP and LIME values which have been calculated for individual populations can be summed for the interpretation of larger aggregated groups. Therefore, the interpretation is flexible and does not need to be recalculated for each specific insight requested by the HR-department or supervisors.

To interpret the model, SHAP is chosen over LIME for the following reasons: Firstly, SHAP is optimized for tree based models, reducing computation time from multiple hours down to a few minutes (Lundberg & Lee, 2017), therefore reducing computation costs for the company and decreasing deployment time. Moreover, LIME is not optimized for the XGB model, making interpretation of this model difficult and time consuming. Secondly, LIME is less consistent in its interpretations than SHAP over multiple runs (Lundberg & Lee, 2017). Using SHAP, sub-research question 4 is answered (section 2.4.4).

As a last step, in the deployment phase recommendations are made to the company on which model to implement and how to interpret its results. Moreover, the findings of this study are communicated using an oral presentation in addition to the findings in this thesis. The actual implementation of the model is not discussed further, as it falls outside of the scope of this thesis.

Chapter 6

Business Understanding

This chapter has been removed for confidentiality reasons.

6.1 Employee Turnover Costs

6.2 Employee Retention Costs

Chapter 7

Data Understanding & Data Preparation

This chapter discusses the preprocessing steps that are taken in order to use the dataset for the purpose of binary classification. Section 7.1 describes the dataset that is provided by the company, and section 7.2 elaborates how the dataset is checked for any problems or irregularities. Thereafter, section 7.3 describes how the identified problems are remedied and how redundant data is removed. Subsequently, section 7.4 discusses the creation of new features and the transformation of existing features using the cleaned dataset. Additionally, old features used in the creation of new features are dropped as their information is captured in new features. Section 7.5 describes how the resulting dataset is split into two datasets and how different versions of the dataset using different aggregation rules are generated.

7.1 Data Description

In collaboration with the company a dataset containing various employee features was constructed, with the features presented in Table 4.2 as a guideline. This dataset is the product of an integration of data distributed amongst a CRM- and HR-system, one dedicated to hour registration and the other dedicated to employee properties registration. These two datasets were integrated by the company and enriched with travel distances, travel times and employee evaluation scores, resulting in one final dataset.

Each entry in the dataset is a unique combination of an employee ID and a period number together with the properties of an employee in this specific period. Each period is identified by a year and period number encoded as YYYYPP. For example, period two in 2016 is encoded as 201602 in the dataset. Each year consists of 12 periods, with each period spanning four or five weeks. Note that the period numbers in the dataset and month numbers used in a calendar overlap slightly. However, they are not identical, but cover a slightly different range of days. These differences do not influence the interpretability of any seasonal patterns in the data, as the aforementioned differences are minor.

An example of the structure of this dataset is shown in Table 7.1. Each employee has multiple entries within the dataset, based on the number of periods they have been employed within the company. Since the dataset covers a time period from January 2016 up until and including April 2021, an employee can have at most 64 entries. In practice, an employee will have a number of entries ranging between 1 and 64, as employees can enter and leave the company at different points in time.

The original dataset provided by the company consists of 39 features. A list of the names of these features, their data type and a brief description is shown in Table 7.2. For most features this brief description suffices, however, for some features some extra elaboration is necessary. These features are elaborated in the following paragraphs.

Table 7.1
Example of the structure of the dataset provided by the company

ID	Period	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	...	X_n
3465363	201701	v_{1ip}	v_{2ip}	v_{3ip}	v_{4ip}	v_{5ip}	v_{6ip}	v_{7ip}	v_{8ip}	...	v_{nip}
3465363	201702	v_{1ip}	v_{2ip}	v_{3ip}	v_{4ip}	v_{5ip}	v_{6ip}	v_{7ip}	v_{8ip}	...	v_{nip}
3465363	201703	v_{1ip}	v_{2ip}	v_{3ip}	v_{4ip}	v_{5ip}	v_{6ip}	v_{7ip}	v_{8ip}	...	v_{nip}
3465363	201704	v_{1ip}	v_{2ip}	v_{3ip}	v_{4ip}	v_{5ip}	v_{6ip}	v_{7ip}	v_{8ip}	...	v_{nip}
6575675	201808	v_{1ip}	v_{2ip}	v_{3ip}	v_{4ip}	v_{5ip}	v_{6ip}	v_{7ip}	v_{8ip}	...	v_{nip}
6575675	201809	v_{1ip}	v_{2ip}	v_{3ip}	v_{4ip}	v_{5ip}	v_{6ip}	v_{7ip}	v_{8ip}	...	v_{nip}
6575675	201810	v_{1ip}	v_{2ip}	v_{3ip}	v_{4ip}	v_{5ip}	v_{6ip}	v_{7ip}	v_{8ip}	...	v_{nip}
6575675	201811	v_{1ip}	v_{2ip}	v_{3ip}	v_{4ip}	v_{5ip}	v_{6ip}	v_{7ip}	v_{8ip}	...	v_{nip}

Note. X_1 to X_n represent the n features in the dataset, were v_{1ip} to v_{nip} represent their specific values for feature n, employee i and period p.

The feature number of days in period is a numerical feature stating the number of days that an employee was able to work in a certain period. Specifically, the number of days on which employees are able to work differ from period to period, one period could have more weekend days or vacation days than another. Consequently, this is an important feature that enables an accurate comparison between periods.

The dataset also contains nine different types of hours (rows 5 - 13). Most hour types, such as overtime are straightforward, however, the two sick hour types deserve some elaboration. Whenever an employee falls ill, its hours are registered as short sick leave hours. When a period of illness exceeds a predetermined time span of six weeks, the upcoming sick leave hours are registered into a different hour type, sick leave long. To illustrate, whenever an employee is on sick leave for eight weeks, the hours in the first six are registered as short sick leave, the hours in the remaining two weeks are registered as long sick leave.

Six features in the dataset are of the date type. It is important to highlight that these dates are not constrained by the start and end date of the dataset. Specifically, birth- and company enter dates can be from before January 2016 and company exit dates can be after April 2021. Moreover, some dates have the property that they can change throughout an employee's tenure. To illustrate, the role start date and role end date of an employee change whenever the end date of a role is reached and a new role is entered. Role changes are associated with an employee stepping up in the corporate ladder, so for example, a change from medior to senior level.

Lastly, four features are included concerning results from company evaluations. Specifically, these features are retrieved from employee satisfaction surveys where employees anonymously evaluate the company. The first three scores: job freedom & responsibility, culture diversity & leadership, and company satisfaction, concern a rating between one and ten, based on the employee's perception of the division of the company he/she is active in. A higher score indicates that the employee considers his environment to be positive and welcoming. The last score, job searching, is a metric for employees who indicated that they have searched for a new job in the last three months, expressed as a fraction of the total number of employees in the department. Each of the aforementioned evaluation scores are available for the company on a location - line of service basis, to keep the company from identifying individual employees. To illustrate, within each unique combination of department and location all employees have the same scores for these features. Since these these four scores represent the working environment of the employee, this aggregation poses no problems for the analysis. They are used to show the general atmosphere that the employee is exposed to on a daily basis. For example, when the job searching feature of a department is high, individual employees can be negatively affected by close colleagues leaving their team/department or actively searching for a new job.

Table 7.2
Features included in the original dataset together with their type and description

Feature	Type	Description
ID	Numerical	Unique identifier
Period	Numerical	Year and period identifier
NumberOfDaysInPeriod	Numerical	Number of days in period
Location	Categorical	Location of the company division
TotalHours[a]	Numerical	Total number of hours worked
SickLeaveShort	Numerical	Sick leave short (< 6 weeks)
SickLeaveLong	Numerical	Sick leave long (registered after 6 weeks)
TrainingHours	Numerical	Hours spent on training
Overtime	Numerical	Hours of overtime
PaidLeave	Numerical	Hours of paid leave
UnpaidLeave	Numerical	Total number of hours of unpaid leave
StudyLeave	Numerical	Hours spent on obligated accountancy related study
AssignmentHours	Numerical	Hours spent on customer assignments
NumberOfAssignments	Numerical	Number of customer assignments worked on
BirthDate	Date	Birth date
Gender	Categorical	Gender (M/F)
MaritalStatus	Categorical	Marital status
Role	Categorical	Role within the company
RoleEndDate	Date	Current role end date
RoleStartDate	Date	Current role start date
CompanyEnterDate	Date	Company enter date
CompanyExitDate	Date	Company exit date
ReasonForExit	Categorical	Reason for leaving the company
InitiativeExit	Categorical	Exit initiative (Company or employee)
Contractstatus	Categorical	Contract information is recent (Y/N)
ContractType	Categorical	Contract type (Permanent or temporary)
EmployeeType	Categorical	Employee type (Non-temporary, student, temporary)
ContractHours	Numerical	Contract hours per week
SalaryTier	Numerical	Salary tier
SalaryPosition[b]	Categorical	Position within the salary tier
SalaryIncrease	Numerical	Salary increase percentage
SalaryEndDate	Date	Salary re-evaluation date
LineOfService	Categorical	Line of service
Distance	Numerical	Travel distance (KM) home - company
TravelTime	Time	Travel time (HH:MM:SS) home - company
JobFreedomResponsibility[c]	Numerical	Freedom and responsibility rating
CultureDiversityLeadership[c]	Numerical	Culture, diversity and leadership rating
CompanySatisfaction[c]	Numerical	Employee satisfaction rating
JobSearching[c]	Numerical	Percentage of people that searched for a new job in the past 3 months

Note. All features mentioned in this table are per employee ID per period.
[a] The sum of all other hour types (`SickLeaveShort`, `SickLeaveLong`, `TrainingHours`, `Overtime`, `PaidLeave`, `UnpaidLeave`, `StudyLeave`, `AssignmentHours`).
[b] Tiers are divided in Low = bottom 25%, Medium = middle 50% and High = top 25%.
[c] Scores are retrieved from employee satisfaction surveys and are on a location - line of service level. Specifically, employees working at the same line of service at the same location, will have the same rating.

7.2 Data Validation

The dataset consists of 164,594 entries spanning over 5,810 unique employees. In order to use the data for machine learning, the integrity of the dataset needs to be validated. Specifically, the data needs to be checked for missing values, duplicate entries and entry errors.

To start, the dataset is checked for missing ('NULL') values. Missing values are present for 10 of the total 39 features. Upon closer inspection, these 10 features are the nine hour types (rows 5 - 13, Table 7.2) and the number of assignments worked on. These features all originate from the same information system, which registers hours as a 'NULL' value when no hours are written in certain hour types. Similarly, when an employee did not work on any assignments in a period, no assignments were registered, thus a 'NULL' value is recorded.

As a next step, the dataset is checked for a second type of missing values, the empty string (''), as some information systems use an empty string to register missing values. Empty strings are present in 8 of the total 39 features. In five of these eight features the empty string values are correctly used and need no remedy. To illustrate, whenever an employee has not left the company, all features associated with this exit are filled with an empty string, as they have no value. Similarly, when an employee has no salary increase in a period, this is reflected by an empty string instead of a zero. Finally, the categorical features marital status and line of service have empty strings. For marital status, a large number of entries (over 5,000) have no category associated to them, for line of service only 16 entries have this problem.

The dataset is also checked for duplicate entries, which can sometimes be introduced into a dataset while merging multiple data sources. Duplicate entries are present in the dataset, 72 rows of data appear in the dataset more than once.

Lastly, the dataset is checked for outliers. This is done in the following manner: For each feature, the extreme values are verified with a content expert from the company. After careful consultation with the content expert, no outliers are detected. A reason for the absence of outliers could be the strict constraints that are placed on the manually entered data in the CRM- and HR-system of the company.

7.3 Data Cleaning

As a starting point for the data cleaning process, the problems uncovered in the data validation (section 7.2) are remedied. Specifically, the different hour types and number of assignments which have 'NULL' values when no hours were recorded are addressed. All 'NULL' values are imputed with a zero, as a zero accurately represents the absence of hours/assignments in these features. Similarly, all salary increase entries with an empty string where imputed with a zero as well.

Thereafter, the empty string ('') values in the two categorical features marital status and line of service are addressed. The feature marital status has a large number (over 5,000) of entries in which no marital status is indicated. Since it is not possible to determine what the actual marital status is of these employees at specific points in time, these entries are grouped into a new category 'Not specified'. Deleting these entries would mean a significant loss of data, therefore this remedy is preferred. The empty string values in line of service are addressed in a different manner. Namely, these entries are removed from the dataset, which is done for the following reasons: First, all of these false entries occur for three specific employees, so only a minority of employees is affected when removing these entries from the dataset. Secondly, no other lines of service exist within the company than the ones already present in the other categories. Therefore, creating a new category 'Other' for these employees does not make sense. Lastly, it is impossible to impute their actual line of service, as the dataset is anonymized, so no inquires can be made to the actual employees.

As a next step the duplicate entries identified in section 7.2 are removed from the dataset, as they are redundant.

These duplicate entries are not the only redundant information in the dataset. Some filtering needs to be applied to the dataset to remove any entries outside of the scope of the thesis. Firstly,

all employees that do not have a permanent contract are removed from the dataset, as they fall outside of the scope of the voluntary turnover problem (Zhao et al., 2019). Concretely, the features employee type and contract type are used for the filtering of these employees. For the employee type, three types exist within the dataset: non-temporary employees, students and temporary employees. Only non-temporary employees are included in the final dataset, students and temporary employees are left out due to the temporary nature of their employment. For the contract type, four categories exist: a permanent contract and three different types of temporary contract types, differentiated by their duration. Employees with any of these three temporary contract types are removed from the final dataset, for the same reason as students and temporary employees are removed. Note that, since these features describe roughly the same property, some overlap exists between both groups that are filtered out.

7.4 Feature Creation & Transformation

After validating and cleaning the data, some features need to be transformed so that they can be used by a machine learning model. Additionally, some features need to be created from existing features in order to enrich the dataset with valuable information.

As mentioned in section 7.1, each period spans a different number of working days, due to how periods are defined by the company. This difference in working days in a period makes it difficult to accurately compare periods. To illustrate, one employee can have more hours in a period compared to another, not because of overwork or other anomalies, but simply because that period consisted of more working days that the other. To resolve this issue, each hour type and the number of assignments are divided by the total number of working days inside a period, enabling a fair comparison between periods. Note that hours and days concerning holidays have been excluded in the dataset provided by the company, contributing to this fair comparison.

The two features dealing with the salary need minor transformations as well. First, the percentage salary increase is transformed into a factor representing the multiplicative increase in salary. This factor representation is useful for the aggregation at a later stage. An example of this transformation is show in equation 7.1, with $SI_{\%}$ as the original percentage salary increase and SI_f as the salary increase factor. Second, the salary position of an employee is currently a categorical feature with the categories Low, Medium, High. This feature is transformed using ordinal encoding with; Low = 0, Medium = 1 and High = 2.

$$SI_f = 1 + \frac{SI_{\%}}{100} \tag{7.1}$$

Another important feature to transform so that a machine learning model is able to interpret it, is the travel time feature. Currently, this feature is denoted as HH:MM:SS (Hours:Minutes:Seconds) and formatted as a string. This feature is converted from hours, minutes and seconds to an integer denoting the total travel time in seconds, to ensure no unnecessary loss of information.

The dataset also contains a variety of categorical features, which some of the selected models are not able to handle without transforming them first. A standard approach in transforming categorical features is to use one hot encoding, also known as creating dummy features. In this approach, each category is transformed to a binary feature, with the category that applies to the entry denoted as a 1 and the others as a 0. The features to which one hot encoding is applied are gender, marital status, line of service and location. The role feature is a categorical feature as well, however, this feature consists of approximately 300 categories making it unfit for one hot encoding. Namely, one hot encoding this feature would introduce approximately 300 features of which most entries are 0, also known as sparse features. A sparse dataset with high dimensionality increases model training time and reduces model performance (El-Khatib, 2010). To avoid these problems, the role categories are ordinally encoded, even though this encoding does not adhere to the hierarchy of the roles within the company. Ordinal encoding is chosen over nominal encoding to enable a fair comparison between the different models, since the MLP model does not accept nominally encoded features.

The feature denoting the period in which the entry is recorded can be used to calculate different duration related features. Specifically, the following features are used to calculate different time related features for each specific entry in the dataset. Firstly, the time difference between the period of an entry and the birth date of an employee are used to calculate the age of an employee. Secondly, the time difference between the company enter date and the period of an entry are used to calculate the tenure of an employee. Thirdly, the time difference between the last promotion date and the period of an entry are used to calculate the time since last promotion of an employee. Furthermore, time passed is denoted with whole years as integers and the remaining months as decimals. To illustrate, 12 years and 4 months is denoted as 12.33 in the dataset.

Lastly, the voluntary turnover feature is created. This feature is created by first checking the company exit date feature. When the company exit date is not equal to the period of an entry, or if an employee has no company exit date, turnover is classified as 0 (no turnover). If the company exit date is equal to the period of an entry, an evaluation of the type of turnover is made by evaluating the reason for exit and initiative of exit features. Entries with employee initialized exits are classified as 1 (turnover). Other reasons and initiatives are either marked as no turnover when it concerned an internal switch, or marked as 'remove' when the turnover is involuntary. Thereafter, the employees that are marked with 'remove' are completely removed from the dataset, since their turnover type is not part of the scope of this thesis.

As a final step, the original features that have been used for transformation, creation and one hot encoding are removed from the dataset, as their new more informative counterparts take their place. To conclude, the features that are removed, transformed and introduced in this section are shown in summary in Table 7.3.

Table 7.3
Features introduced, transformed and dropped during the feature creation and transformation process

Feature	Operation	Description
Location	Transformed	From categorical to one hot encoding
TotalHours	Transformed	Total number of hours worked
SickLeaveShort	Transformed	From period to day level
SickLeaveLong	Transformed	From period to day level
TrainingHours	Transformed	From period to day level
Overtime	Transformed	From period to day level
PaidLeave	Transformed	From period to day level
UnpaidLeave	Transformed	From period to day level
StudyLeave	Transformed	From period to day level
AssignmentHours	Transformed	From period to day level
NumberOfAssignments	Transformed	From period to day level
Gender	Transformed	Binary encoding
MaritalStatus	Transformed	One hot encoding
Role	Transformed	Ordinal encoding
SalaryPosition	Transformed	Ordinal encoding
SalaryIncrease	Transformed	From percentage to factor
LineOfService	Transformed	One hot encoding
TravelTime	Transformed	Converted to seconds
BirthDate	Dropped	Used for age
RoleStartDate	Dropped	Used for time since last promotion
CompanyEnterDate	Dropped	Used for tenure
CompanyExitDate	Dropped	Used for turnover target feature
ReasonForExit	Dropped	Used for turnover target feature
InitiativeExit	Dropped	Used for turnover target feature
Age	Added	Calculated from birth date
TimeSinceLastPromotion	Added	Calculated from role start date
Tenure	Added	Calculated from company entry date
Turnover	Added	Derived from company exit date, reason for exit and exit initiative

7.5 Dataset Aggregation

As mentioned in section 3.1, the turnover process is not instantaneously, but something that develops over a longer period of time. Therefore, the current state of the dataset, a single period per employee, most likely does not capture the turnover process completely. Consequently, in order to accurately reflect the turnover process in the data, the dataset needs to be aggregated from a period level to a timespan covering multiple periods. To achieve this aggregation, two aggregation methods are employed, a rolling window aggregation and a last period aggregation. Additionally, as mentioned in section 2.4.3, a significant part of the turnover dataset spans across a global pandemic, giving rise to very different working conditions. Therefore, in the aggregation process, a dataset with pandemic and without pandemic turnover is constructed so that their performance can be compared. The inner workings of these aggregations are elaborated in this section.

7.5.1 Rolling Window Aggregation

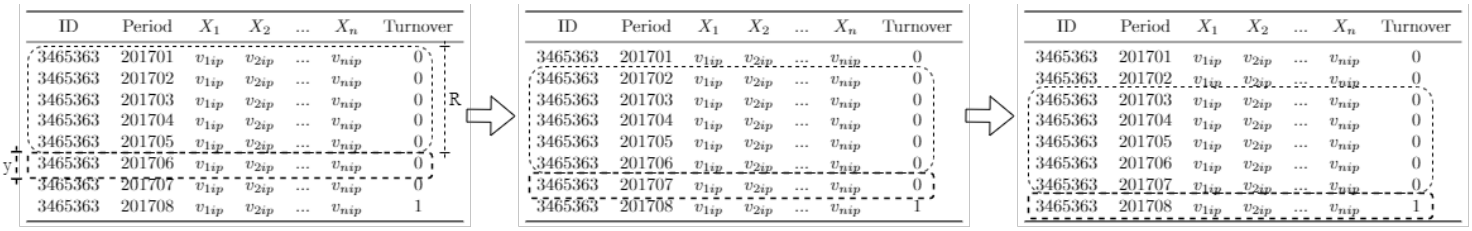
As mentioned in section 5.4, a rolling window is applied to aggregate the period entries to multi-period windows. To start, this section shows a general representation of the rolling window aggregation applied to the dataset. Thereafter, the operations applied within each window are discussed.

Generally, a rolling window aggregation is applied as follows: first, a portion of the data within a dataset is selected of size R. Additionally, a target window directly after R is selected of window size y. User defined aggregation transformations are applied to the data in window R, and the target feature is determined by looking at the data inside target y. Thereafter, the window slides over the data, each step including a more recent data entry, and dropping the oldest. This process is repeated until the end of the dataset is reached and no steps can be taken without decreasing R and y. To illustrate, consider R = 3 and y = 1. In the first step, R spans the first three periods and y covers the fourth, in the next step, R covers periods two, three and four and y covers the fifth.

Before the rolling window aggregation is applied, the dataset is split into entries that have taken place before and during pandemic times, with the non-pandemic dataset ending on period 201912 and the pandemic dataset starting on 202004. A few months in which the pandemic was gaining traction are excluded, as the shift in working conditions changed gradually during this time.

To use this rolling window aggregation method on the pandemic and non-pandemic dataset, all entries belonging to the same employee are grouped together. On each of these groups, containing the data of one employee sorted from first to last entry, a rolling window is applied from the oldest to the newest entry as explained in the previous section. This process is illustrated in Figure 7.1, with window size R and target window size y. As no optimal values for R and y are known, various values were used generating differently built up datasets of which the performance can be compared. Specifically, R is varied between 1 and 12 periods, and y is varied between a 1 and 4 period ahead forecast. Note that, employees with less entries than the sum of the chosen window size (R) and target window size (y) are removed before the aggregation as not enough data is present to at least aggregate one complete window.

Figure 7.1
Illustration of the rolling window principle applied to the employee entries



Note. In this figure aggregation window size is denoted with R and target window size with y

Now that the general process of applying a rolling window aggregation to the dataset is known, a closer look needs to be taken at the computations that take place inside the window. Specifically, how multiple rows of data are aggregated to form one row of data.

To start, the computations inside the aggregation window (R) are clarified. In order to get an accurate insight into the average routine of an employee within a window, the mean of the hour types and the number of assignments entries within the window is taken. An average is chosen to weigh the information in each period leading up to the potential turnover equally. For some employee properties, the last entry within the window (R) is taken, as these properties represent the status of the employee at the point of (not) turning over. This choice is made for the salary tier, salary position, age, tenure, time since last promotion, travel time, gender, role, and the four ratings (job freedom, culture, company satisfaction and job searching). For the salary

increase, the product was taken, as the salary increase in each entry is described with a factor and a multiplication accurately represents the total change from the beginning of the window to the end. Lastly, for all one hot encoded features (marital status, line of service and location), the maximum of each dummy variable inside a window is taken. To illustrate, whenever an employee divorces, two different marital statuses are encoded with a one inside the window. When taking the maximum of all dummy variables, both statuses are encoded with a 1 in the final aggregation, encoding the change within the window.

Lastly, since the target is to predict if an employee is going to leave in the next y periods, the only information that is needed from the target windows is the turnover feature. Turnover is 1 if the employee leaves at the end of the target window (as shown in the right most target window in Figure 7.1), turnover is 0 if the employee does not leave at the end of the target window (as shown in the left two target windows in Figure 7.1).

One issue arises when using this approach. Throughout an employee's tenure, the target feature (turnover) is 0, whereas only in the final period the target feature might be 1 (only if the employee left). By taking many windows over these entries, many samples with no turnover are introduced, as opposed to a predetermined number of turnover samples. This effect increases when the window size shrinks, as more windows fit within the total tenure of an employee. Two approaches are taken to remedy this problem of imbalanced data. Firstly, whenever an employee leaves the company, only the aggregated windows in which this employee leaves is kept, all other aggregated windows created from this employee containing no turnover are removed. This remedy reduces the number of non turnover samples, while also reducing the noise in the non turnover class, as windows before the turnover might contain signs of turnover while being labeled as non turnover. Secondly, during the modeling process, different undersampling ratios are applied to counter the imbalance.

7.5.2 Last Window Aggregation

To complement the complex rolling window aggregation, a more straightforward aggregation method is used as well. The reason being, that at this point it is unknown if a model will perform better on a more complex or straightforward aggregation of the data entries.

Similar to the rolling window aggregation approach, the last window aggregation approach uses a target window of $y \in [1, 2, 3, 4]$, representing a 1 to 4 period ahead forecast. Moreover, the aggregation window (R) is fixed to the last 12 periods right before the target window. Therefore, only taking the last window of each employee for the aggregation without rolling over all entries. A range of 12 periods is chosen, as certain (promotion related) features tend to be relevant only once every 12 periods.

Before the last window aggregation is applied, the dataset is split into entries that have taken place before and during pandemic times, just as was done for the rolling window aggregation. For the last window approach, the non-pandemic split is identical to the rolling window split, however for the pandemic dataset a larger period is taken. Concretely, the pandemic dataset starts exactly R + y - 1 periods before 202004, placing the target feature of the earliest departing employees exactly at the start of the pandemic. This choice is made to negate the effect of the large size of the aggregation window in combination with the small training data period. Only taking the data from 202004 and onward would only take turnover in the last periods into account. Additionally, target window sizes larger than $y = 1$ would make the window larger than the dataset, eliminating two, three and four period ahead forecasts. A choice was made to not incorporate this approach into the rolling window aggregation, as the windows are generally small enough to include enough training data within the pandemic dataset.

Within the last window, the following approaches are taken to aggregate the individual features. To start, for each of the features sick leave short, sick leave long, hours of paid leave, hours of unpaid leave and overwork, three new features are created. Specifically, these features indicate if these hour types occurred within the last 3, 6 and 12 months of the last period window (R). Furthermore, these features are binary encoded, having a 1 if the hour type occurred and a 0 if it did not. Additionally, the time since last promotion is binary encoded as well, denoting if an

employee received a promotion in the last 12 periods, as promotions are distributed generally once every 12 periods. Moreover, for the salary increase feature (recall that this is denoted as a factor), the product of the entries inside the window are taken, similar to the rolling window aggregation.

For the total hours, assignment hours and number of assignments, bins are created in concordance with a subject expert. The following bins are created for each of these three features, with each features averaged over the corresponding months:

Total hours (h_t) is divided into the bins:

- 0 ($h_t < 6$)
- 1 ($6 \geq h_t < 8$)
- 2 ($h_t \geq 8$)

Assignment hours (h_a) is divided into the bins:

- 0 ($h_a < 2$)
- 1 ($2 \leq h_a < 4$)
- 2 ($4 \leq h_a < 6$)
- 3 ($6 \leq h_a < 8$)
- 4 ($h_a \geq 8$)

Number of assignments (a_n) is divided into the bins:

- 0 ($a_n < 1$)
- 1 ($1 \leq a_n < 2$)
- 2 ($2 \leq a_n < 4$)
- 3 ($a_n \geq 4$)

Lastly, as for all other features, the last entry within the aggregation window (R) is taken. Additionally, within the target window (y), the turnover is determined in the exact same manner as the rolling window aggregation.

Chapter 8

Modeling & Evaluation

This chapter elaborates on how the previously generated datasets undergo the modeling process. Section 8.1 describes how the best performing dataset is selected, and section 8.2 discusses how a subset of features is selected from the best performing dataset. Both selections are made, based on the performance of a random forest model with default hyperparameters, using cross-validation. Section 8.3 elaborates on how the resulting dataset is used for model selection. Specifically, the six models discussed in section 3.2.1 are optimized using hyperparameter tuning and cross-validation, and the performances of the best performing models are evaluated. The best performing model is chosen for the threshold tuning process. Section 8.4 describes this threshold tuning process, in which the optimal threshold to make a distinction between the two binary classes is determined.

For each step in the modeling process, 10 repeats of a 5-fold stratified cross-validation is used, meaning that model performance is evaluated over 50 iterations and averaged. Additionally, throughout the whole modeling process, the area under the precision recall curve (AUC_{PR}) is used for ranking the models. Moreover, the model with highest AUC_{PR} within an analysis, is considered the best performing model. For the last step, threshold tuning, the F1 score is used as a performance benchmark, as the AUC_{PR} is used in this step to find the optimal balance between the precision and recall of the final model.

8.1 Dataset Selection

In this section, all datasets generated in the dataset generation process (section 7.5) are compared based on the performance of a vanilla (default model parameters) random forest on the datasets. Specifically, for each dataset, the random forest model is trained 50 times (5-fold stratified cross validation with 10 repeats), after which the performance benchmarks discussed in section 3.2.2 are computed over these 50 runs. Moreover, the area under the precision recall curve is used as the main indicator for performance, as this metric indicates the potential of the dataset after threshold tuning later in the process, as explained in section 3.2.2. This process is repeated for both the pandemic and non-pandemic datasets after which a look is taken at a combination of both.

8.1.1 Non-pandemic Datasets

Initially, the performance of the random forest model on the non-pandemic datasets is evaluated. The top 10 best performing datasets are shown in Table 8.3, the full results of all datasets including standard deviations can be found in appendix B.1. Table 8.3 shows that the last window aggregation method has a superior performance as opposed to the rolling window aggregation method. Moreover, as expected, model performance drops when the y step ahead forecast increases. Implying that the model has more difficulty in its classifications when it needs to classify cases further in the future.

Table 8.1
Top 10 performing datasets for the non-pandemic dataset selection

Aggregation Type	R	y	Accuracy	Precision	Recall	F1	AUC_{Pr}
Last Window	12	1	0.767	0.685	0.216	0.327	0.552
Last Window	12	2	0.768	0.666	0.211	0.319	0.537
Last Window	12	3	0.772	0.666	0.193	0.297	0.533
Last Window	12	4	0.775	0.655	0.191	0.294	0.519
Rolling Window	12	1	0.987	0.000	0.000	0.000	0.377
Rolling Window	11	1	0.987	0.000	0.000	0.000	0.341
Rolling Window	10	2	0.987	0.020	0.000	0.000	0.323
Rolling Window	9	3	0.987	0.300	0.003	0.005	0.320
Rolling Window	8	4	0.987	0.180	0.001	0.003	0.301
Rolling Window	10	1	0.987	0.000	0.000	0.000	0.294

Note. With AUC_{PR} as the area under the precision recall curve.

As mentioned in section 7.5.1, the rolling window datasets have a severe imbalance due to the nature of the rolling window approach. This imbalance causes the model to place more importance on the non-turnover class during training, resulting in a model that favours predicting non-turnover when it is presented with new samples. In Table 8.1, a near perfect accuracy is achieved for the rolling window datasets, whereas the precision, recall and F1 are zero, or close to zero. This result can be explained by the imbalance of the datasets, as this model classifies nearly every non-turnover sample correctly and fails to identify nearly every turnover sample, because it almost always predicts a sample to be non-turnover.

To remedy this unbalance, random undersampling is applied to the majority class of the best performing rolling window dataset (R=12, y=1), to see if performance increases if the imbalance is reduced. This method makes use of an undersampling ratio (U_r) which is defined as the ratio between the number of samples in the minority class ($N_{minority}$) and the number of samples in the majority class ($N_{majority}$). The formula for this ratio is shown in equation 8.1

$$U_r = \frac{N_{minority}}{N_{majority}} \tag{8.1}$$

The undersampling ratio can vary between the original ratio of the majority and minority class and one (50:50 ratio). A wide range of undersampling ratios within the aforementioned range is applied to the best performing rolling window dataset and again, the performances over 50 iterations are computed. The results of this process are shown in Table 8.2, additionally, the full results including standard deviations are shown in appendix B.2.

Table 8.2 shows that, after undersampling the majority class, no performance gain is realized. The AUC_{PR} drops when the undersample ratio increases. Therefore, it can be concluded that undersampling does not increase the performance of this dataset above the performance of the overall best performing dataset. Moreover, undersampling does not increase the performance above its original level.

To conclude, the overall best performing dataset is the dataset making use of the last window aggregation with a training window size of 12 and a target window size of 1. Moreover, undersampling the best performing rolling window dataset did not increase its performance above the overall best performing dataset.

8.1.2 Pandemic Datasets

Similar to the non-pandemic datasets, the pandemic datasets are evaluated as well. The top 10 best performing datasets are shown in Table 8.3, additionally, the full results of all datasets including standard deviations can be found in appendix B.3. Table 8.3 shows that the last window approach

Table 8.2
Results for undersampling the best performing non-pandemic rolling window aggregation

Aggregation Type	R	y	U_r	Accuracy	Precision	Recall	F1	AUC_{Pr}
Rolling Window	12	1	0.02	0.987	0.000	0.000	0.000	0.259
Rolling Window	12	1	0.03	0.987	0.060	0.001	0.001	0.186
Rolling Window	12	1	0.04	0.987	0.040	0.000	0.001	0.134
Rolling Window	12	1	0.05	0.987	0.020	0.000	0.000	0.115
Rolling Window	12	1	0.1	0.987	0.077	0.001	0.002	0.062
Rolling Window	12	1	0.2	0.985	0.113	0.018	0.030	0.046
Rolling Window	12	1	0.4	0.955	0.051	0.141	0.074	0.039
Rolling Window	12	1	0.6	0.876	0.034	0.319	0.062	0.033
Rolling Window	12	1	0.8	0.762	0.027	0.496	0.051	0.031
Rolling Window	12	1	1	0.647	0.023	0.626	0.044	0.029

Note. U_r as undersampling ratio (equation 8.1)

is the best performing approach for the pandemic datasets as well. However, as opposed to the non-pandemic dataset, a larger target window increases model performance. This characteristic is further elaborated in section 8.1.3.

Table 8.3
Top 10 performing datasets for the pandemic dataset selection

Aggregation Type	R	y	Accuracy	Precision	Recall	F1	AUC_{Pr}
Last Window	12	4	0.909	0.849	0.076	0.137	0.295
Last Window	12	2	0.905	0.792	0.049	0.091	0.258
Last Window	12	3	0.906	0.697	0.055	0.100	0.255
Last Window	12	1	0.899	0.237	0.011	0.020	0.246
Rolling Window	3	4	0.991	0.360	0.017	0.032	0.097
Rolling Window	6	1	0.991	0.380	0.018	0.034	0.094
Rolling Window	5	2	0.991	0.220	0.010	0.020	0.088
Rolling Window	4	3	0.991	0.320	0.015	0.028	0.087
Rolling Window	1	1	0.991	0.520	0.016	0.031	0.085
Rolling Window	7	1	0.991	0.000	0.000	0.000	0.084

Note. With AUC_{PR} as the area under the precision recall curve.

For the pandemic dataset, the rolling window has a similar imbalance as the non-pandemic dataset. Therefore, a closer look is taken at the performance difference when using the undersampling technique discussed in the previous section. Undersampling is applied to the best performing rolling window aggregation dataset, with training window size 3 (R) and target training size 4 (y).

The results of this process are shown in Table 8.4, additionally, the full results including standard deviations are shown in appendix B.4. Table 8.4 shows that, after undersampling the majority class, no performance gain is realized. The AUC_{PR} drops when the undersample ratio increases. Therefore, it can be concluded that undersampling does not increase the performance of this dataset above the performance of the overall best performing dataset. Moreover, undersampling does not increase the performance above its original level.

To conclude, the overall best performing dataset is the dataset making use of the last window aggregation with a training window size of 12 and a target window size of 4. Moreover, undersampling the best performing rolling window dataset did not increase its performance above the overall best performing dataset.

Table 8.4
Results for undersampling the best performing pandemic rolling window aggregation

Aggregation Type	R	y	U_r	Accuracy	Precision	Recall	F1	AUC_{Pr}
Rolling Window	3	4	0.02	0.991	0.300	0.014	0.026	0.061
Rolling Window	3	4	0.03	0.991	0.230	0.011	0.021	0.050
Rolling Window	3	4	0.04	0.991	0.305	0.015	0.028	0.053
Rolling Window	3	4	0.05	0.991	0.267	0.014	0.026	0.044
Rolling Window	3	4	0.1	0.990	0.208	0.018	0.033	0.043
Rolling Window	3	4	0.2	0.985	0.080	0.043	0.051	0.040
Rolling Window	3	4	0.4	0.942	0.036	0.190	0.060	0.031
Rolling Window	3	4	0.6	0.854	0.023	0.354	0.043	0.026
Rolling Window	3	4	0.8	0.740	0.018	0.493	0.034	0.025
Rolling Window	3	4	1	0.633	0.015	0.605	0.030	0.025

Note. U_r as undersampling ratio (equation 8.1)

8.1.3 Full Dataset Evaluation

Currently, the dataset is split into two parts, pandemic data and non-pandemic data. However, the results displayed in the previous sections indicate that in practice, it might be beneficial to use the full dataset, as opposed to using two individual splits. This assumption is formed for two reasons. Firstly, when comparing the best models for both datasets, model performance for the non-pandemic dataset is much higher. This result could indicate, that the pandemic dataset has more noise and an insufficient amount of training data, leading to low model generalization. Secondly, recall that the last window aggregation uses R periods leading up to target period y for training. Consequently, increasing y leads to a lower period boundary at the start of the split, as explained in section 7.5.2. This lower boundary causes more non-pandemic data to be included in the training window. Moreover, when taking a closer look at Table 8.3, it becomes clear that a larger target window (y) equates to a better model performance for the last window datasets. Thus, when more non-pandemic data is included in the training data, model performance increases according to the data in this table.

To test this assumption, the following experiment is conducted. A random forest model is trained on the full dataset, after which its performance is evaluated using the same 50 iterations as for the other experiments in this section. Specifically, four datasets are generated using the full dataset as basis, the last window aggregation method and y $\in [1, 2, 3, 4]$. The results of this experiment are shown in Table 8.5, the full results including the standard deviation are shown in appendix B.5.

Table 8.5
Model performance on the full dataset

Aggregation Type	R	y	Accuracy	Precision	Recall	F1	AUC_{Pr}
Last Window	12	1	0.890	0.948	0.684	0.795	0.882
Last Window	12	2	0.882	0.949	0.653	0.773	0.882
Last Window	12	3	0.879	0.933	0.646	0.763	0.852
Last Window	12	4	0.897	0.962	0.683	0.799	0.877

Note. With AUC_{PR} as the area under the precision recall curve.

In Table 8.5 it can be seen that using the full dataset aggregation with the last window method increases model performance greatly. To illustrate, the AUC_{Pr} is approximately three times as high as in the best performing non-pandemic dataset. This increase in performance can be explained by the fact that the full dataset has more samples, providing more information to

the model on turnover and non-turnover samples. Moreover, this increase in samples is especially helpful for the turnover class, as this is the minority class in each dataset. Therefore, a greater variety of turnover cases is presented to the model, making these cases easier to recognize for the model.

The table also shows that AUC_{Pr} performance for y=1 and y=2 is similar (after rounding), indicating that a two period ahead forecast is also possible with a negligible loss of performance. No reasonable explanation is found for the higher performance for y=4 as opposed to y=3, as the prediction difficulty is generally expected to increase when the forecast horizon increases.

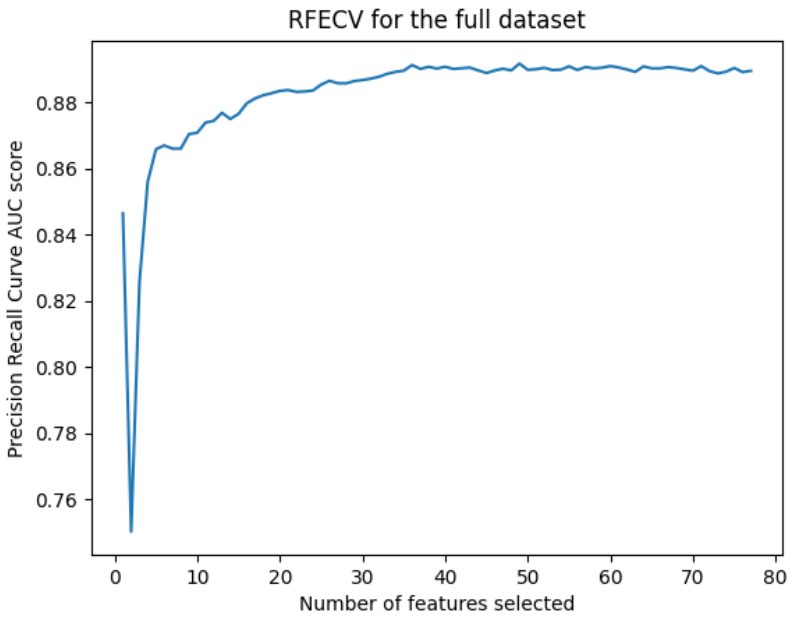
To conclude, using the full dataset with a one period ahead forecast gives the best model performance in comparison with all other datasets investigated in this section. Therefore, this dataset is chosen for the next steps in the modeling process.

8.2 Feature Selection

Currently, the dataset consists of 78 different training features, many of which might play no significant role in the turnover prediction. To keep the dimensions of the dataset as low as possible without hurting the performance, it is beneficial to eliminate these redundant features. Moreover, a dataset with lower dimension leads to lower model training times and potentially a higher model performance (Cunningham, 2008).

To achieve the dimension reduction, the RFECV method discussed in section 5.5 is used. This method recursively drops the feature with the lowest performance from the dataset, until no features are left to drop. Thereafter, the model identifies the set of features with the highest performance (AUC_{Pr} in this case) it encountered in its iterations. For each subset of features within the RFECV method, 50 iterations are used for cross-validation, the same as in the previous experiments.

Figure 8.1
Results of applying the RFECV method for feature reduction to the full dataset



Using the aforementioned RFECV feature reduction method, the AUC_{Pr} for each subset of features is retrieved. The results of the RFECV process are visualized in Figure 8.1, with the different AUC_{Pr} scores on the y-axis and the number of features displayed on the x-axis. In this

figure it can be seen that performance is rather similar for each feature dropped from 78 features down to ~35 features, after which the performance drops dramatically. The highest AUC_{Pr} of the model is achieved when 49 features are included in the model, with an AUC_{Pr} of 0.892. Therefore, some performance gain is realized when including less features in the final dataset. A list of these features is shown in appendix B.7. Lastly, note that the subset of one feature has a high AUC_{Pr} as well. Upon further inspection, this feature is the job freedom & responsibility score feature, indicating that this is an important predictor for turnover.

To validate these results, the new dataset with reduced features is cross-validated, again using 50 iterations. The results of this validation are shown in Table 8.6, additionally, the full results including the standard deviation are shown in appendix B.6. The table shows that the validation AUC_{Pr} is 0.895, which is very close (within 1σ) to the AUC_{Pr} of 0.892 found in the RFECV process. To conclude, these results validate the new set of 49 features, which will be used for the next steps in the modeling process.

Table 8.6
Model performance validation after RFECV

Aggregation Type	R	y	Accuracy	Precision	Recall	F1	AUC_{Pr}
Last Window	12	1	0.882	0.936	0.665	0.778	0.895

Note. With AUC_{PR} as the area under the precision recall curve.

8.3 Hyperparameter Tuning & Model Selection

Now that the best performing dataset together with its optimal number of features is selected, the next step is to compare model performance on this dataset. In this section, models are trained on the dataset using a variety of different hyperparameters. These hyperparameters are defined as a grid of model setting of which a random selection is made. Specifically, a large search space is defined for the hyperparameters for each model to make sure that a wide variety of values are considered in the random search. Thereafter, the model performance is cross-validated over 5 iterations, after which the model performance for the specific combination of hyperparameters is reported. This process is repeated 500 times for each model, meaning that 500 different hyperparameter combinations are evaluated. Due to computation times, it is not possible to evaluate more than 500 random combinations with 5-fold cross validation, as this is too time-intensive due to the computational power constraints. This lower number of cross validations could lead to a higher standard deviation for the performance benchmarks when the model is unstable. However, as the goal of this section is exploring different hyperparameter combinations, a larger search space is deemed more important than a (potentially) lower standard deviation.

For each different model, its best performance found in the hyperparameter tuning process is reported in Table 8.7. The full results of the hyperparameter tuning process, including the performance scores, their standard deviations and the hyperparameters for the top 50 of each model, can be found in appendix B.5.

Table 8.7 shows that the best performing model after hyperparameter tuning is the extreme gradient boosting (XGB) model. To elaborate, this model has the highest AUC_{PR} of all models and outperforms all other models that have been evaluated. Therefore, this model and the optimal set of hyperparameters (appendix B.14) found in this section, are selected for threshold tuning in the next step.

Table 8.7
Performance scores of the best performing models after hyperparameter tuning

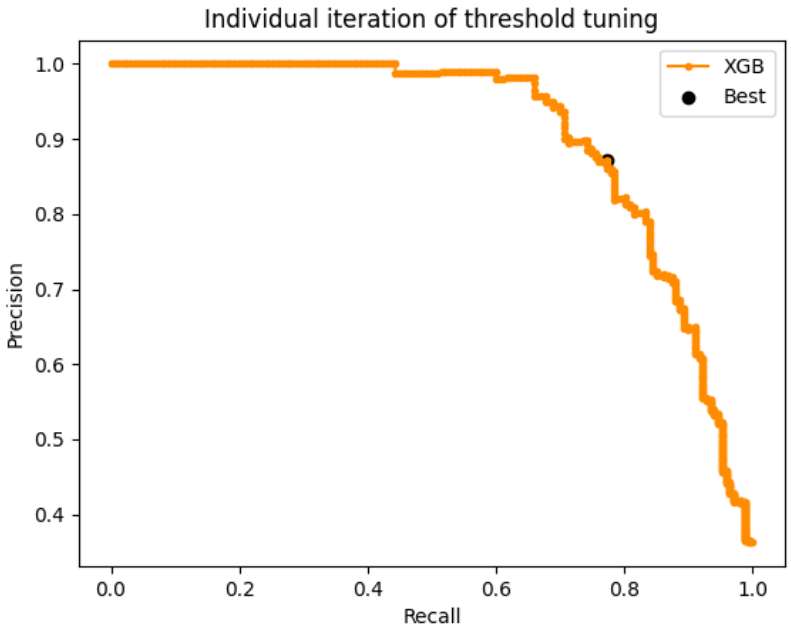
Model	Accuracy	Precision	Recall	F1	AUC_{Pr}
XGB	0.900	0.933	0.733	0.821	0.910
GBT	0.893	0.911	0.727	0.808	0.904
RF	0.895	0.956	0.696	0.805	0.898
ADA	0.884	0.868	0.739	0.798	0.881
DT	0.846	0.775	0.715	0.744	0.823
MLP	0.747	0.611	0.547	0.572	0.625

Note. With AUC_{PR} as the area under the precision recall curve.

8.4 Threshold Tuning

As explained in section 3.2.2, the precision recall curve represents the trade off between the precision and recall score for each threshold. All models in the previous sections were evaluated on their AUC_{PR}. Recall that a higher AUC_{PR} indicates that a higher F1 score can be achieved after threshold tuning. Now that the best model and its optimal hyperparameters according to the AUC_{PR} are found, it is time to investigate the optimal threshold. Specifically, different thresholds and their corresponding precision and recall scores are evaluated in the form of the F1 score. Subsequently, after investigating the F1 score at each threshold, the best threshold is chosen. Just as in most of the previous experiments, the thresholds are cross-validated over 50 iterations. In Figure 8.2 a plot of one of the 50 iterations is shown, to illustrate the process. In each iteration an optimal threshold is found (marked with 'Best' in Figure 8.2) together with its corresponding F1 score, of which the mean and standard deviation over 50 iterations is calculated.

Figure 8.2
Illustration of the results of an individual iteration of the threshold tuning process



The results of the threshold tuning process are shown in Table 8.8 and the full results including

the standard deviations are shown in appendix B.20.

Table 8.8
Results for threshold tuning

Model	Acc	Pr	Rc	F1	T
XGB	0.902	0.888	0.795	0.837	0.283

Note. With T as the optimal threshold

Table 8.8 shows that the optimal F1 score after threshold tuning is 0.837, which is higher than the initial F1 score of 0.821 found after hyperparameter tuning (Table 8.7). The threshold that corresponds to this F1 score is 0.283.

This results implies that with the optimal model and threshold found, no further modeling is necessary. However, as can be seen in Table 8.8, this optimal F1 score is associated with a recall score that is lower than the precision score. Consequently, the number of false negatives with respect to the true positives, is higher than the number of false positives with respect to the true positives.

When looking at this result from the turnover perspective, the following can be concluded: With the current threshold, the model slightly favours classifying employees that leave as not leaving (FN), instead of classifying employees that do not leave as leaving (FP). Relating this behavior to turnover costs, false negatives are more costly to the company than false positives, in other words, the cost of turnover is higher than the cost of retention per employee (sections 6.1 & 6.2). Therefore, it could be beneficial to scale the importance of precision and recall by their cost ratio. Resulting in a threshold that optimizes the cost in terms of precision and recall.

In order to accurately represent the cost ratio between false positives and false negatives, the F_β score is used (equation 3.13). In this generalized version of the F1 score, the importance of recall in terms of precision can be defined as β. Specifically, β is defined as the cost ratio between a false negative and a false positive classification. The turnover and retention costs determined in sections 6.1 & 6.2 are used to determine this ratio as follows:

The cost of a false negative is equal to the cost of a turnover. Moreover, the cost of a false positive is equal to the cost retention. Using these costs figures, β is defined as $\beta = 7.92$, so that β balances the importance of the precision and recall scores with respect to their costs.

As a next step, the F_β score with a β of 7.92 is used to determine the threshold that balances the costs of turnover and retention. The results of the threshold tuning are shown in Table 8.9, with the full results including the standard deviations shown in appendix B.21.

Table 8.9
Results for threshold tuning that balances the cost of misclassifications

Model	Acc	Pr	Rc	F7.92	T
XGB	0.421	0.354	0.999	0.971	0.001

Note. With T as the optimal threshold

Table 8.9 shows that the optimal threshold when balancing precision and recall is 0.001. This threshold is so low, that it causes the model to identify nearly every sample as turnover, as shown by the near perfect recall score. Additionally, the precision score associated with this recall score is one of the lowest possible scores, as shown in Figure 8.2. Moreover, the accuracy score shows that the model has difficulty in classifying samples correctly when using this threshold.

To conclude, the model threshold that balances the cost of turnover and retention has no practical use. It identifies nearly every employee at risk of turning over, due to the high costs associated with turnover as opposed to retention. For this reason, the threshold of 0.283, associated with the optimal F1 score is chosen for the final model.

8.5 Relating Model Performance to Turnover

This section has been removed for confidentiality reasons.

Chapter 9

Interpretation

This chapter discusses how the model built in chapter 8 is interpreted on a global level (test data) and individual level, so that it can be used for employee retention strategies. Specifically, section 9.1 describes the interpretation of turnover characteristics within the test set as a whole, and section 9.2 discusses the interpretation of two individual classifications.

9.1 Global Interpretation

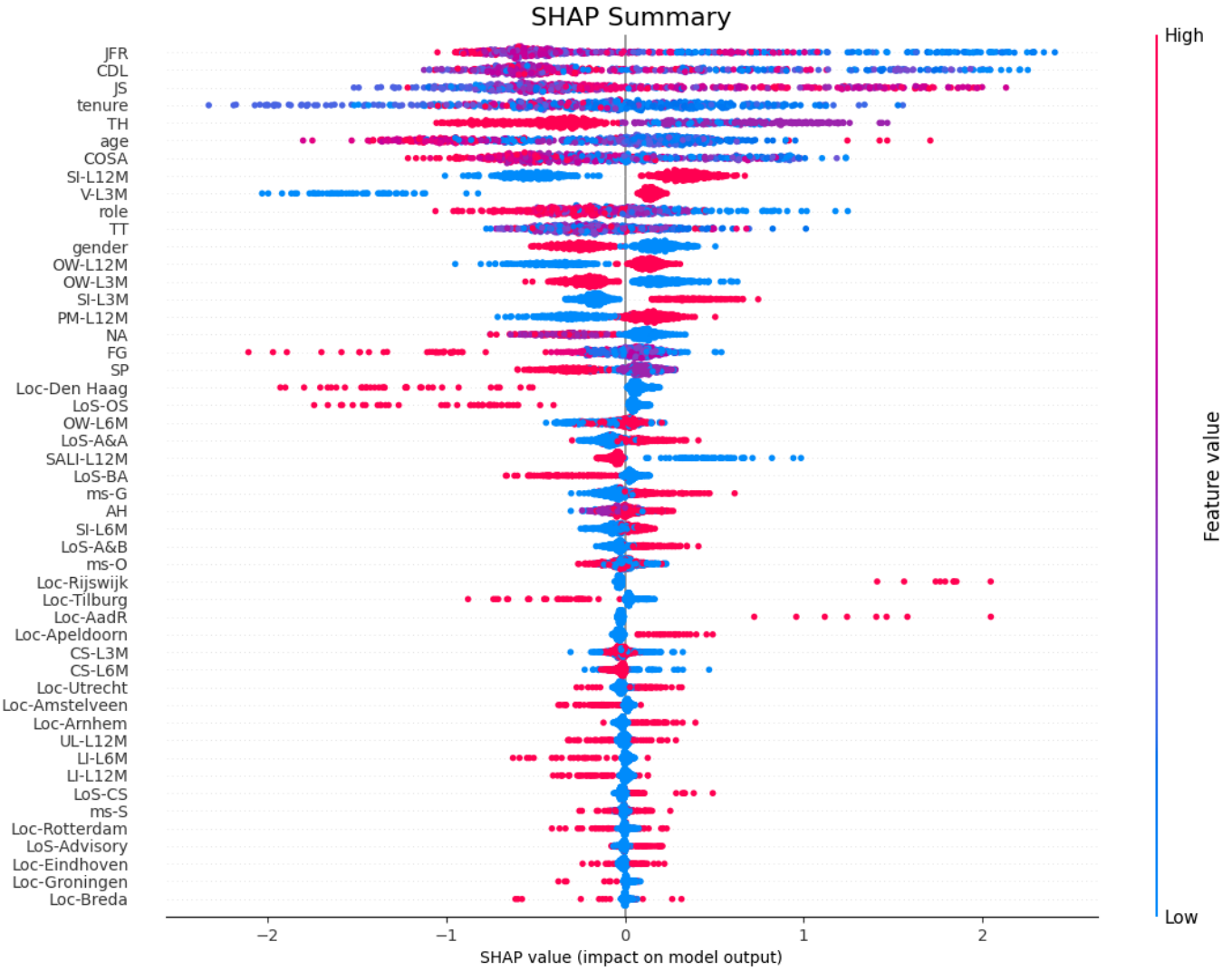
When dealing with employee turnover, it is important to identify the employees at risk so that the HR-department and the supervisor are aware of the turnover intentions of specific employees. However, the knowledge that an employee will leave the company is only part of the solution, as the sole identification will not retain the employee. Therefore, in this section, a closer look is taken at how the different features contribute to turnover and non-turnover classification for the test data. These insights on general turnover intentions amongst employees, can be used for global retention strategies.

In order to interpret the final model from chapter 8, the following steps are taken. The dataset is split into a stratified train (80%) and test (20%) set, after which the aforementioned model is trained on the train data and classifications for turnover are made using the test set. Thereafter, the SHAP values for each feature for each individual classification are calculated. A positive SHAP value indicates that a feature has a contribution to a turnover classification, whereas a negative SHAP value contributes to a non-turnover classification. Moreover, the size of the SHAP value indicates the impact of the feature on the final prediction in log odds. The results of this analysis are shown in Figure 9.1.

To elaborate, Figure 9.1 shows all features included in the final model on the y-axis and the corresponding SHAP values on the x-axis for each individual classification. Negative SHAP values have a negative contribution to turnover, or in other words, a positive contribution to retention. Positive SHAP values have a positive contribution to turnover. Moreover, the original values of the individual classifications are represented with a color, ranging from low (blue) to high (red). Some features are abbreviated, the full names of these features can be found in the list of abbreviations. The features are ranked on their importance as determined by SHAP, with the most important feature highest on the y-axis and the least important lowest.

When interpreting the model to use it for employee retention purposes, it is important to make the distinction between two types of features. The difference between these two types of features is whether they can be influenced by the company or not. Examples of features that can be influenced are salary increase in the last 12 months (SALI-L12M), and job freedom & responsibility (JFR). Examples of features that cannot be influenced are features that describe an employee, such as demographics. These features are important for the background of the employee, but cannot be used for retention. For this reason, the focus of the interpretation is on features that the company can actively use for intervention.

Figure 9.1
Overview of the SHAP values when interpreting the test data



Note. Negative SHAP values contribute to retention, whereas positive SHAP values contribute to turnover.

Four features are included in the model that the company uses to measure employee satisfaction and working environment. Currently, those features are used by the company to monitor the different departments at different locations and make adjustments when needed. For this reason, these features can be very suitable for intervention strategies, as they are already used as a guide for change. Two of these features are the job freedom & responsibility (JFR), and culture diversity & leadership (CDL). When looking at Figure 9.1, it becomes clear that a low JFR or CDL is generally associated with a positive contribution to the turnover of an employee. Moreover, a high JFR or CDL, does not guarantee a positive contribution to retention, as indicated by the high valued features scattered along the whole x-axis. For the company satisfaction (COSA) feature this contribution is more straightforward. A high COSA has a positive effect on retention, whereas lower values are associated with a lower effect on retention or even a contribution to turnover. Lastly, the job searching (JS) feature has a similar effect on turnover. When JS is low, and thus not many employees in a department have been looking for a new job, a positive effect on retention is found. Contrastingly, a high value for JS does not always lead to turnover, as shown by the high

valued features both on the left and right side of the center. To conclude, for all four features, a low value has a clear interaction with either retention (JS) or turnover (JFR, CDL and COSA). When the feature values are high, the contribution is not straightforward and differs on a case by case basis.

In the dataset, three features associated with overwork are included. Overwork in the last three months (OW-L3M), overwork in the last 6 months (OW-L6M) and overwork in the last 12 months (OW-L12M), each indicating if an employee did (1) or did not (0) work overtime in the last 3, 6 or 12 months. When looking at these three features in Figure 9.1 some interesting effects become clear. To start, when an employee worked overtime in the last three months, and thus the values for OW-L3M are high, a positive effect on employee retention is found. This finding indicates that recent overwork shows that the employee is committed to the company. Moreover, when inspecting OW-L6M and OW-L12M this effect seems to flip, with overwork having a positive effect on turnover instead of retention. Additionally, for OW-L12M the effect on turnover seems to be more pronounced and severe as opposed to OW-L6M, showing that the positive relationship between overwork and turnover becomes stronger when a longer aggregation period is used.

As mentioned in section 6.2, financial compensation and growth are identified in the literature as an important tool for retention. Therefore, it is important to investigate the effects of related features on turnover and retention in this specific dataset. To start, whenever an employee received a promotion in the last 12 months (PM-L12M) a positive effect on turnover is found, whereas no promotion is associated with a positive effect on retention. This finding can be explained by the fact that senior employees that are with the company for a long time, cannot grow further as they hit their, so-called, professional ceiling. Employees that already received a promotion, could be inclined to use this promotion to negotiate a better contract at a new employer. Salary increase in the last 12 months (SALI-L12M), has exactly the opposite effect on turnover. An increase in salary has a small but positive effect on retention, whereas no salary increase contributes to turnover by a large extend. A reason for this effect could be the following. Normally employees receive a salary increase each 12 months, making it something that employees expect to happen and thus only having a small positive effect on retention. However, whenever an employee does not receive a salary increase, a strong negative signal is given to the employee, contributing to the turnover intentions. Lastly, each employee has a current salary described by their salary tier (FG) and their salary position withing this tier (SP). A high salary tier and a high salary position are both associated with a positive effect on retention, whereas no clear effect on turnover or retention is found when the salary position or tier are low. This finding indicates that highly rewarded employees are more likely to stay. To elaborate, employees enter the company on low salary positions and at the beginning of a scale, so a higher FG and SP is often associated with more senior employees, indicating that senior employees are less likely to leave than junior employees.

Lastly, five features related to employees who had to take time off because of health related issues are included in the model. These features are split into short sick leave in the last 3, 6 and 12 months (SI-L3M, SI-L6M and SI-L12M) and long sick leave in the last 6 and 12 months (LI-L6M and LI-L12M). All three short sick leave features show a clear pattern: no short illness contributes to retention, whereas sick leave contributes to turnover for all three time horizons. For the two long sick leave features, no clear effect is found: no long sick leave has a very minor contribution to either turnover or retention, whereas long sick contributes to retention in most cases but not for all cases. Generally, short and long sick can have many different causes that the company has no influence on. An exception is when the sick leave is caused by a work related issue like a burnout, in this case the company can, and should, take action.

As mentioned previously in this section, some features are included that the company cannot influence. These features, however, can still provide important insights for the company regarding general trends. To elaborate, Figure 9.1 shows that young people are more likely to leave than older people and that males are more likely to leave than females. Even though these features cannot be influenced by the company, as they are demographics, they still show what current turnover trends are. Moreover, retention strategies such as employee engagement programs for young males could still positively affect employee retention merely by making an effort to retain employees with these demographics (Pandita & Ray, 2018). Furthermore, if a company location has a positive effect

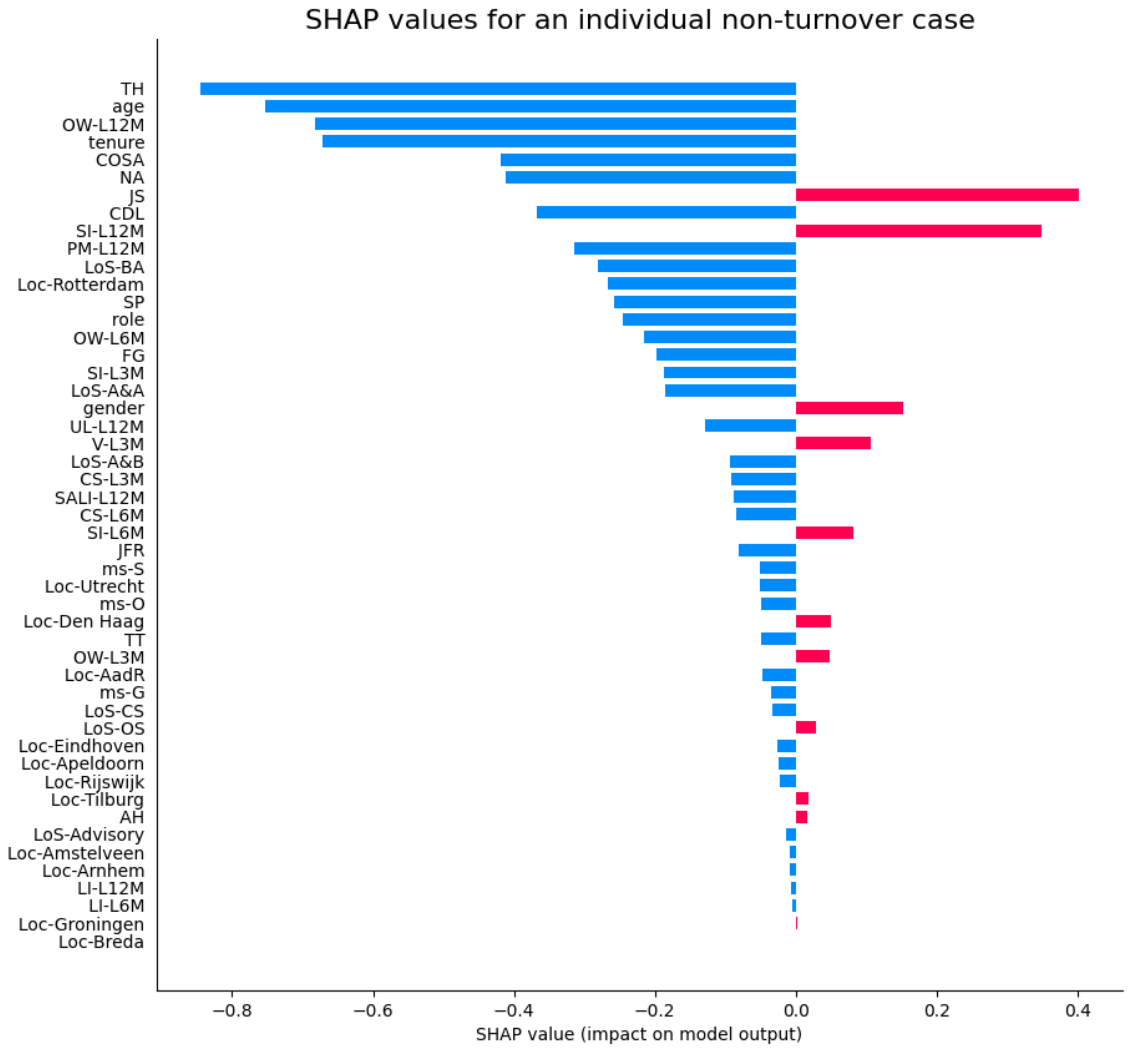
on turnover, the HR-department could investigate this location for potential problems leading to increased turnover. To conclude, features that cannot be influenced by the company, can still be used to provide insights in turnover trends, guiding general retention strategies.

9.2 Individual Interpretation

The global turnover trends discussed in section 9.1 yield important guidelines for global retention strategies. However, since the model identifies individuals that are at risk of turnover, individual retention strategies can be employed as well. Depending on the case, individual retention interventions might be preferred, as they are specialized exactly for the needs of the employee.

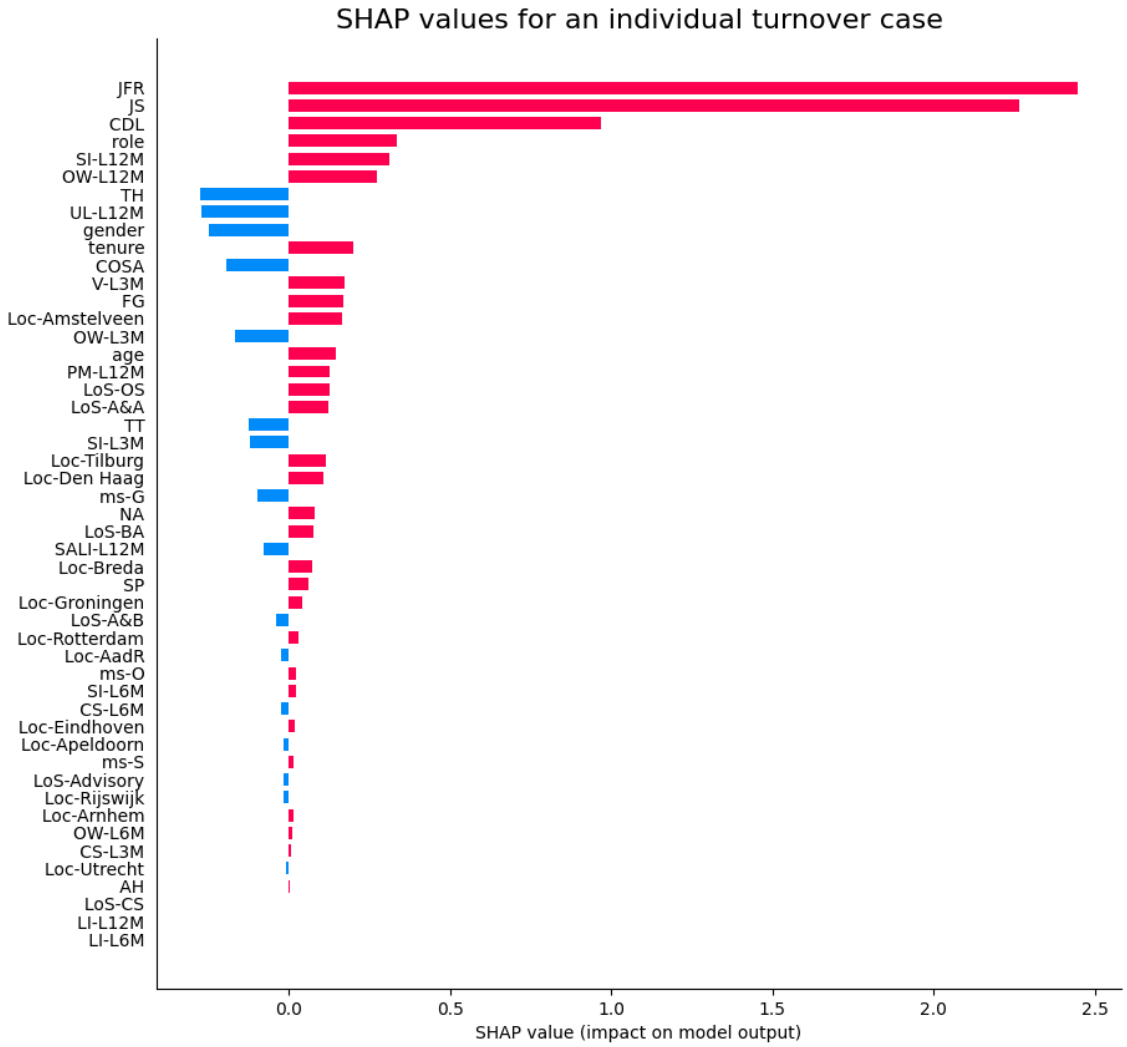
To illustrate how SHAP can be used to gain insights into employee turnover, the SHAP values for a non-turnover and turnover case are shown in figures 9.2 and 9.3 respectively. In these figures the features ranked from most to least important are shown on the y-axis together with the corresponding SHAP values in log odds the on x-axis. Blue bars contribute to retention whereas red bars contribute to turnover, with no correspondence to the actual feature values. Moreover, each bar corresponds to an individual point plotted in Figure 9.1.

Figure 9.2
Example of the individual feature interpretation for an employee that does not leave the company



When looking at the non-turnover case shown in Figure 9.2 it becomes clear that four features have a major contribution to the retention of this employee. Namely, its total hours (TH), age, overwork in the last 12 months (OW-L12M) and tenure. Moreover, 10 features have some contribution to turnover, but not enough to fully classify this person at risk of turnover.

Figure 9.3
Example of the individual feature interpretation for an employee that does leave the company



When looking at the turnover case shown in Figure 9.3 it can be seen that three features have a major contribution to the turnover of this employee. Specifically, its job freedom & responsibility (JFR), job searching (JS), and culture diversity & leadership (CDL). Moreover, 16 features have some contribution to retention, but not enough to fully classify this person at no risk of turnover.

When using this information for retention strategies, supervisors or the HR-department can investigate the individual features that are identified as drivers for turnover of the employee. Features with a high positive SHAP value have the highest potential to change the outcome of a classification and are, therefore, the most important to focus on. As a next step, they can relate these feature to the trend in the global population (Figure 9.1), to identify if the effect is part of a more global problem. Moreover, they can inspect the actual values of the features and compare those to company standards or direct colleagues. OW-L12M for example, is a driver for retention for the non-turnover employee, whereas it is a driver for turnover for the turnover employee. If

both employees in the example were to be direct colleagues, overwork hours could be compared and problems could be identified.

To conclude, the global and individual insights can help supervisors and the HR-department to identify potential problems with a department, individual, location or the whole company. Since the supervisors and HR-department are experts on their employees and the working environment, they can use these insights to implement interventions that they deem appropriate for the specific situation.

Chapter 10

Conclusion

This chapter discusses the overall conclusion of this thesis. Specifically, section 10.1 presents the answers to the main research question and the four sub-research questions. Section 10.2 discusses the relevance of the thesis with respect to the scientific contributions and the company relevance. Section 10.3 highlights the ethical considerations that should be taken into account in the context of this thesis. Section 10.4 elaborates on the limitations that this this faces and section 10.5 presents some direction for future research based on the findings in this thesis.

10.1 Key Findings

As stated in section 2.2, the goal of this thesis is to provide the company with insights into the turnover problem by developing a machine learning model that can classify employees at risk of leaving the company. Moreover, the goal is to provide means to interpret this model, so that actionable insights can be obtained for individual and global retention strategies. In order to obtain this goal, a main research question was formulated together with several sub-research questions in sections 2.3 & 2.4. In this section, the findings of the thesis are related to each of the sub-research questions, after which these findings are related to the main research question.

1. What are the current costs associated with employee turnover and employee retention?

This section has been removed for confidentiality reasons.

2. How to identify future turnover and its causes using machine learning techniques?

To identify future employee turnover using historical employee turnover data, some steps need to be taken. To start, a dataset needs to be aggregated using the last 13 period entries of each employee in the dataset, after the removal of unwanted entries. Within these last 13 periods, 12 periods are used as training data, whereas the last period is used to determine if an employee left the company or not. This dataset can be optimized by removing unnecessary features, that create noise or simply have no meaningful contribution to the final prediction. Using this optimized dataset, an extreme gradient boosting model can be trained, as this model proves to provide the best performance out of all six models that are evaluated. In order to balance the false negative and false positive classifications of this model, a threshold of 0.283 can be used.

Lastly, since the extreme gradient boosting model is a black box model, the causes for turnover can be identified using SHAP. This algorithm and the causes for turnover are discussed in detail in sub-research question 4.

3. How do models trained on pandemic data and non-pandemic data compare?

Models trained on pandemic and non-pandemic data have a large difference in performance. Specifically, a model trained on non-pandemic data has a higher performance than one trained on

pandemic data. Moreover, performance of the model trained on pandemic data seems to improve when more non-pandemic data is introduced. Therefore, it can be concluded that the pandemic dataset might have an insufficient amount of training data due to the short period it covers. Additionally, introducing non-pandemic data into the pandemic dataset provides the model with a greater variety of turnover samples, which improves its classification capabilities. Therefore, a model is trained on the full dataset, resulting in a vast improvement in performance over both the pandemic and non-pandemic datasets. To conclude, splitting the dataset reduces performance in both splits with the largest performance drop in the split with the least amount of samples. Thus, using the full dataset without split is preferred.

4. How can supervisors and the HR-department interpret the output of the model, such that it can be used for intervention strategies?

Supervisors and the HR-department can use the output of the model together with the corresponding SHAP values for global and individual intervention strategies. For global retention strategies, they can use the classifications of the model, to get information on who is going to leave to company. Additionally, they can use the global summary provided by SHAP to investigate global drivers for turnover. Subsequently, they can implement measures that target these drivers in order to increase retention.

On an individual level, supervisors and the HR-department can investigate each turnover classification for its main drivers. Specifically, when a supervisor gets notified that someone in one of his teams might leave, he can use the drivers provided by SHAP for intervention strategies that are tailored to the individual. An individual intervention can have different effects than a global intervention, as the intervention is catered to the exact needs of the employee. Moreover, when the intervention is implemented by a supervisor that knows the employee, the supervisor can employ his own expertise to accurately create an appropriate intervention for the employee.

Lastly, because SHAP values are additive, supervisors and the HR-department can also easily add the individual SHAP values of different sub-groups to gain specific insights in areas they deem important. This property makes the output of SHAP very flexible and gives the people working with the data the specific insights they need.

To conclude, the model identifies employees at risk of turnover and provides insights into the reasons for turnover using SHAP. These insights can be used to either target a group of employees with a global intervention or specific employees with an individual intervention. Both intervention strategies can be used in tandem or alone, based on what the supervisors and HR-department deem necessary.

How to gain insights into future turnover and apply these insights to reduce employee turnover costs using historical employee turnover data?

The insights gained in answering each of the four sub-research questions can be used to answer the main research question. Future turnover can be predicted using an extreme gradient boosting model trained on the last 13 periods of historic data of each employee, without distinguishing between pandemic and non-pandemic data. This model classifies the majority of turnover and non-turnover employees correctly, with a small number of misclassifications. Insights on these classifications can be gained using SHAP, which shows the impact of each feature on the final classification. Supervisors and the HR-department can use the classifications and feature contributions for intervention strategies on both a global and individual level to increase employee retention.

Total retention costs for the last ~5 years are higher than turnover costs. This difference is rooted in the fact that the costs for retention are incurred for every non-temporary employee, whereas turnover costs are only incurred for employees that leave the company. Moreover, turnover costs are currently nearly eight times as high as retention costs when looking at a single employee.

An overall cost reduction can be achieved by using the classifications and interpretations provided by the model, to specifically target intervention strategies at groups or individuals that

are at risk of turnover. This strategy will increase the overall retention costs, since most interventions are of a financial nature. However, this strategy will result in an overall cost reduction for the following reasons. Firstly, the interventions have more potential to actually convert the employee, as they are informed and based on insights provided by the model instead of the current generic interventions. Secondly, for each employee that is retained, a saving on the turnover costs is made. However, some employees at risk are not identified by the model, which will incur turnover costs that cannot be negated. Furthermore, it is assumed that retention strategies are employed in a way that the employee is unaware of the classification, so that undesired side effects such as placing the idea of leaving in the employees mind are avoided.

10.2 Relevance

In order to place this thesis into the larger theoretical landscape, the scientific contributions are discussed in this section. Moreover, the relevance of the thesis for the company is discussed to highlight the practical implications of this thesis.

10.2.1 Scientific Contribution

This thesis contributes to the scientific literature in a variety of ways. The first contribution to the literature is a novel holistic approach to the employee turnover problem. Where previous research only investigated a specific part of the employee turnover problem, this thesis unifies the costs, identification and interpretation of turnover into one structured holistic approach. Moreover, the implementation of SHAP for turnover interpretation on a global and individual level is completely novel.

The second contribution has to do with some minor methodological contributions when applying machine learning models to the employee turnover problem. Current papers in this research area terminate the modeling process when finding the optimal model according to the AUC_{Pr} or the ROC-curve. However, in this thesis: two more steps are introduced, threshold tuning and relating the model back to an actual confusion matrix. Threshold tuning further increases model performance and the confusion matrix gives practical insights into real world model performance.

The third contribution is the validation and generalization of insights obtained in previous research. Firstly, this thesis expands the findings of previous research to the financial industry using novel source data. Moreover, the finding that tree based models perform well on turnover data is generalized, with a nearly identical model ranking as found in the literature review. The only exception to this ranking is the AdaBoost model, which performed better than found in the literature. Secondly, figures on turnover and retention costs are provided together with turnover rates. Thirdly, the questionnaire proposed by McKinney et al. (2007) is validated through its implementation in this thesis.

The fourth contribution deals with the insights into the influence of the pandemic on the employee turnover dataset. Currently, no papers give insight into dealing with pandemic and non-pandemic turnover data. This thesis shows that using a combination of pandemic and non-pandemic data, improves model performance as opposed to using two separate models trained on different datasets.

The fifth and last contribution is of a more practical nature. Currently, the methodology used is applied to a business case at a specific company. However, the methodology and takeaways provided in this thesis serve as a backbone for similar research in other areas and are easily adaptable to different contexts.

10.2.2 Company Relevance

This research is relevant for the company in a variety of ways. Firstly, the research provides insights into two costs figures that were previously unknown to the company, the costs of retention and turnover. Specifically, insights are provided into employee retention and turnover costs on a global

and individual level and a structured method is proposed to measure them. These insights build a strong business case for the company, showing a clear need for cost reduction with respect to employee turnover costs.

Secondly, this thesis provides a structured approach on implementing a machine learning model for the classification of employees at risk of turnover. Specifically, the company can recreate the steps taken in this research to implement an actual machine learning model into their workflow. Thirdly, insights into global turnover trends are presented in this thesis. Additionally, guidelines on how to interpret these trends on an individual and global level are provided, so that they can be applied to a machine learning model in practice.

Fourthly, some examples are presented on how the insights obtained by model interpretation can be used to achieve a turnover cost reduction. Moreover, some guidelines when dealing with the implementation of retention strategies are discussed.

To illustrate, the ideas presented in this research can be used in practice as follows: to start, the company can continuously measure turnover and retention costs using the methods presented in the thesis. These measurements will provide insights into the impact of the model in practice and enhance the currently known cost figures. Moreover, the company can implement the model presented in this thesis to classify employees at risk of turnover. Whenever an employee is classified as at risk of turnover, the HR-department and the supervisor are notified of an employee at risk. Subsequently, they can use SHAP to gain insights into this specific employee, and make efforts to retain the employee. Lastly, SHAP can be used as a tool to monitor global turnover trends and those insights can be used for global retention strategies.

10.3 Ethical Considerations

The dataset used consists of personal information, which is very sensitive in nature. In order to prevent any breaches of personal data, the following measures were taken. Firstly, all data was pseudonymised so that it was not possible to trace back to actual employees. To elaborate, employee ID's were modified, salary numbers were converted to bins and employee locations were converted to travel times. Secondly, all processing was done in the cloud, so that no sensitive data left the company's database, minimizing the risk for any possible data breaches.

Another ethical consideration has to do with the potential future usage of the model discussed in this thesis. A model that predicts turnover multiple periods ahead could theoretically be used to evaluate the potential tenure of applicants in a solicitation process. Applicants with a short potential tenure could be refused based on model classification alone. For the current state of the model, this misuse is not possible because the model requires at least 13 periods of data. Besides, not all features that the model requires are known to the company in the application phase of the potential employee. However, when the model is developed further in the future, this issue requires careful handling, as it would be unethical to refuse employees based on potential tenure alone.

10.4 Limitations

Despite that the research in this thesis was conducted with attention to detail and scientific integrity, still some limitations apply. These limitations are highlighted in this section.

The first limitation has to do with the calculation of the total cost reduction associated with employee turnover reduction. With the current insights some global conclusions can be made, however, these insights are limited by some essential data which is missing. In order to accurately estimate the cost reduction that can be achieved per employee, data is needed on the conversion rates of various retention strategies and the effect of a retention budget on these conversion rates. Moreover, it is likely that not every employee can be retained, irrespective of the efforts made by the company. Applying the costs of retention and the corresponding conversion rates to the output of the model will provide more detailed insights into the overall cost reduction.

Related to the limited insights into the conversion rates and retention costs of individual employees, are limited insights into the turnover costs on an individual level. In this thesis estimates of the costs of turnover for different employees are aggregated into a single turnover costs figure all employees. The reason for the usage of this aggregated cost figure is that it is not possible to accurately determine the costs of each employee that left the company in the last five years. In practice however, different employees have different turnover costs depending on their seniority level and role within the company.

Another limitation is that the model predictions are memoryless. Whenever an employee is classified as a turnover case, the company will likely invest in the retention of this employee. However, the interventions implemented by the company to retain the employee might not have an instant effect. When new data becomes available in the next period, the model makes new classifications and could classify the same employee as a potential turnover case again. Executing new retention efforts based on this classification is possibly a waste of resources, since the results of retention strategies could be subject to delay.

A limitation with respect to the future usability of the model is also encountered. This thesis relies on the assumption that past turnover behavior is representative for future turnover behavior. It is likely that this assumption holds for the near future, since the company is very stable. However, potential business decisions such as a restructuring of the company, and diversification into new markets, could change the company culture in such a way that past data is no longer representative for future turnover.

Lastly, three limitations with respect to the modeling and interpretation are present. Firstly, cross-validation for hyperparameter tuning is limited by computational power and time. When more time or computational power is available, more combinations of hyperparameters can be evaluated with more validation iterations. Subsequently, evaluating more hyperparameter combinations could potentially lead to undiscovered parameters with higher performance. Secondly, RFECV is employed for feature selection, which is inherently unstable in its output. Specifically, the output of RFECV can change depending on the order in which the features are dropped, for the reasons explained in section 3.3. This effect is minimized by using a large amount of cross-validations, but without evaluating all different feature permutations, some instability will always be present. Thirdly, the SHAP plots in chapter 9 present the SHAP values of the features in log odds. Actual probabilities would be more helpful when relating the values to insights, however, the current implementation of SHAP does not support this.

10.5 Future Research

In addition to the key findings and the limitations discussed in the previous sections, this thesis also revealed some suggestions for future research. This section discusses these suggestions.

As mentioned in section 10.4, in order to accurately estimate the cost reduction that can be achieved, more insights are needed. Specifically, future research could investigate the effects of different retention budgets on the probability to retain an employee. Additionally, this budget could be related to the various retention strategies evaluated in previous research.

Related to the research area mentioned in the previous paragraph, future research could focus on optimizing the retention budget for individual employees. To elaborate, as mentioned in section 10.4, depending on the type of employee different turnover costs are incurred. To address this imbalance, future research could focus on establishing a balance between this individual turnover cost and a proposed retention budget. A retention budget which is higher than the costs of turnover would not make sense from a financial perspective.

Furthermore, this research proposes a novel method of threshold tuning based on the size of turnover and retention costs. In its current state, the method failed to provide a useful model. However, further development of this proposed method in future research could yield interesting model results when optimizing a model for the balance between turnover and retention costs.

Another area of future research is related to how the research problem is approached. Due to the nature of employee turnover data, a time aspect is present. Specifically, employees stay

at the company for a certain period of time, after which they may decide to leave. Because of these characteristics, this problem could also be approached from a survival analysis perspective. Future research could investigate the merits of this approach.

Lastly, section 10.4 mentions the limitations associated with the RFECV methods for feature selection. The problems related to the instability of the methods caused by the order in which the features are dropped can be addressed using SHAP values for feature importance. Experimental insights into applying SHAP to RFECV are provided by Garbacz (2020), however, no scientific research has been conducted on this method yet. In case such research arises in the future, new opportunities for more reliable feature selection might present themselves.

Bibliography

Baldi, P., Brunak, S., Chauvin, Y., Andersen, C. A. F. & Nielsen, H. (2000). Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, *16*(5), 412–424. https://doi.org/10.1093/bioinformatics/16.5.412

Bergstra, J. & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, *13*(10), 281–305. http://jmlr.org/papers/v13/bergstra12a.html

Boswell, W. R., Ren, L. R. & Hinrichs, A. T. (2008). Voluntary employee turnover: Determinants, processes, and future directions. *The SAGE handbook of organizational behavior: Volume i - micro approaches* (pp. 196–216). SAGE Publications Ltd. https://doi.org/10.4135/9781849200448.n12

Breiman, L., Friedman, J. H., Olshen, R. A. & Stone, C. J. (1984). *Classification and regression trees*. Wadsworth; Brooks.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Cascio, W. F. (1991). *Managing human resources: Productivity, quality of work life, profits* (3rd ed.). New York: McGraw Hill.

Chawla, N. V. (2010). Data mining for imbalanced datasets: An overview. In O. Maimon & L. Rokach (Eds.), *Data mining and knowledge discovery handbook* (pp. 875–886). Springer US. https://doi.org/10.1007/978-0-387-09823-4_45

Chen, T. & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Cunningham, P. (2008). Dimension reduction. In M. Cord & P. Cunningham (Eds.), *Machine learning techniques for multimedia: Case studies on organization and retrieval* (pp. 91–112). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-75171-7_4

Das, B. & Baruah, M. (2013). Employee retention: A review of literature. *IOSR Journal of Business and Management*, *14*, 08–16. https://doi.org/10.9790/487X-1420816

El-Khatib, K. (2010). Impact of feature reduction on the efficiency of wireless intrusion detection systems. *IEEE Transactions on Parallel and Distributed Systems*, *21*(8), 1143–1149. https://doi.org/10.1109/TPDS.2009.142

El-Rayes, N., Fang, M., Smith, M. & Taylor, S. (2020). Predicting employee attrition using tree-based models. *International Journal of Organizational Analysis*, *28*(6), 1273–1291. https://doi.org/10.1108/IJOA-10-2019-1903

Esmaieeli Sikaroudi, A., Ghousi, R. & Sikaroudi, A. (2015). A data mining approach to employee turnover prediction (case study: Arak automotive parts manufacturing). *Journal of Industrial and Systems Engineering*, *8*(4), 106–121. http://www.jise.ir/article_10857.html

Fallucchi, F., Coladangelo, M., Giuliano, R. & William De Luca, E. (2020). Predicting employee attrition using machine learning techniques. *Computers*, *9*(4). https://doi.org/10.3390/computers9040086

Freund, Y. & Schapire, R. (1999). A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, *14*(771-780), 1612.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*(5), 1189–1232. https://doi.org/10.1214/aos/1013203451

Gabrani, G. & Kwatra, A. (2018). Machine learning based predictive model for risk assessment of employee attrition. *Lecture Notes in Computer Science*, *10963*, 189–201. https://doi.org/10.1007/978-3-319-95171-3_16

Gao, X., Wen, J. & Zhang, C. (2019). An improved random forest algorithm for predicting employee turnover. *Mathematical Problems in Engineering*, *2019*, 1–12. https://doi.org/10.1155/2019/4140707

Garbacz, M. (2020). *Open-sourcing shaprfecv — improved feature selection powered by shap*. Retrieved August 11, 2021, from https://medium.com/ing-blog/open-sourcing-shaprfecv-improved-feature-selection-powered-by-shap-994fe7861560

He, H. & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, *21*(9), 1263–1284. https://doi.org/10.1109/TKDE.2008.239

Hinkin, T. R. & Tracey, J. B. (2000). The cost of turnover: Putting a price on the learning curve. *Cornell hotel and restaurant administration quarterly*, *41*(3), 14–21.

Inoue, A., Jin, L. & Rossi, B. (2017). Rolling window selection for out-of-sample forecasting with time-varying parameters. *Journal of Econometrics*, *196*(1), 55–67. https://doi.org/10.1016/j.jeconom.2016.03.006

Jain, P., Jain, M. & Pamula, R. (2020). Explaining and predicting employees' attrition: A machine learning approach. *SN Applied Sciences*, *2*. https://doi.org/10.1007/s42452-020-2519-4

Khera, S. N. & Divya. (2019). Predictive modelling of employee turnover in indian it industry using machine learning techniques. *Vision: The Journal of Business Perspective*, *23*, 12–21. https://doi.org/10.1177/0972262918821221

Krstajic, D., Buturovic, L. J., Leahy, D. E. & Thomas, S. (2014). Cross-validation pitfalls when selecting and assessing regression and classification models. *Journal of Cheminformatics*, *6*(1), 10. https://doi.org/10.1186/1758-2946-6-10

Lundberg, S. M. & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems 30*, 4765–4774. http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

Ma, X., Zhai, S., Fu, Y., Lee, L. & Shen, J. (2019). Predicting the occurrence and causes of employee turnover with machine learning. *Computer Engineering and Applications*, *8*, 217–227. https://doi.org/10.18495/COMENGAPP.V8I3.316

McKinney, W., Bartlett, K. & Mulvaney, M. (2007). Measuring the costs of employee turnover in illinois public park and recreation agencies: An exploratory study. *Journal of Park and Recreation Administration*, *25*(1). https://js.sagamorepub.com/jpra/article/view/1370

Mishra, S. (2015). Nonlinear system identification using functional link multilayer perceptron artificial neural networks. *Nonlinear System Identification Using Functional Link Multilayer Perceptron Artificial Neural Networks*, *10*, 31542–31546.

Misra, P. & Yadav, A. S. (2020). Improving the classification accuracy using recursive feature elimination with cross-validation. *Int. J. Emerg. Technol*, *11*(3), 659–665.

Mitrovska, S. & Eftimov, L. (2016). Calculating the cost for employee turnover in the it industry in macedonia by using a web calculator. *Journal of Human Resource Management*, *19*, pp.24–33.

Monisaa Tharani, S. K. & Vivek Raj, S. N. (2020). Predicting employee turnover intention in it ites industry using machine learning algorithms. *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 508–513. https://doi.org/10.1109/I-SMAC49090.2020.9243552

Pandita, D. & Ray, S. (2018). Talent management and employee engagement – a meta-analysis of their impact on talent retention. *Industrial and Commercial Training*, *50*(4), 185–199. https://doi.org/10.1108/ICT-09-2017-0073

Patel, A., Pardeshi, N., Patil, S., Sutar, S., Sadafule, R. & Bhat, S. (2020). Employee attrition predictive model using machine learning. *International Research Journal of Engineering and Technology (IRJET)*, *7*(5).

Pinkovitz, W. H., Moskal, J. & Green, G. (1997). How much does your employee turnover cost. *Small Business Forum*, *14*(3), 70–71.

Punnoose, R. & Ajit, P. (2016). Prediction of employee turnover in organizations using machine learning algorithms. *International Journal of Advanced Research in Artificial Intelligence*, *5*(9). https://doi.org/10.14569/IJARAI.2016.050904

Ribeiro, M. T., Singh, S. & Guestrin, C. (2016). "why should i trust you?": Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144. https://doi.org/10.1145/2939672.2939778

Rokach, L. & Maimon, O. (2005). Decision trees. In O. Maimon & L. Rokach (Eds.), *Data mining and knowledge discovery handbook* (pp. 165–192). Springer US. https://doi.org/10.1007/0-387-25465-X_9

Sá, J., Almeida, A., Pereira da Rocha, B., Mota, M., De Souza, J. R. & Dentel, L. (2016). Lightning forecast using data mining techniques on hourly evolution of the convective available potential energy, 1–5. https://doi.org/10.21528/CBIC2011-27.1

Saito, T. & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, *10*(3), 1–21. https://doi.org/10.1371/journal.pone.0118432

Sisodia, D. S., Vishwakarma, S. & Pujahari, A. (2017). Evaluation of machine learning models for employee churn prediction. *2017 International Conference on Inventive Computing and Informatics (ICICI)*, 1016–1020. https://doi.org/10.1109/ICICI.2017.8365293

Speiser, J. L., Miller, M. E., Tooze, J. & Ip, E. (2019). A comparison of random forest variable selection methods for classification prediction modeling. *Expert Systems with Applications*, *134*, 93–101. https://doi.org/10.1016/j.eswa.2019.05.028

Tett, R. P. & Meyer, J. P. (1993). Job satisfaction, organizational commitment, turnover intention, and turnover: Path analyses based on meta-analytic findings. *Personnel Psychology*, *46*(2), 259–293. https://doi.org/10.1111/j.1744-6570.1993.tb00874.x

Tziner, A. & Birati, A. (1996). Assessing employee turnover costs: A revised approach. *Human Resource Management Review*, *6*(2), 113–122. https://doi.org/10.1016/S1053-4822(96)90015-7

Vasa, J. & Masrani, K. (2019). Foreseeing employee attritions using diverse data mining strategies. *International Journal of Recent Technology and Engineering (IJRTE)*, *8*(3), 620–626. https://doi.org/10.35940/ijrte.B2406.098319

Wirth, R. & Hipp, J. (2000). Crisp-dm: Towards a standard process model for data mining. *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining*, *1*, 29–39.

Witten, I. H., Frank, E. & Hall, M. A. (2011). *Data mining: Practical machine learning tools and techniques* (Third Edition). Morgan Kaufmann. https://doi.org/10.1016/C2009-0-19715-5

Yadav, S., Jain, A. & Singh, D. (2018). Early prediction of employee attrition using data mining techniques. *2018 IEEE 8th International Advance Computing Conference (IACC)*, 349–354. https://doi.org/10.1109/IADCC.2018.8692137

Zhao, Y., Hryniewicki, M., Cheng, F., Fu, B. & Zhu, X. (2019). Employee turnover prediction with machine learning: A reliable approach. https://doi.org/10.1007/978-3-030-01057-7

Appendix A

Questionnaire

A.1 Questionnaire Turnover Cost

1. How many hours did you spend on the exit of the employee?

2. What where the administrative expenses related to the exit of the employee?

3. What was the loss of productivity in hours caused by the exit of the employee?

4. How many hours of overwork for colleagues were caused by the exit of the employee?

5. Has the employee that left already been replaced?

   Yes: Go to 6

   No: Go to 10

6. How many hours did you spend on screening new candidates?

7. Did an employee receive a recruitment bonus for recruiting the new employee?

8. Did the new employee receive a relocation allowance?

9. What were the costs associated with formal and informal training of the new employee?

   Go to: 12

10. How many hours do you estimate to spend on screening new candidates?

11. What do you estimate the costs associated with formal and informal training of the new employee are going to be?

    Go to: 12

12. What is the turnover loss associated with customers that left together with the employee?

A.2 Original Questionnaire Turnover Cost

61

Figure 3: Public Park and Recreation Costing Employee Turnover Survey

Human Resource Turnover

Please answer the following questions with the closest possible figure that represents the expense incurred by your agency. If no expense was incurred during your latest turnover, please respond N/A.

Separation Costs Associated with Departing Employee

1.1 Separation Pay
- 1.11 Terminal vacation pay $_____
- 1.12 Unused sick pay $_____
- 1.13 Payment for other fringe benefits $_____

1.2 Cost of time associated with exit interviews (hours x pay rate) $_____

1.3 Administrative costs associated with exit $_____

1.4 Loss of productivity (estimate) $_____

1.5 Overtime to existing staff required before replacement staff hired $_____

Costs Associated with Replacing Employee

2.1 Advertising for replacement $_____

2.2 Time for management to screen applications/resumes (hours x pay rate) $_____

2.3 Interviews
- 2.31 Number of candidates interviewed _____ candidates
- 2.32 Number of staff hours involved in interview process _____ hours
- 2.33 Staff hourly rate of pay $_____
- 2.34 Total cost of staff time for interviews $_____

2.4 Miscellaneous interview expenses (e.g., transportation, accommodations, meals) $_____

2.5 Reference/background checks
- 2.51 Number of hours to check references/background checks _____hours
- 2.52 Hourly rate of pay for individual responsible for completing reference/background checks $_____
- 2.53 Total cost of staff time for reference checks $_____

2.6 Cost of pre-employment testing (drug testing, etc.) $_____

2.7 Cost of appointment for new hire
- 2.71 Additional incentives $_____
- 2.72 Moving costs $_____

Costs Associated with Placing/Training New Employee

3.1 Cost of training (literature, formal and informal) $_____

Thank you for your time and responses.

Source: McKinney et al. (2007)

Appendix B

Model Performance Benchmarks

B.1 Non-pandemic dataset full results

Table B.1
Full results for the dataset selection non-pandemic

Aggregation Type	R	y	Acc	σ_{acc}	Pr	σ_{Pr}	Rc	σ_{Rc}	F1	σ_{F1}	AUC_{Pr}	σ_{AUC}
Rolling Window	1	1	0.987	0.000	0.020	0.140	0.000	0.001	0.000	0.002	0.134	0.021
Rolling Window	1	2	0.987	0.000	0.100	0.300	0.001	0.002	0.001	0.004	0.139	0.023
Rolling Window	1	3	0.987	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.132	0.020
Rolling Window	1	4	0.987	0.000	0.080	0.271	0.001	0.002	0.001	0.003	0.146	0.025
Rolling Window	2	1	0.987	0.000	0.060	0.237	0.000	0.001	0.001	0.003	0.154	0.030
Rolling Window	2	2	0.987	0.000	0.200	0.400	0.001	0.003	0.003	0.006	0.148	0.022
Rolling Window	2	3	0.987	0.000	0.020	0.140	0.000	0.001	0.000	0.002	0.151	0.034
Rolling Window	2	4	0.987	0.000	0.180	0.384	0.001	0.003	0.003	0.006	0.148	0.022
Rolling Window	3	1	0.987	0.000	0.050	0.206	0.000	0.001	0.001	0.003	0.167	0.026
Rolling Window	3	2	0.987	0.000	0.040	0.196	0.000	0.001	0.001	0.002	0.167	0.027
Rolling Window	3	3	0.987	0.000	0.040	0.196	0.000	0.002	0.001	0.004	0.164	0.027
Rolling Window	3	4	0.987	0.000	0.060	0.237	0.000	0.002	0.001	0.003	0.163	0.025
Rolling Window	4	1	0.987	0.000	0.160	0.367	0.001	0.002	0.002	0.005	0.189	0.025
Rolling Window	4	2	0.987	0.000	0.020	0.140	0.000	0.001	0.000	0.002	0.178	0.032
Rolling Window	4	3	0.987	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.183	0.028
Rolling Window	4	4	0.987	0.000	0.140	0.347	0.001	0.002	0.002	0.005	0.191	0.028
Rolling Window	5	1	0.987	0.000	0.080	0.271	0.001	0.002	0.001	0.003	0.208	0.027
Rolling Window	5	2	0.987	0.000	0.020	0.140	0.000	0.001	0.000	0.002	0.198	0.037
Rolling Window	5	3	0.987	0.000	0.200	0.400	0.001	0.003	0.003	0.006	0.198	0.036
Rolling Window	5	4	0.987	0.000	0.240	0.427	0.002	0.003	0.004	0.007	0.221	0.037
Rolling Window	6	4	0.987	0.000	0.100	0.300	0.001	0.002	0.001	0.004	0.243	0.033
Rolling Window	6	1	0.987	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.210	0.031
Rolling Window	6	2	0.987	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.213	0.036
Rolling Window	6	3	0.987	0.000	0.140	0.347	0.001	0.002	0.002	0.005	0.235	0.038
Rolling Window	7	3	0.987	0.000	0.020	0.140	0.000	0.001	0.000	0.002	0.249	0.036
Rolling Window	7	4	0.987	0.000	0.140	0.347	0.001	0.003	0.002	0.005	0.258	0.040
Rolling Window	7	1	0.987	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.239	0.040
Rolling Window	7	2	0.987	0.000	0.020	0.140	0.000	0.001	0.000	0.002	0.245	0.031
Rolling Window	8	2	0.987	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.265	0.038
Rolling Window	8	3	0.987	0.000	0.200	0.400	0.002	0.003	0.003	0.006	0.265	0.045
Rolling Window	8	4	0.987	0.000	0.180	0.384	0.001	0.003	0.003	0.006	0.301	0.042
Rolling Window	8	1	0.987	0.000	0.020	0.140	0.000	0.001	0.000	0.002	0.260	0.036
Rolling Window	9	1	0.987	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.288	0.045
Rolling Window	9	2	0.987	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.283	0.042
Rolling Window	9	3	0.987	0.000	0.300	0.458	0.003	0.004	0.005	0.008	0.320	0.037
Rolling Window	10	1	0.987	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.294	0.046
Rolling Window	10	2	0.987	0.000	0.020	0.140	0.000	0.001	0.000	0.002	0.323	0.037
Rolling Window	11	1	0.987	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.341	0.042
Rolling Window	12	1	0.987	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.377	0.049
Last Window	12	1	0.767	0.010	0.685	0.064	0.216	0.033	0.327	0.041	0.552	0.035
Last Window	12	2	0.768	0.012	0.666	0.078	0.211	0.034	0.319	0.043	0.537	0.042
Last Window	12	3	0.772	0.010	0.666	0.081	0.193	0.032	0.297	0.040	0.533	0.043
Last Window	12	4	0.775	0.011	0.655	0.081	0.191	0.039	0.294	0.050	0.519	0.043

Note. Acc = Accuracy, Pr = Precision, Rc = Recall, AUC_{PR} = Area under the precision recall curve. Next to each performance benchmark its standard deviation (σ) is given.

Table B.2
Results for undersampling the best performing non-pandemic rolling window aggregation

Agg Type	R	y	U_r	Acc	σ_{acc}	Pr	σ_{Pr}	Rc	σ_{Rc}	F1	σ_{F1}	AUC_{Pr}	σ_{AUC}
RW	12	1	0.02	0.987	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.259	0.048
RW	12	1	0.03	0.987	0.000	0.060	0.237	0.001	0.002	0.001	0.004	0.186	0.031
RW	12	1	0.04	0.987	0.000	0.040	0.196	0.000	0.002	0.001	0.003	0.134	0.025
RW	12	1	0.05	0.987	0.000	0.020	0.140	0.000	0.001	0.000	0.002	0.115	0.022
RW	12	1	0.1	0.987	0.000	0.077	0.248	0.001	0.003	0.002	0.005	0.062	0.011
RW	12	1	0.2	0.985	0.001	0.113	0.074	0.018	0.013	0.030	0.020	0.046	0.009
RW	12	1	0.4	0.955	0.006	0.051	0.013	0.141	0.033	0.074	0.017	0.039	0.009
RW	12	1	0.6	0.876	0.011	0.034	0.003	0.319	0.038	0.062	0.006	0.033	0.005
RW	12	1	0.8	0.762	0.012	0.027	0.002	0.496	0.041	0.051	0.004	0.031	0.004
RW	12	1	1	0.647	0.017	0.023	0.001	0.626	0.039	0.044	0.003	0.029	0.005

Note. RW = Rolling Window, U_r = Undersampling ratio, Acc = Accuracy, Pr = Precision, Rc = Recall, AUC_{PR} = Area under the precision recall curve. Next to each performance benchmark its standard deviation (σ) is given.

B.2 Pandemic dataset full results

Table B.3
Full results for the dataset selection pandemic

Aggregation Type	R	y	Acc	σ_{acc}	Pr	σ_{Pr}	Rc	σ_{Rc}	F1	σ_{F1}	AUC_{Pr}	σ_{AUC}
Rolling Window	1	1	0.991	0.000	0.520	0.500	0.016	0.017	0.031	0.033	0.085	0.039
Rolling Window	1	2	0.991	0.000	0.080	0.271	0.002	0.007	0.004	0.014	0.063	0.031
Rolling Window	1	3	0.990	0.000	0.200	0.400	0.006	0.011	0.011	0.022	0.070	0.030
Rolling Window	1	4	0.990	0.000	0.260	0.439	0.008	0.013	0.015	0.026	0.061	0.026
Rolling Window	2	1	0.991	0.000	0.220	0.414	0.006	0.011	0.012	0.022	0.064	0.029
Rolling Window	2	2	0.990	0.000	0.200	0.400	0.006	0.011	0.011	0.022	0.073	0.036
Rolling Window	2	3	0.990	0.000	0.360	0.480	0.011	0.015	0.021	0.028	0.071	0.027
Rolling Window	2	4	0.990	0.000	0.300	0.458	0.011	0.016	0.021	0.032	0.070	0.042
Rolling Window	3	1	0.990	0.000	0.240	0.427	0.007	0.012	0.013	0.023	0.078	0.033
Rolling Window	3	2	0.990	0.000	0.300	0.458	0.009	0.014	0.018	0.027	0.075	0.030
Rolling Window	3	3	0.990	0.000	0.380	0.485	0.014	0.017	0.026	0.034	0.071	0.039
Rolling Window	3	4	0.991	0.000	0.360	0.480	0.017	0.022	0.032	0.043	0.097	0.051
Rolling Window	4	1	0.990	0.000	0.360	0.480	0.011	0.015	0.021	0.028	0.081	0.036
Rolling Window	4	2	0.990	0.000	0.300	0.458	0.011	0.016	0.021	0.032	0.078	0.043
Rolling Window	4	3	0.991	0.000	0.320	0.466	0.015	0.022	0.028	0.042	0.087	0.040
Rolling Window	4	4	0.991	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.069	0.047
Rolling Window	5	1	0.990	0.000	0.400	0.490	0.014	0.018	0.028	0.034	0.081	0.040
Rolling Window	5	2	0.991	0.000	0.220	0.414	0.010	0.019	0.020	0.037	0.088	0.047
Rolling Window	5	3	0.991	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.081	0.045
Rolling Window	5	4	0.991	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.040	0.035
Rolling Window	6	4	0.991	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.045	0.043
Rolling Window	6	1	0.991	0.000	0.380	0.485	0.018	0.023	0.034	0.043	0.094	0.046
Rolling Window	6	2	0.991	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.070	0.041
Rolling Window	6	3	0.991	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.041	0.028
Rolling Window	7	3	0.991	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.040	0.044
Rolling Window	7	4	0.991	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.029	0.031
Rolling Window	7	1	0.991	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.084	0.048
Rolling Window	7	2	0.991	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.041	0.030
Rolling Window	8	2	0.991	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.031	0.028
Rolling Window	8	3	0.991	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.020	0.016
Rolling Window	8	4	0.991	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.012	0.007
Rolling Window	8	1	0.991	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.047	0.036
Rolling Window	9	1	0.991	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.033	0.023
Rolling Window	9	2	0.991	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.016	0.008
Rolling Window	9	3	0.991	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.011	0.006
Rolling Window	10	1	0.991	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.028	0.023
Rolling Window	10	2	0.991	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.013	0.009
Rolling Window	11	1	0.991	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.010	0.005
Rolling Window	12	1	0.988	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.016	0.006
Last Window	12	1	0.899	0.002	0.237	0.364	0.011	0.017	0.020	0.032	0.246	0.048
Last Window	12	2	0.905	0.003	0.792	0.322	0.049	0.028	0.091	0.051	0.258	0.055
Last Window	12	3	0.906	0.004	0.697	0.327	0.055	0.033	0.100	0.058	0.255	0.052
Last Window	12	4	0.909	0.005	0.849	0.226	0.076	0.046	0.137	0.078	0.295	0.059

Note. Acc = Accuracy, Pr = Precision, Rc = Recall, AUC_{PR} = Area under the precision recall curve. Next to each performance benchmark its standard deviation (σ) is given.

Table B.4
Results for undersampling the best performing pandemic rolling window aggregation

Agg Type	R	y	U_r	Acc	σ_{acc}	Pr	σ_{Pr}	Rc	σ_{Rc}	F1	σ_{F1}	AUC_{Pr}	σ_{AUC}
RW	3	4	0.02	0.991	0.000	0.300	0.458	0.014	0.021	0.026	0.040	0.061	0.044
RW	3	4	0.03	0.991	0.000	0.230	0.415	0.011	0.019	0.021	0.037	0.050	0.030
RW	3	4	0.04	0.991	0.000	0.305	0.456	0.015	0.021	0.028	0.040	0.053	0.026
RW	3	4	0.05	0.991	0.000	0.267	0.426	0.014	0.021	0.026	0.039	0.044	0.025
RW	3	4	0.1	0.990	0.001	0.208	0.304	0.018	0.022	0.033	0.040	0.043	0.027
RW	3	4	0.2	0.985	0.003	0.080	0.092	0.043	0.036	0.051	0.042	0.040	0.021
RW	3	4	0.4	0.942	0.014	0.036	0.018	0.190	0.091	0.060	0.029	0.031	0.022
RW	3	4	0.6	0.854	0.021	0.023	0.006	0.354	0.108	0.043	0.011	0.026	0.012
RW	3	4	0.8	0.740	0.028	0.018	0.004	0.493	0.126	0.034	0.008	0.025	0.013
RW	3	4	1	0.633	0.027	0.015	0.002	0.605	0.101	0.030	0.005	0.025	0.013

Note. RW = Rolling Window, U_r = Undersampling ratio, Acc = Accuracy, Pr = Precision, Rc = Recall, AUC_{PR} = Area under the precision recall curve. Next to each performance benchmark its standard deviation (σ) is given.

B.3 Full dataset full results

Table B.5
Full results for the full dataset

Aggregation Type	R	y	Acc	σ_{acc}	Pr	σ_{Pr}	Rc	σ_{Rc}	F1	σ_{F1}	AUC_{Pr}	σ_{AUC}
Last Window	12	1	0.890	0.004	0.948	0.011	0.684	0.009	0.795	0.008	0.882	0.004
Last Window	12	2	0.882	0.005	0.949	0.009	0.653	0.014	0.773	0.010	0.882	0.004
Last Window	12	3	0.879	0.004	0.933	0.010	0.646	0.011	0.763	0.009	0.852	0.005
Last Window	12	4	0.897	0.004	0.962	0.010	0.683	0.014	0.799	0.010	0.877	0.004

Note. Acc = Accuracy, Pr = Precision, Rc = Recall, AUC_{PR} = Area under the precision recall curve. Next to each performance benchmark its standard deviation (σ) is given.

B.4 Feature selection full results

Table B.6
Full model performance validation after RFECV

Aggregation Type	R	y	Acc	σ_{acc}	Pr	σ_{Pr}	Rc	σ_{Rc}	F1	σ_{F1}	AUC_{Pr}	σ_{AUC}
Last Window	12	1	0.882	0.004	0.936	0.012	0.665	0.012	0.778	0.009	0.895	0.004

Note. Acc = Accuracy, Pr = Precision, Rc = Recall, AUC_{PR} = Area under the precision recall curve. Next to each performance benchmark its standard deviation (σ) is given.

Table B.7
Resulting feature set from RFECV

Feature names		
Total hours	Assignment hours	Number of assignments
Sick leave short (3 months)	Sick leave short (6 months)	Sick leave short (12 months)
Sick leave long (6 months)	Sick leave long (12 months)	Conducted study (3 months)
Conducted study (6 months)	Paid leave (3 months)	Unpaid leave (12 months)
Overwork (3 months)	Overwork (6 months)	Overwork (12 months)
Salary Tier	Salary Position	Salary increase (12 months)
Age	Tenure	Promotion (12 months)
Travel time	Gender	Role
Job freedom & Responsibility	Culture diversity & leadership	Company satisfaction
Job searching	MS Married	MS Unmarried
MS Living togetger	LoS Auditing	LoS Acountancy
LoS Advisory	LoS Tax	LoS Central Staff
LoS Support	Loc Alphen a/d Rijn	Loc Amstelveen
Loc Apeldoorn	Loc Arnhem	Loc Breda
Loc Den Haag	Loc Eindhoven	Loc Groningen
Loc Rijswijk	Loc Rotterdam	Loc Tilburg
Loc Utrecht		

Note. The suffix (X months) indicates that this is a binary feature that shows if this occured (1) or not (0) in the last X periods.

Note. The prefixes MS, LoS and Loc indicate that this is a dumy feature for the features Marital Status, Line of Service and Location respectively.

B.5 Hyperparameter tuning full results

Abbreviations for the table headers in this section:

D_{Max}	Max Depth
F_{Max}	Max Features
LNs_{Max}	Max Leaf Nodes
SL_{Min}	Min Samples Leaf
SS_{Min}	Min Samples Split
E_n	Number of estimators
B	Bootstrapping
Alg	Algorithm
η	Learning Rate
Sub	Subsample
L	Loss
BO	Booster
CBM	Colsample Bylevel
CBT	Colsample Bytree
γ	gamma & Minimum loss reduction
CW_{Min}	Min Child Weight
λ	Lambda & Regularization term
SV	Solver
HLS	Hidden Layer Size
α	Regularization term
AF	Activation Function

Abbreviations for table entries in this section:

mse	Mean Squared Error
fm_mse	Friedman Mean Squared Error
dev	Deviance
exp	Exponential

Table B.8
Top 50 hyperparameters for hyperparameter tuning with a Decision Tree

Model	*Criterion*	D_{Max}	F_{Max}	LNs_{Max}	SL_{Min}	SS_{Min}	*Rank*
DT	entropy	20	47	None	5	5	1
DT	entropy	50	38	None	10	18	2
DT	gini	8	None	None	11	8	3
DT	entropy	6	30	50	6	4	4
DT	gini	120	33	50	7	8	5
DT	gini	15	49	None	25	5	6
DT	gini	15	41	50	8	10	7
DT	gini	7	25	None	7	15	8
DT	gini	6	49	50	3	18	9
DT	gini	9	13	None	5	7	10
DT	entropy	15	48	None	24	5	11
DT	entropy	11	41	30	4	5	12
DT	entropy	90	13	50	6	18	13
DT	entropy	50	37	20	3	8	14
DT	entropy	8	20	None	3	13	15
DT	gini	20	46	50	8	6	16
DT	entropy	70	34	None	26	15	17
DT	gini	20	36	50	2	6	18
DT	gini	10	13	None	2	16	19
DT	entropy	12	45	None	29	13	20
DT	entropy	8	38	30	26	3	21
DT	gini	120	39	30	36	20	22
DT	gini	30	46	20	26	14	23
DT	gini	9	45	20	5	3	24
DT	entropy	None	23	50	18	3	25
DT	entropy	15	23	50	32	11	26
DT	gini	9	32	50	35	20	27
DT	gini	None	32	20	31	11	28
DT	entropy	12	15	50	17	13	29
DT	entropy	40	24	30	25	7	30
DT	gini	7	47	20	37	15	31
DT	gini	None	35	30	36	13	32
DT	entropy	30	40	20	31	13	33
DT	gini	40	22	30	37	17	34
DT	gini	5	None	50	14	15	35
DT	entropy	7	34	20	36	14	36
DT	gini	10	37	None	47	11	37
DT	entropy	5	41	30	25	18	38
DT	entropy	40	38	20	43	4	39
DT	gini	70	44	30	42	10	40
DT	entropy	120	48	None	45	12	41
DT	entropy	50	12	None	23	4	42
DT	entropy	6	21	20	26	18	43
DT	entropy	11	42	20	50	5	44
DT	entropy	11	31	30	35	13	45
DT	gini	6	24	None	23	20	46
DT	entropy	5	32	30	33	8	47
DT	entropy	6	21	50	9	20	48
DT	entropy	150	40	20	56	14	49
DT	gini	120	43	30	50	4	50

Table B.9
Top 50 results for hyperparameter tuning with a Decision Tree

Rank	Acc	σ_{acc}	Pr	σ_{Pr}	Rc	σ_{Rc}	F1	σ_{F1}	AUC_{Pr}	σ_{AUC}
1	0.846	0.008	0.775	0.024	0.715	0.011	0.744	0.011	0.823	0.012
2	0.846	0.004	0.797	0.013	0.678	0.019	0.732	0.009	0.819	0.004
3	0.850	0.019	0.840	0.038	0.642	0.040	0.728	0.038	0.816	0.023
4	0.839	0.022	0.874	0.087	0.573	0.035	0.689	0.034	0.810	0.042
5	0.861	0.012	0.866	0.036	0.656	0.021	0.746	0.019	0.808	0.021
6	0.851	0.020	0.860	0.063	0.628	0.028	0.725	0.030	0.806	0.025
7	0.848	0.013	0.817	0.040	0.663	0.020	0.731	0.019	0.802	0.022
8	0.847	0.012	0.862	0.027	0.609	0.043	0.713	0.029	0.801	0.021
9	0.847	0.008	0.900	0.021	0.575	0.021	0.702	0.017	0.799	0.020
10	0.844	0.028	0.837	0.069	0.622	0.047	0.713	0.051	0.799	0.040
11	0.842	0.011	0.829	0.025	0.623	0.023	0.711	0.021	0.799	0.031
12	0.860	0.012	0.925	0.033	0.599	0.025	0.727	0.025	0.799	0.030
13	0.850	0.016	0.867	0.034	0.611	0.033	0.717	0.033	0.796	0.041
14	0.855	0.007	0.917	0.056	0.592	0.036	0.717	0.013	0.794	0.011
15	0.844	0.008	0.892	0.047	0.573	0.034	0.696	0.018	0.792	0.017
16	0.852	0.011	0.833	0.017	0.655	0.030	0.733	0.023	0.791	0.040
17	0.832	0.016	0.803	0.013	0.611	0.053	0.693	0.038	0.790	0.033
18	0.874	0.012	0.917	0.021	0.653	0.028	0.763	0.025	0.788	0.027
19	0.837	0.013	0.817	0.037	0.617	0.050	0.701	0.032	0.784	0.020
20	0.830	0.012	0.827	0.049	0.579	0.039	0.679	0.024	0.784	0.025
21	0.836	0.019	0.852	0.050	0.575	0.039	0.686	0.037	0.778	0.029
22	0.831	0.015	0.838	0.046	0.573	0.057	0.678	0.037	0.776	0.033
23	0.849	0.009	0.897	0.016	0.583	0.032	0.706	0.024	0.776	0.011
24	0.851	0.015	0.894	0.048	0.595	0.020	0.714	0.027	0.775	0.024
25	0.830	0.014	0.822	0.040	0.580	0.015	0.680	0.022	0.774	0.036
26	0.833	0.023	0.826	0.059	0.591	0.049	0.688	0.045	0.772	0.048
27	0.821	0.026	0.783	0.073	0.596	0.023	0.676	0.040	0.770	0.029
28	0.835	0.017	0.831	0.034	0.595	0.070	0.690	0.045	0.770	0.034
29	0.819	0.018	0.798	0.033	0.563	0.047	0.660	0.039	0.768	0.034
30	0.826	0.027	0.826	0.051	0.560	0.076	0.665	0.063	0.766	0.043
31	0.831	0.014	0.848	0.039	0.561	0.030	0.675	0.027	0.765	0.025
32	0.819	0.020	0.783	0.033	0.581	0.055	0.667	0.044	0.763	0.030
33	0.836	0.021	0.854	0.048	0.575	0.064	0.685	0.050	0.759	0.036
34	0.819	0.020	0.818	0.049	0.541	0.064	0.649	0.047	0.759	0.025
35	0.832	0.010	0.857	0.034	0.555	0.016	0.673	0.017	0.759	0.026
36	0.827	0.006	0.806	0.024	0.586	0.038	0.678	0.021	0.757	0.013
37	0.809	0.008	0.774	0.054	0.559	0.044	0.645	0.009	0.756	0.028
38	0.821	0.021	0.832	0.071	0.544	0.048	0.655	0.038	0.755	0.036
39	0.813	0.015	0.815	0.028	0.519	0.049	0.633	0.041	0.754	0.029
40	0.819	0.013	0.786	0.041	0.581	0.012	0.668	0.018	0.754	0.027
41	0.814	0.018	0.805	0.058	0.534	0.032	0.641	0.033	0.753	0.043
42	0.817	0.020	0.779	0.020	0.577	0.073	0.660	0.055	0.753	0.060
43	0.822	0.017	0.887	0.055	0.498	0.070	0.633	0.052	0.753	0.033
44	0.820	0.009	0.836	0.036	0.528	0.027	0.646	0.020	0.753	0.025
45	0.818	0.015	0.800	0.050	0.559	0.031	0.657	0.026	0.749	0.031
46	0.812	0.015	0.810	0.045	0.523	0.048	0.634	0.039	0.748	0.023
47	0.808	0.034	0.793	0.112	0.538	0.047	0.637	0.051	0.747	0.035
48	0.811	0.035	0.799	0.101	0.544	0.102	0.639	0.077	0.746	0.046
49	0.800	0.016	0.765	0.026	0.520	0.071	0.616	0.047	0.746	0.025
50	0.815	0.019	0.802	0.062	0.544	0.024	0.647	0.029	0.745	0.038

Table B.10
Top 50 hyperparameters for hyperparameter tuning with a Random Forest

Model	*Criterion*	D_{Max}	F_{Max}	LNs_{Max}	SL_{Min}	SS_{Min}	E_n	B	*Rank*
RF	entropy	40	32	None	2	10	100	TRUE	1
RF	entropy	9	20	50	4	19	159	TRUE	2
RF	gini	90	43	50	5	5	160	TRUE	3
RF	entropy	40	24	50	10	19	496	FALSE	4
RF	gini	50	29	30	3	19	364	TRUE	5
RF	gini	10	38	30	1	8	448	TRUE	6
RF	entropy	50	10	50	6	19	105	TRUE	7
RF	gini	30	31	None	10	12	305	TRUE	8
RF	entropy	40	6	None	12	12	471	FALSE	9
RF	entropy	150	43	None	13	2	437	TRUE	10
RF	gini	150	16	30	6	9	341	FALSE	11
RF	gini	10	23	30	4	11	314	TRUE	12
RF	entropy	20	49	50	13	7	486	TRUE	13
RF	gini	150	35	None	12	16	415	TRUE	14
RF	gini	70	49	50	17	3	163	TRUE	15
RF	entropy	15	36	50	22	11	412	TRUE	16
RF	gini	30	17	20	1	4	466	FALSE	17
RF	gini	None	31	50	28	18	223	FALSE	18
RF	entropy	50	21	None	31	15	382	FALSE	19
RF	gini	120	17	None	29	16	463	FALSE	20
RF	gini	120	38	30	18	8	128	TRUE	21
RF	gini	12	22	30	18	3	458	TRUE	22
RF	gini	15	8	30	18	8	475	FALSE	23
RF	entropy	20	22	20	17	11	120	FALSE	24
RF	gini	50	29	50	34	3	235	FALSE	25
RF	gini	20	15	30	17	10	158	TRUE	26
RF	gini	5	15	30	3	15	197	FALSE	27
RF	entropy	7	sqrt	50	11	5	418	TRUE	28
RF	entropy	70	25	None	36	16	256	FALSE	29
RF	entropy	10	38	30	27	19	490	TRUE	30
RF	gini	70	47	20	17	20	342	TRUE	31
RF	entropy	11	36	20	24	16	340	TRUE	32
RF	gini	70	43	30	9	3	310	FALSE	33
RF	entropy	8	28	30	30	9	134	TRUE	34
RF	gini	15	26	None	42	2	483	FALSE	35
RF	gini	10	22	50	42	6	426	FALSE	36
RF	entropy	7	6	30	23	14	329	FALSE	37
RF	gini	120	43	50	27	12	475	FALSE	38
RF	gini	70	19	30	42	17	173	FALSE	39
RF	gini	120	48	None	28	13	114	TRUE	40
RF	gini	12	18	None	28	7	387	TRUE	41
RF	gini	50	29	30	41	10	454	FALSE	42
RF	gini	30	3	None	18	8	195	FALSE	43
RF	entropy	11	auto	30	32	15	313	FALSE	44
RF	gini	50	8	20	29	11	256	FALSE	45
RF	gini	None	3	30	14	2	382	FALSE	46
RF	gini	5	19	50	32	17	338	FALSE	47
RF	entropy	6	36	30	40	5	487	FALSE	48
RF	gini	120	3	None	21	15	384	FALSE	49
RF	gini	40	48	None	13	12	179	FALSE	50

Table B.11
Top 50 results for hyperparameter tuning with a Random Forest

Rank	Acc	σ_{acc}	Pr	σ_{Pr}	Rc	σ_{Rc}	F1	σ_{F1}	AUC_{Pr}	σ_{AUC}
1	0.895	0.011	0.956	0.017	0.696	0.040	0.805	0.025	0.898	0.022
2	0.881	0.014	0.957	0.017	0.646	0.037	0.771	0.030	0.881	0.030
3	0.879	0.011	0.951	0.022	0.645	0.035	0.768	0.025	0.881	0.027
4	0.883	0.014	0.955	0.018	0.657	0.043	0.777	0.031	0.880	0.025
5	0.875	0.010	0.955	0.021	0.630	0.025	0.759	0.021	0.877	0.029
6	0.877	0.010	0.957	0.020	0.635	0.026	0.763	0.022	0.876	0.029
7	0.872	0.014	0.952	0.014	0.622	0.038	0.752	0.032	0.875	0.033
8	0.877	0.013	0.945	0.017	0.644	0.037	0.765	0.029	0.873	0.032
9	0.867	0.018	0.933	0.028	0.617	0.045	0.742	0.040	0.872	0.033
10	0.876	0.015	0.937	0.010	0.646	0.045	0.764	0.034	0.872	0.030
11	0.876	0.012	0.955	0.020	0.632	0.032	0.760	0.027	0.872	0.030
12	0.875	0.010	0.953	0.018	0.630	0.029	0.759	0.023	0.871	0.030
13	0.874	0.014	0.940	0.010	0.638	0.045	0.759	0.034	0.871	0.029
14	0.875	0.015	0.948	0.019	0.635	0.044	0.760	0.035	0.871	0.034
15	0.868	0.013	0.938	0.014	0.619	0.039	0.745	0.031	0.857	0.032
16	0.868	0.011	0.935	0.012	0.618	0.032	0.744	0.026	0.856	0.033
17	0.867	0.012	0.948	0.015	0.608	0.034	0.740	0.028	0.856	0.033
18	0.869	0.013	0.932	0.014	0.624	0.041	0.747	0.031	0.853	0.032
19	0.867	0.013	0.930	0.011	0.620	0.042	0.743	0.032	0.853	0.034
20	0.867	0.014	0.927	0.014	0.623	0.038	0.745	0.031	0.853	0.036
21	0.864	0.011	0.932	0.014	0.608	0.032	0.735	0.026	0.852	0.033
22	0.865	0.012	0.931	0.017	0.611	0.034	0.737	0.028	0.850	0.035
23	0.852	0.014	0.920	0.030	0.575	0.028	0.708	0.029	0.849	0.033
24	0.859	0.013	0.934	0.013	0.591	0.039	0.723	0.032	0.847	0.032
25	0.865	0.012	0.924	0.009	0.616	0.036	0.739	0.028	0.847	0.035
26	0.858	0.013	0.922	0.024	0.596	0.030	0.724	0.029	0.846	0.034
27	0.854	0.013	0.940	0.024	0.568	0.031	0.708	0.029	0.846	0.031
28	0.845	0.013	0.923	0.025	0.549	0.032	0.688	0.031	0.845	0.034
29	0.863	0.011	0.920	0.016	0.614	0.038	0.735	0.028	0.845	0.033
30	0.861	0.011	0.923	0.014	0.604	0.032	0.730	0.026	0.845	0.032
31	0.857	0.009	0.924	0.016	0.591	0.029	0.720	0.022	0.843	0.033
32	0.861	0.011	0.932	0.016	0.596	0.031	0.727	0.026	0.842	0.032
33	0.863	0.014	0.903	0.023	0.628	0.035	0.741	0.031	0.840	0.013
34	0.852	0.010	0.891	0.024	0.597	0.024	0.715	0.021	0.836	0.031
35	0.859	0.011	0.922	0.018	0.597	0.028	0.725	0.024	0.835	0.033
36	0.855	0.010	0.908	0.018	0.597	0.029	0.720	0.023	0.834	0.033
37	0.836	0.015	0.919	0.034	0.518	0.033	0.662	0.034	0.833	0.033
38	0.861	0.009	0.902	0.021	0.623	0.028	0.737	0.021	0.833	0.025
39	0.850	0.011	0.894	0.027	0.590	0.028	0.710	0.024	0.833	0.036
40	0.849	0.010	0.906	0.015	0.577	0.030	0.704	0.025	0.832	0.035
41	0.847	0.011	0.895	0.031	0.577	0.021	0.701	0.022	0.832	0.035
42	0.860	0.013	0.921	0.015	0.602	0.038	0.727	0.030	0.831	0.033
43	0.813	0.013	0.924	0.039	0.437	0.032	0.593	0.034	0.829	0.040
44	0.835	0.016	0.908	0.039	0.523	0.031	0.663	0.034	0.828	0.035
45	0.836	0.015	0.906	0.036	0.530	0.036	0.668	0.035	0.826	0.031
46	0.797	0.015	0.938	0.040	0.372	0.037	0.532	0.043	0.825	0.038
47	0.847	0.013	0.908	0.024	0.566	0.033	0.697	0.029	0.825	0.034
48	0.850	0.016	0.909	0.014	0.578	0.048	0.706	0.039	0.824	0.038
49	0.805	0.016	0.926	0.044	0.408	0.035	0.566	0.041	0.822	0.042
50	0.848	0.011	0.810	0.018	0.670	0.050	0.732	0.028	0.822	0.022

Table B.12
Top 50 hyperparameters for hyperparameter tuning with a Gradient Boosting Tree

Model	*Criterion*	D_{Max}	F_{Max}	SL_{Min}	SS_{Min}	E_n	*Sub*	η	*L*	*Rank*
GBT	mse	20	49	7	11	173	0.95	0.075	dev	1
GBT	fm_mse	120	40	15	14	285	0.9	0.025	dev	2
GBT	mse	7	25	18	8	298	0.85	0.025	dev	3
GBT	mse	10	44	15	15	400	0.85	0.15	dev	4
GBT	mse	150	9	4	2	426	1	0.05	exp	5
GBT	fm_mse	None	6	4	7	278	0.85	0.1	dev	6
GBT	fm_mse	9	sqrt	22	11	382	0.95	0.075	exp	7
GBT	fm_mse	8	24	18	15	358	1	0.3	dev	8
GBT	mse	30	11	17	4	473	1	0.01	dev	9
GBT	mse	8	12	1	3	103	1	0.3	exp	10
GBT	fm_mse	4	40	14	5	494	0.95	0.3	exp	11
GBT	fm_mse	10	auto	16	14	229	0.8	0.2	exp	12
GBT	mse	6	40	12	16	288	0.7	0.025	exp	13
GBT	fm_mse	40	42	2	4	146	0.85	0.05	dev	14
GBT	fm_mse	8	36	23	11	290	0.9	0.15	exp	15
GBT	mse	90	35	10	13	465	0.85	0.01	exp	16
GBT	mse	None	47	13	8	465	0.5	0.025	dev	17
GBT	fm_mse	9	8	3	4	194	0.9	0.2	exp	18
GBT	mse	30	10	14	6	248	1	0.2	dev	19
GBT	fm_mse	30	11	21	17	387	1	0.15	dev	20
GBT	mse	10	17	10	8	100	1	0.15	exp	21
GBT	mse	10	38	18	2	381	0.9	0.4	dev	22
GBT	mse	20	31	9	4	216	0.8	0.15	dev	23
GBT	fm_mse	150	41	15	3	349	1	0.15	exp	24
GBT	fm_mse	12	21	22	16	234	1	0.025	exp	25
GBT	fm_mse	120	46	28	2	204	0.9	0.1	exp	26
GBT	mse	150	43	1	2	445	0.6	0.025	dev	27
GBT	fm_mse	6	45	21	19	257	1	0.6	dev	28
GBT	mse	5	37	22	7	482	0.95	0.025	exp	29
GBT	fm_mse	4	auto	20	9	497	0.8	0.15	exp	30
GBT	fm_mse	3	36	26	7	242	1	0.4	dev	31
GBT	mse	40	log2	11	2	483	0.85	0.2	exp	32
GBT	fm_mse	20	12	32	19	413	1	0.025	exp	33
GBT	mse	3	38	9	2	249	0.6	0.15	exp	34
GBT	fm_mse	10	40	7	12	394	0.5	0.01	exp	35
GBT	fm_mse	40	28	35	10	363	0.9	0.025	exp	36
GBT	mse	8	20	16	16	350	1	0.9	exp	37
GBT	fm_mse	20	30	26	3	212	0.95	0.6	exp	38
GBT	mse	None	9	12	2	252	0.95	0.7	exp	39
GBT	mse	11	10	5	11	498	0.95	0.5	exp	40
GBT	mse	11	4	18	9	233	0.8	0.075	exp	41
GBT	fm_mse	15	11	15	10	245	0.95	0.7	dev	42
GBT	fm_mse	11	10	14	19	463	0.6	0.15	exp	43
GBT	fm_mse	150	29	53	10	445	0.9	0.025	exp	44
GBT	mse	6	17	32	20	287	0.8	0.1	dev	45
GBT	fm_mse	30	40	51	2	377	0.9	0.05	dev	46
GBT	mse	5	48	46	5	487	0.95	0.025	dev	47
GBT	fm_mse	40	27	35	19	129	0.9	0.075	exp	48
GBT	fm_mse	15	47	20	16	237	0.6	0.025	exp	49
GBT	mse	8	29	16	18	116	0.95	0.5	exp	50

Table B.13
Top 50 results for hyperparameter tuning with a Gradient Boosting Tree

Rank	Acc	σ_{acc}	Pr	σ_{Pr}	Rc	σ_{Rc}	F1	σ_{F1}	AUC_{Pr}	σ_{AUC}
1	0.893	0.017	0.911	0.028	0.727	0.050	0.808	0.033	0.904	0.021
2	0.894	0.017	0.917	0.022	0.726	0.047	0.810	0.034	0.904	0.023
3	0.894	0.019	0.924	0.021	0.719	0.054	0.808	0.039	0.904	0.023
4	0.894	0.011	0.893	0.015	0.750	0.037	0.815	0.022	0.903	0.018
5	0.891	0.012	0.910	0.015	0.723	0.048	0.805	0.027	0.903	0.023
6	0.894	0.015	0.921	0.016	0.723	0.053	0.809	0.031	0.903	0.021
7	0.897	0.014	0.909	0.018	0.745	0.040	0.819	0.028	0.903	0.021
8	0.900	0.011	0.896	0.016	0.769	0.040	0.827	0.023	0.903	0.021
9	0.893	0.016	0.934	0.016	0.708	0.053	0.804	0.035	0.902	0.026
10	0.894	0.010	0.908	0.031	0.737	0.038	0.812	0.020	0.902	0.020
11	0.893	0.016	0.872	0.014	0.769	0.048	0.817	0.031	0.902	0.016
12	0.890	0.013	0.886	0.016	0.744	0.041	0.808	0.026	0.901	0.019
13	0.891	0.021	0.929	0.029	0.705	0.053	0.801	0.042	0.901	0.023
14	0.891	0.010	0.924	0.030	0.712	0.031	0.803	0.020	0.901	0.019
15	0.891	0.011	0.887	0.011	0.748	0.040	0.811	0.024	0.901	0.020
16	0.892	0.016	0.926	0.026	0.711	0.048	0.803	0.033	0.900	0.025
17	0.893	0.009	0.910	0.009	0.729	0.031	0.809	0.019	0.900	0.024
18	0.894	0.011	0.908	0.031	0.738	0.039	0.813	0.023	0.900	0.024
19	0.896	0.014	0.905	0.023	0.746	0.050	0.817	0.029	0.900	0.021
20	0.894	0.007	0.901	0.018	0.742	0.032	0.813	0.016	0.900	0.019
21	0.896	0.012	0.915	0.023	0.735	0.046	0.814	0.025	0.900	0.019
22	0.895	0.010	0.896	0.007	0.750	0.033	0.816	0.020	0.900	0.020
23	0.891	0.014	0.900	0.020	0.733	0.038	0.808	0.028	0.900	0.024
24	0.891	0.015	0.891	0.021	0.743	0.044	0.810	0.030	0.899	0.022
25	0.890	0.019	0.926	0.026	0.705	0.051	0.799	0.039	0.899	0.024
26	0.888	0.016	0.890	0.019	0.733	0.047	0.803	0.031	0.899	0.024
27	0.889	0.007	0.945	0.018	0.685	0.026	0.794	0.016	0.899	0.019
28	0.895	0.014	0.880	0.010	0.768	0.042	0.820	0.027	0.899	0.015
29	0.894	0.015	0.921	0.024	0.724	0.040	0.810	0.030	0.899	0.019
30	0.891	0.017	0.879	0.023	0.752	0.039	0.811	0.030	0.899	0.021
31	0.896	0.015	0.882	0.020	0.770	0.040	0.822	0.029	0.898	0.019
32	0.893	0.012	0.906	0.012	0.735	0.048	0.810	0.026	0.898	0.021
33	0.889	0.015	0.909	0.018	0.717	0.049	0.801	0.032	0.898	0.027
34	0.894	0.018	0.889	0.023	0.752	0.042	0.815	0.034	0.898	0.024
35	0.887	0.018	0.939	0.020	0.683	0.053	0.790	0.040	0.898	0.027
36	0.889	0.014	0.902	0.016	0.723	0.045	0.801	0.030	0.898	0.027
37	0.896	0.010	0.890	0.018	0.760	0.040	0.819	0.021	0.897	0.017
38	0.895	0.014	0.896	0.025	0.751	0.041	0.817	0.027	0.897	0.019
39	0.896	0.010	0.907	0.017	0.742	0.038	0.815	0.021	0.897	0.017
40	0.897	0.009	0.919	0.011	0.737	0.038	0.817	0.020	0.896	0.019
41	0.893	0.013	0.903	0.013	0.735	0.041	0.809	0.026	0.896	0.024
42	0.892	0.011	0.904	0.010	0.731	0.039	0.808	0.024	0.896	0.020
43	0.888	0.014	0.888	0.014	0.732	0.042	0.802	0.029	0.896	0.020
44	0.890	0.019	0.900	0.019	0.726	0.051	0.803	0.037	0.896	0.028
45	0.890	0.015	0.883	0.018	0.744	0.043	0.807	0.030	0.896	0.022
46	0.892	0.017	0.894	0.027	0.743	0.041	0.811	0.031	0.896	0.026
47	0.887	0.018	0.909	0.018	0.708	0.049	0.795	0.037	0.896	0.026
48	0.890	0.016	0.908	0.020	0.721	0.043	0.803	0.032	0.896	0.027
49	0.887	0.018	0.922	0.012	0.695	0.054	0.792	0.038	0.896	0.024
50	0.891	0.013	0.878	0.017	0.756	0.042	0.812	0.025	0.895	0.021

Table B.14
Top 50 hyperparameters for hyperparameter tuning with Extreme Gradient Boosting

Model	*BO*	*CBL*	*CBT*	γ	η	D_{Max}	CW_{Min}	E_n	λ	*Sub*	*Rank*
XGB	gbtree	0.7	0.4	0	0.1	40	0.01	306	1	0.9	1
XGB	gbtree	0.4	0.6	0.5	0.075	12	0.01	378	0.8	0.7	2
XGB	gbtree	0.6	0.7	0.2	0.1	90	1	218	0.4	0.95	3
XGB	gbtree	0.5	0.4	0	0.15	3	0.25	431	0.5	1	4
XGB	gbtree	0.6	0.9	0.5	0.2	30	0.01	254	0.8	0.95	5
XGB	gbtree	0.6	0.7	0.01	0.025	150	0	248	0.7	0.9	6
XGB	gbtree	0.6	0.5	0.5	0.025	50	0.01	292	0	0.5	7
XGB	gbtree	1	0.9	5	0.15	20	0.25	327	0.9	0.7	8
XGB	gbtree	1	0.4	5	0.075	6	0	209	0.2	0.9	9
XGB	gbtree	0.5	0.4	0.2	0.075	40	3	248	0.2	1	10
XGB	gbtree	0.4	0.8	0.01	0.075	50	0	433	0.1	1	11
XGB	gbtree	1	1	3	0.15	120	0.5	158	0.9	0.85	12
XGB	gbtree	0.5	1	0.5	0.2	11	1	289	0.6	0.95	13
XGB	gbtree	0.5	0.5	0.01	0.15	15	0.05	429	0.9	0.85	14
XGB	gbtree	1	0.6	0.2	0.1	40	0.25	282	10	0.95	15
XGB	gbtree	0.9	0.5	0.25	0.05	15	0	217	4	0.5	16
XGB	gbtree	0.7	1	0.25	0.15	50	1	178	0.5	0.95	17
XGB	gbtree	0.4	0.7	3	0.1	90	0.01	270	1	0.9	18
XGB	gbtree	0.9	0.5	0	0.01	120	0.5	101	0	0.8	19
XGB	gbtree	0.5	0.8	2	0.1	15	0.25	418	0.5	0.6	20
XGB	gbtree	0.4	0.7	1	0.05	70	0.01	214	0.1	0.5	21
XGB	gbtree	0.7	0.5	2	0.025	11	0.25	364	8	0.85	22
XGB	gbtree	0.5	0.9	0.5	0.3	90	0.05	393	3	1	23
XGB	gbtree	0.7	0.7	0.25	0.001	11	0	438	0.6	0.95	24
XGB	gbtree	0.7	1	0.2	0.075	90	0	305	6	1	25
XGB	gbtree	0.9	0.6	0	0.005	30	0.05	418	0.6	0.5	26
XGB	gbtree	0.4	0.8	5	0.1	90	0.01	171	0.2	0.6	27
XGB	gbtree	0.9	1	0.25	0.1	40	0	412	0.9	0.6	28
XGB	gbtree	0.8	0.8	0.5	0.01	90	0	255	3	0.8	29
XGB	gbtree	0.8	0.6	0.25	0.01	70	3	412	0.6	0.8	30
XGB	gbtree	1	0.7	0	0.15	4	0.25	368	0.4	0.7	31
XGB	gbtree	0.8	0.7	0.1	0.2	3	1	349	6	0.9	32
XGB	gbtree	0.8	0.9	0.01	0.15	90	0.25	233	0.7	1	33
XGB	gbtree	0.8	0.5	2	0.4	4	0	350	10	0.85	34
XGB	gbtree	1	0.6	0.1	0.3	150	0	375	50	1	35
XGB	gbtree	0.4	0.4	0.25	0.025	90	3	470	0.9	0.6	36
XGB	gbtree	0.5	0.5	0.01	0.3	3	0	358	0.6	0.9	37
XGB	gbtree	0.5	1	0	0.01	30	0	228	0.5	0.9	38
XGB	gbtree	1	0.8	0.1	0.05	40	5	194	1	0.9	39
XGB	gbtree	0.7	0.7	0.5	0.075	20	5	244	1	1	40
XGB	gbtree	0.9	0.6	2	0.15	4	0	280	9	1	41
XGB	gbtree	1	0.4	0.1	0.4	50	0.05	187	0.7	0.7	42
XGB	gbtree	0.7	0.9	0.01	0.2	15	0.01	115	7	0.6	43
XGB	gbtree	0.8	0.8	0.5	0.075	3	1	280	0.9	0.9	44
XGB	gbtree	0.7	0.4	0.25	0.3	20	0	452	100	0.6	45
XGB	gbtree	0.9	0.5	0.01	0.005	10	0.5	418	3	0.8	46
XGB	gbtree	0.4	0.4	4	0.1	30	0.01	403	2	0.5	47
XGB	gbtree	0.5	0.5	0.5	0.15	40	0.25	212	50	0.95	48
XGB	gbtree	0.8	0.7	1	0.01	40	1	286	0.6	0.5	49
XGB	gbtree	0.4	0.7	0.1	0.4	6	0.05	267	2	1	50

Table B.15
Top 50 results for hyperparameter tuning with Extreme Gradient Boosting

Rank	Acc	σ_{acc}	Pr	σ_{Pr}	Rc	σ_{Rc}	F1	σ_{F1}	AUC_{Pr}	σ_{AUC}
1	0.900	0.012	0.933	0.014	0.733	0.034	0.821	0.025	0.910	0.024
2	0.896	0.015	0.913	0.021	0.736	0.048	0.814	0.030	0.908	0.024
3	0.896	0.013	0.916	0.011	0.735	0.043	0.815	0.027	0.908	0.021
4	0.901	0.017	0.909	0.022	0.758	0.041	0.827	0.032	0.907	0.018
5	0.898	0.013	0.915	0.022	0.742	0.044	0.818	0.027	0.907	0.019
6	0.894	0.012	0.936	0.023	0.711	0.039	0.807	0.026	0.907	0.022
7	0.899	0.012	0.940	0.024	0.721	0.029	0.816	0.022	0.907	0.021
8	0.897	0.016	0.926	0.022	0.729	0.040	0.815	0.032	0.907	0.017
9	0.896	0.017	0.933	0.024	0.719	0.048	0.811	0.034	0.907	0.020
10	0.896	0.017	0.908	0.032	0.740	0.035	0.815	0.030	0.906	0.021
11	0.896	0.011	0.924	0.024	0.727	0.040	0.813	0.023	0.906	0.021
12	0.896	0.015	0.912	0.029	0.738	0.036	0.815	0.028	0.906	0.022
13	0.897	0.015	0.908	0.021	0.744	0.040	0.817	0.028	0.905	0.025
14	0.899	0.010	0.916	0.012	0.745	0.028	0.822	0.020	0.905	0.021
15	0.898	0.014	0.916	0.019	0.740	0.043	0.818	0.028	0.905	0.022
16	0.891	0.009	0.927	0.017	0.706	0.027	0.801	0.019	0.905	0.023
17	0.893	0.013	0.906	0.019	0.734	0.039	0.811	0.026	0.904	0.020
18	0.894	0.015	0.910	0.019	0.735	0.046	0.812	0.030	0.904	0.021
19	0.897	0.013	0.958	0.014	0.701	0.042	0.809	0.029	0.904	0.020
20	0.896	0.013	0.909	0.019	0.740	0.038	0.816	0.026	0.904	0.025
21	0.897	0.012	0.922	0.014	0.732	0.034	0.816	0.024	0.903	0.023
22	0.893	0.016	0.931	0.019	0.711	0.042	0.805	0.033	0.903	0.024
23	0.894	0.016	0.905	0.019	0.738	0.047	0.812	0.032	0.903	0.023
24	0.893	0.013	0.954	0.022	0.691	0.037	0.801	0.028	0.903	0.020
25	0.895	0.017	0.921	0.021	0.726	0.049	0.811	0.034	0.902	0.024
26	0.895	0.011	0.952	0.018	0.700	0.032	0.806	0.022	0.902	0.024
27	0.892	0.016	0.915	0.020	0.721	0.046	0.806	0.033	0.902	0.021
28	0.896	0.016	0.907	0.014	0.742	0.053	0.815	0.032	0.902	0.023
29	0.891	0.011	0.948	0.021	0.689	0.034	0.797	0.023	0.902	0.023
30	0.894	0.017	0.941	0.024	0.706	0.044	0.806	0.034	0.902	0.025
31	0.896	0.013	0.901	0.013	0.750	0.042	0.818	0.026	0.902	0.018
32	0.896	0.020	0.897	0.026	0.751	0.050	0.817	0.038	0.901	0.023
33	0.896	0.010	0.914	0.018	0.736	0.037	0.815	0.021	0.901	0.018
34	0.897	0.021	0.895	0.024	0.757	0.052	0.820	0.039	0.901	0.024
35	0.893	0.017	0.895	0.020	0.743	0.050	0.811	0.034	0.901	0.024
36	0.894	0.015	0.916	0.019	0.729	0.040	0.811	0.030	0.901	0.026
37	0.892	0.016	0.878	0.022	0.760	0.039	0.814	0.030	0.901	0.018
38	0.894	0.009	0.940	0.019	0.705	0.036	0.805	0.021	0.900	0.018
39	0.890	0.015	0.906	0.019	0.720	0.040	0.802	0.029	0.900	0.026
40	0.890	0.017	0.896	0.019	0.733	0.053	0.806	0.035	0.900	0.024
41	0.894	0.020	0.910	0.028	0.731	0.050	0.810	0.039	0.900	0.023
42	0.894	0.010	0.909	0.022	0.736	0.034	0.813	0.020	0.900	0.018
43	0.896	0.014	0.917	0.012	0.731	0.045	0.813	0.029	0.899	0.024
44	0.891	0.017	0.910	0.022	0.720	0.044	0.804	0.033	0.899	0.022
45	0.896	0.017	0.901	0.020	0.748	0.043	0.817	0.033	0.899	0.025
46	0.891	0.013	0.959	0.011	0.678	0.037	0.794	0.029	0.899	0.025
47	0.894	0.020	0.911	0.017	0.730	0.054	0.810	0.040	0.899	0.025
48	0.893	0.018	0.908	0.020	0.729	0.050	0.808	0.037	0.899	0.025
49	0.892	0.014	0.943	0.017	0.695	0.040	0.800	0.031	0.899	0.028
50	0.889	0.015	0.890	0.012	0.736	0.051	0.805	0.030	0.899	0.018

Table B.16
Top 50 hyperparameters for hyperparameter tuning with AdaBoost

Model	*Alg*	η	E_n	*Rank*
ADA	SAMME.R	0.9	498	1
ADA	SAMME.R	0.9	479	2
ADA	SAMME.R	0.8	468	3
ADA	SAMME.R	0.9	421	4
ADA	SAMME.R	1	427	5
ADA	SAMME.R	0.9	416	6
ADA	SAMME.R	0.9	414	7
ADA	SAMME.R	0.7	490	8
ADA	SAMME.R	0.9	371	9
ADA	SAMME.R	0.9	354	10
ADA	SAMME.R	1	421	11
ADA	SAMME.R	0.9	367	12
ADA	SAMME.R	0.8	418	13
ADA	SAMME.R	0.7	478	14
ADA	SAMME.R	0.8	398	15
ADA	SAMME.R	0.8	425	16
ADA	SAMME.R	0.8	392	17
ADA	SAMME.R	0.8	389	18
ADA	SAMME.R	1	399	19
ADA	SAMME.R	0.7	418	20
ADA	SAMME.R	0.8	381	21
ADA	SAMME.R	0.8	376	22
ADA	SAMME.R	0.7	407	23
ADA	SAMME.R	0.7	411	24
ADA	SAMME.R	0.6	475	25
ADA	SAMME.R	0.7	403	26
ADA	SAMME.R	1	355	27
ADA	SAMME.R	0.7	383	28
ADA	SAMME.R	0.9	317	29
ADA	SAMME.R	0.7	366	30
ADA	SAMME.R	1	329	31
ADA	SAMME.R	0.7	352	32
ADA	SAMME.R	0.6	395	33
ADA	SAMME.R	0.8	297	34
ADA	SAMME.R	0.8	300	35
ADA	SAMME.R	0.7	349	36
ADA	SAMME.R	0.8	308	37
ADA	SAMME.R	1	308	38
ADA	SAMME.R	1	317	39
ADA	SAMME.R	0.5	453	40
ADA	SAMME.R	1	294	41
ADA	SAMME.R	0.5	441	42
ADA	SAMME.R	1	249	43
ADA	SAMME.R	0.4	494	44
ADA	SAMME.R	0.4	494	44
ADA	SAMME.R	0.7	301	46
ADA	SAMME.R	0.6	312	47
ADA	SAMME.R	1	260	48
ADA	SAMME.R	0.8	246	49
ADA	SAMME.R	0.7	284	50

Table B.17
Top 50 results for hyperparameter tuning with AdaBoost

Rank	Acc	σ_{acc}	Pr	σ_{Pr}	Rc	σ_{Rc}	F1	σ_{F1}	AUC_{Pr}	σ_{AUC}
1	0.884	0.017	0.868	0.028	0.739	0.035	0.798	0.031	0.881	0.025
2	0.884	0.018	0.873	0.031	0.734	0.034	0.798	0.031	0.880	0.026
3	0.883	0.017	0.883	0.032	0.720	0.030	0.793	0.031	0.878	0.026
4	0.883	0.014	0.871	0.022	0.733	0.029	0.796	0.025	0.878	0.026
5	0.885	0.015	0.879	0.025	0.732	0.032	0.799	0.027	0.878	0.024
6	0.883	0.014	0.872	0.020	0.733	0.031	0.796	0.025	0.877	0.025
7	0.883	0.014	0.872	0.020	0.732	0.032	0.796	0.026	0.877	0.025
8	0.881	0.016	0.881	0.028	0.717	0.031	0.790	0.030	0.877	0.025
9	0.881	0.014	0.869	0.018	0.730	0.034	0.793	0.027	0.877	0.025
10	0.882	0.015	0.874	0.021	0.725	0.035	0.792	0.029	0.877	0.025
11	0.884	0.015	0.877	0.024	0.732	0.030	0.798	0.026	0.877	0.023
12	0.882	0.014	0.870	0.016	0.730	0.034	0.793	0.027	0.876	0.025
13	0.883	0.017	0.883	0.032	0.721	0.031	0.794	0.031	0.876	0.027
14	0.880	0.016	0.879	0.028	0.712	0.031	0.786	0.030	0.876	0.024
15	0.881	0.015	0.882	0.027	0.715	0.025	0.790	0.026	0.876	0.026
16	0.883	0.017	0.883	0.032	0.719	0.028	0.792	0.029	0.876	0.026
17	0.881	0.014	0.883	0.026	0.713	0.027	0.789	0.026	0.876	0.026
18	0.883	0.014	0.890	0.027	0.714	0.025	0.792	0.025	0.876	0.026
19	0.881	0.013	0.874	0.024	0.724	0.027	0.792	0.024	0.876	0.023
20	0.879	0.016	0.878	0.027	0.709	0.033	0.785	0.030	0.876	0.025
21	0.880	0.014	0.880	0.027	0.712	0.025	0.787	0.025	0.876	0.026
22	0.881	0.013	0.885	0.022	0.712	0.025	0.789	0.023	0.875	0.027
23	0.879	0.016	0.878	0.022	0.709	0.037	0.785	0.031	0.875	0.026
24	0.879	0.017	0.877	0.026	0.712	0.036	0.786	0.032	0.875	0.026
25	0.879	0.018	0.884	0.035	0.703	0.032	0.783	0.033	0.874	0.027
26	0.877	0.015	0.872	0.021	0.709	0.032	0.782	0.028	0.874	0.026
27	0.881	0.016	0.873	0.030	0.724	0.029	0.791	0.028	0.874	0.024
28	0.878	0.017	0.878	0.029	0.706	0.033	0.782	0.032	0.873	0.026
29	0.877	0.015	0.867	0.022	0.714	0.031	0.783	0.027	0.873	0.026
30	0.877	0.018	0.878	0.034	0.703	0.030	0.781	0.032	0.873	0.026
31	0.878	0.016	0.867	0.029	0.719	0.028	0.786	0.028	0.873	0.025
32	0.878	0.018	0.879	0.034	0.706	0.032	0.783	0.033	0.872	0.027
33	0.876	0.017	0.880	0.029	0.696	0.034	0.777	0.032	0.872	0.026
34	0.878	0.016	0.875	0.031	0.712	0.027	0.785	0.028	0.871	0.029
35	0.879	0.015	0.876	0.028	0.712	0.027	0.785	0.027	0.871	0.028
36	0.879	0.019	0.882	0.033	0.706	0.035	0.784	0.035	0.871	0.028
37	0.878	0.017	0.874	0.030	0.709	0.031	0.783	0.031	0.871	0.028
38	0.877	0.014	0.866	0.027	0.715	0.027	0.783	0.025	0.871	0.023
39	0.879	0.015	0.869	0.031	0.720	0.024	0.788	0.026	0.871	0.024
40	0.874	0.015	0.877	0.024	0.691	0.033	0.773	0.029	0.870	0.026
41	0.875	0.017	0.865	0.030	0.711	0.033	0.780	0.031	0.869	0.023
42	0.874	0.015	0.877	0.023	0.694	0.031	0.775	0.028	0.869	0.027
43	0.875	0.015	0.866	0.025	0.707	0.030	0.779	0.028	0.868	0.025
44	0.871	0.018	0.872	0.027	0.688	0.038	0.769	0.034	0.868	0.029
44	0.871	0.018	0.872	0.027	0.688	0.038	0.769	0.034	0.868	0.029
46	0.873	0.017	0.864	0.025	0.702	0.036	0.775	0.031	0.868	0.028
47	0.873	0.017	0.875	0.029	0.691	0.034	0.773	0.033	0.867	0.028
48	0.874	0.016	0.866	0.026	0.706	0.035	0.778	0.030	0.867	0.026
49	0.875	0.018	0.871	0.038	0.702	0.027	0.778	0.031	0.867	0.031
50	0.873	0.019	0.866	0.030	0.700	0.041	0.774	0.036	0.866	0.028

Table B.18
Top 50 hyperparameters for hyperparameter tuning with a Multilayer Perceptron

Model	*SV*	η	*HLS*	α	*AF*	*Rank*
MLP	adam	adaptive	(100, 50, 50)	0.01	logistic	1
MLP	adam	constant	(50, 100)	0	logistic	2
MLP	adam	constant	(100,)	0.001	logistic	3
MLP	adam	constant	(50, 400)	0.00001	logistic	4
MLP	adam	constant	(300, 100)	0.001	logistic	5
MLP	adam	constant	(200, 100)	0	logistic	6
MLP	adam	adaptive	(200, 300)	0.0001	logistic	7
MLP	adam	constant	(200, 200)	0.1	logistic	8
MLP	adam	constant	(300, 300, 100)	0.01	logistic	9
MLP	adam	constant	(200, 300, 100)	0.001	logistic	10
MLP	adam	constant	(50, 100)	0.0001	logistic	11
MLP	adam	adaptive	(200, 100)	0.1	logistic	12
MLP	adam	adaptive	(300, 300)	0.001	logistic	13
MLP	adam	adaptive	(100, 100)	0.1	logistic	14
MLP	adam	invscaling	(400, 100, 50)	0	logistic	15
MLP	adam	adaptive	(200, 200)	0.001	logistic	16
MLP	adam	adaptive	(400, 400)	0.1	logistic	17
MLP	adam	invscaling	(50, 100)	0.25	logistic	18
MLP	adam	invscaling	(400, 200, 100)	0	logistic	19
MLP	adam	constant	(300, 100)	0.0001	logistic	20
MLP	adam	constant	(300, 300, 100)	0.0001	logistic	21
MLP	adam	invscaling	(200, 100)	0.25	logistic	22
MLP	adam	constant	(100, 50, 50)	0	logistic	23
MLP	adam	constant	(300, 300)	0.00001	logistic	24
MLP	adam	invscaling	(300, 100)	0.5	tanh	25
MLP	adam	adaptive	(100, 50, 50)	0.2	tanh	26
MLP	adam	adaptive	(300, 50)	0.1	logistic	27
MLP	sgd	adaptive	(50, 50)	0.001	relu	28
MLP	adam	invscaling	(200,)	0.01	relu	29
MLP	adam	adaptive	(300, 400)	0.01	logistic	30
MLP	adam	constant	(200, 100)	0.1	tanh	31
MLP	adam	adaptive	(200, 50)	0.00001	tanh	32
MLP	adam	invscaling	(100,)	0.2	logistic	33
MLP	adam	constant	(200,)	0.0001	relu	34
MLP	adam	adaptive	(300, 100)	0	tanh	35
MLP	adam	constant	(200, 200)	0.2	logistic	36
MLP	adam	constant	(200, 300, 100)	0.0001	tanh	37
MLP	adam	constant	(400,)	0.0001	relu	38
MLP	adam	constant	(200,)	0	tanh	39
MLP	adam	invscaling	(200, 100, 50)	0.2	tanh	40
MLP	adam	adaptive	(200, 50)	0.5	tanh	41
MLP	adam	invscaling	(300, 100)	0.25	tanh	42
MLP	adam	invscaling	(200, 100, 50)	0.9	tanh	43
MLP	adam	constant	(300, 300)	0.2	tanh	44
MLP	adam	invscaling	(50,)	0.25	relu	45
MLP	adam	adaptive	(200, 100, 50)	0.5	tanh	46
MLP	adam	adaptive	(200, 300, 100)	0.01	tanh	47
MLP	adam	constant	(200, 200, 200)	0.1	tanh	48
MLP	adam	invscaling	(200, 100, 50)	0.001	tanh	49
MLP	adam	invscaling	(100, 300, 100)	0.1	tanh	50

Table B.19
Top 50 results for hyperparameter tuning with a Multilayer Perceptron

Rank	Acc	σ_{acc}	Pr	σ_{Pr}	Rc	σ_{Rc}	F1	σ_{F1}	AUC_{Pr}	σ_{AUC}
1	0.747	0.022	0.611	0.047	0.547	0.075	0.572	0.042	0.625	0.029
2	0.749	0.011	0.652	0.057	0.452	0.111	0.521	0.071	0.615	0.028
3	0.740	0.017	0.671	0.074	0.362	0.071	0.461	0.050	0.615	0.038
4	0.742	0.017	0.616	0.053	0.496	0.064	0.544	0.019	0.615	0.029
5	0.718	0.058	0.608	0.098	0.480	0.180	0.505	0.065	0.609	0.044
6	0.750	0.023	0.630	0.018	0.486	0.158	0.532	0.120	0.609	0.050
7	0.740	0.017	0.660	0.097	0.434	0.165	0.488	0.121	0.608	0.036
8	0.738	0.010	0.668	0.051	0.339	0.109	0.436	0.080	0.605	0.032
9	0.735	0.010	0.624	0.061	0.456	0.166	0.501	0.091	0.605	0.026
10	0.734	0.022	0.650	0.085	0.364	0.119	0.450	0.085	0.603	0.045
11	0.735	0.024	0.609	0.056	0.463	0.123	0.512	0.083	0.603	0.033
12	0.734	0.011	0.696	0.066	0.283	0.084	0.390	0.090	0.602	0.016
13	0.737	0.021	0.671	0.084	0.364	0.149	0.445	0.106	0.601	0.042
14	0.730	0.012	0.608	0.077	0.488	0.160	0.512	0.110	0.598	0.025
15	0.725	0.023	0.587	0.076	0.512	0.113	0.532	0.041	0.597	0.046
16	0.729	0.014	0.706	0.062	0.237	0.084	0.345	0.087	0.597	0.033
17	0.724	0.021	0.664	0.110	0.376	0.208	0.430	0.111	0.596	0.027
18	0.737	0.014	0.663	0.053	0.344	0.118	0.436	0.107	0.595	0.036
19	0.730	0.024	0.625	0.098	0.410	0.088	0.483	0.039	0.594	0.056
20	0.735	0.012	0.640	0.079	0.414	0.114	0.485	0.054	0.594	0.033
21	0.727	0.024	0.583	0.062	0.500	0.088	0.530	0.048	0.591	0.048
22	0.722	0.011	0.647	0.097	0.346	0.175	0.414	0.096	0.591	0.026
23	0.730	0.024	0.607	0.084	0.500	0.157	0.521	0.100	0.589	0.046
24	0.735	0.018	0.636	0.075	0.420	0.112	0.489	0.064	0.587	0.033
25	0.733	0.016	0.670	0.040	0.303	0.152	0.394	0.121	0.587	0.031
26	0.728	0.019	0.602	0.078	0.470	0.111	0.511	0.066	0.583	0.026
27	0.721	0.018	0.611	0.073	0.341	0.192	0.392	0.189	0.583	0.040
28	0.688	0.002	0.250	0.387	0.002	0.003	0.005	0.006	0.582	0.148
29	0.725	0.018	0.686	0.097	0.281	0.150	0.361	0.167	0.580	0.026
30	0.731	0.023	0.635	0.074	0.349	0.117	0.436	0.105	0.580	0.049
31	0.733	0.014	0.623	0.055	0.408	0.114	0.478	0.075	0.575	0.025
32	0.731	0.019	0.629	0.089	0.428	0.157	0.479	0.114	0.574	0.035
33	0.733	0.018	0.712	0.061	0.267	0.113	0.367	0.139	0.574	0.042
34	0.720	0.021	0.693	0.128	0.253	0.165	0.327	0.154	0.572	0.061
35	0.728	0.021	0.597	0.050	0.407	0.115	0.473	0.084	0.568	0.034
36	0.725	0.010	0.623	0.056	0.344	0.106	0.428	0.083	0.567	0.025
37	0.727	0.013	0.644	0.058	0.332	0.145	0.412	0.109	0.565	0.027
38	0.692	0.043	0.583	0.143	0.457	0.222	0.444	0.165	0.564	0.057
39	0.718	0.018	0.575	0.047	0.433	0.112	0.481	0.068	0.564	0.030
40	0.717	0.025	0.622	0.088	0.318	0.196	0.379	0.146	0.560	0.076
41	0.731	0.030	0.586	0.058	0.422	0.238	0.445	0.223	0.560	0.108
42	0.730	0.027	0.578	0.072	0.400	0.205	0.446	0.181	0.555	0.090
43	0.728	0.024	0.507	0.258	0.355	0.237	0.393	0.224	0.553	0.096
44	0.719	0.021	0.571	0.126	0.308	0.228	0.359	0.190	0.551	0.085
45	0.689	0.034	0.643	0.159	0.314	0.282	0.319	0.174	0.551	0.024
46	0.712	0.045	0.555	0.140	0.400	0.274	0.405	0.214	0.549	0.091
47	0.711	0.009	0.587	0.047	0.332	0.230	0.372	0.172	0.548	0.018
48	0.706	0.022	0.567	0.068	0.466	0.194	0.473	0.111	0.545	0.030
49	0.721	0.011	0.600	0.030	0.342	0.124	0.419	0.106	0.544	0.026
50	0.714	0.015	0.631	0.078	0.200	0.035	0.302	0.046	0.542	0.068

B.6 Threshold tuning full results

Table B.20
Full results for threshold tuning

Model	Acc	σ_{acc}	Pr	σ_{Pr}	Rc	σ_{Rc}	F1	σ_{F1}	T	σ_T
XGB	0.902	0.013	0.888	0.042	0.795	0.035	0.837	0.022	0.283	0.138

Note. Acc = Accuracy, Pr = Precision, Rc = Recall, T = Threshold. Next to each performance benchmark its standard deviation (σ) is given.

Table B.21
Full results for threshold tuning that balances the cost of misclassifications

Model	Acc	σ_{acc}	Pr	σ_{Pr}	Rc	σ_{Rc}	F7.92	$\sigma_{F7.92}$	T	σ_T
XGB	0.421	0.080	0.354	0.034	0.999	0.002	0.971	0.003	0.001	0.001

Note. Acc = Accuracy, Pr = Precision, Rc = Recall, T = Threshold. Next to each performance benchmark its standard deviation (σ) is given.