UNIVERSITY OF LJUBLJANA
SCHOOL OF ECONOMICS AND BUSINESS

MASTER'S THESIS

OPPORTUNITIES AND CHALLENGES OF MACHINE LEARNING IMPLEMENTATION TO PREDICT DEBT COLLECTION PERFORMANCE

Ljubljana, September 2020 JERNEJ KNECHTL

AUTHORSHIP STATEMENT

The undersigned Jernej Knechtl, a student at the School of Economics and Business, University of Ljubljana, (hereafter: SEB LU), author of this written final work of studies with the title Opportunities and challenges of machine learning implementation to predict debt collection performance, prepared under the supervision of Jurij Jaklič, PhD

DECLARE

1. this written final work of studies to be based on the results of my own research;

2. the printed form of this written final work of studies to be identical to its electronic form;

3. the text of this written final work of studies to be language-edited and technically in adherence with the SEB LU's Technical Guidelines for Written Works, which means that I cited and/or quoted works and opinions of other authors in this written final work of studies in accordance with the SEB LU's Technical Guidelines for Written Works;

4. to be aware of the fact that plagiarism (in written or graphical form) is a criminal offence and can be prosecuted in accordance with the Criminal Code of the Republic of Slovenia;

5. to be aware of the consequences a proven plagiarism charge based on the this written final work could have for my status at the SEB LU in accordance with the relevant SEB LU Rules;

6. to have Retrieved all the necessary permits to use the data and works of other authors which are (in written or graphical form) referred to in this written final work of studies and to have clearly marked them;

7. to have acted in accordance with ethical principles during the preparation of this written final work of studies and to have, where necessary, Retrieved permission of the Ethics Committee;

8. my consent to use the electronic form of this written final work of studies for the detection of content similarity with other written works, using similarity detection software that is connected with the SEB LU Study Information System;

9. to transfer to the University of Ljubljana free of charge, non-exclusively, geographically and time-wise unlimited the right of saving this written final work of studies in the electronic form, the right of its reproduction, as well as the right of making this written final work of studies available to the public on the World Wide Web via the Repository of the University of Ljubljana;

10. my consent to the publication of my personal data that are included in this written final work of studies and in this declaration, when this written final work of studies is published.

Ljubljana, September 1st, 2020 Author's signature:_____

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

In this day and age, machine learning can be found in every step of our lives. It is present in our everyday life, for example, in online shopping where the search results, recommendations as well as customer support (e.g., chatbots) are the result of machine learning algorithms in the background (Sentance, 2019). It can make our lives easier with practical applications such as traffic jam predictions or instant translations that do not only translate the words but also preserve the voice (Hao, 2019; McFadden, 2019). Banks use machine learning to provide greater security with fraud detection and to determine the creditworthiness of a person (Mejia, 2019; Walker, 2019). These are just a few machine learning examples from our everyday lives. Although the beginning of the development and use of machine learning in applications dates back to the middle of the twentieth century, we have not experienced its full potential yet.

The use of machine learning in companies is steadily growing. Deloitte predicted that the number of pilot projects using machine learning would double in 2018 compared to 2017 and double again by 2020. Following the growth of the implementation of machine learning algorithms, the growth of investments into machine learning will grow from 12$ billion in 2017 to 57.6$ billion by 2021 (Lee, Stewart, & Calugar-Pop, 2018). Increasing interest in this technology can also be explained by the fact that machine learning systems are exceptional learners and can outperform human abilities in a wide range of activities. A good indicator for that is the image recognition error rate of images from the ImageNet database through time. In the year 2010, the error rate of image recognition for a machine learning algorithm was at 30%. Six years later, the error rate was under 4%, and it continues to decline (Brynjolfsson & McAfee, 2017, pp. 4–7).

Because machine learning algorithms are very versatile and have a broad scope of utilisation, we can observe the use of machine learning in almost every industry. In this thesis, we focus on the use of machine learning in the field of finance, specifically in the area of debt collection.

Regardless of the industry that a company is in, some customers will inevitably have difficulties settling their duties on time. According to Eurostat, in the year 2018, Slovenia's private debt represented 72.8% of the gross domestic product (GDP), that is 33,290 million euro of household debt (Eurostat, n.d.). Unpaid debts do not pose a problem only for the debtor, as interest accrues over time, but can also have a significant impact on the companies that own the debt since it cuts into their revenue. For these companies, it is crucial to recover as much debt as possible, as this will affect their profit (Qingchen, Geer, & Bhulai, 2018).

The debt collection process can consist of two phases – a prelegal and a legal phase. If the prelegal phase is unsuccessful, a legal process can be started. Since the legal process is expensive, can stretch over an extended period, and occupy many resources, it is avoided as

much as possible and viewed only as a last resort (Qingchen et al., 2018). Therefore, companies need to recover as much debt as possible in the prelegal phase. To maximise the collected amount in the prelegal phase, companies frequently offer the debtors interest-free extensions, payment plans and in some cases, even partial debt write-offs (Qingchen et al., 2018). Many companies have turned to the use of machine learning to gain more information from data to optimise the collection process.

A reliable prediction of the probability of debt repayment would allow the debt collection companies to focus primarily on the debts that have a high likelihood of debt repayment. Furthermore, the companies could save the cost of the debt collection process on the debts that have a low probability by performing only necessary activities or writing the debts off.

The purpose of this thesis is to explore the key opportunities and challenges a company faces with the implementation of machine learning algorithms to predict the performance of debt collection. A successful debt outcome prediction will enable a debt collection company to identify the debts that are more likely to be successfully recovered. Therefore, it will be possible to allocate more resources to the debts that are classified as successful. The activities carried out on the debts that will be predicted as unsuccessful will be limited to only necessary activities. These debts are also suitable for a potential debt write-off. It is believed that a system of selective resource allocation could increase the general performance of debt collection.

An obstacle in the research originated from a lack of information about the debtor. Unlike a financial company that offers loans, a debt collection company has little information about the debtor at their disposal. Data that is handed over includes information about the debt, occasionally lacking even the debtor contact data, which debt collection companies have to obtain themselves. To reduce the lack of features, we also include the data about activities that have been carried out as part of the debt collection process. Consequently, we predict the outcome of the debt collection in two time points. (1) At the beginning of the debt collection process with data that is handed over by the creditor, and (2) one month after the debt collection process has started with the combination of the data handed over and the additional data, which is the product of actions performed in the process of debt collection so far.

The research goals of the thesis are:

- to review relevant literature on the use of machine learning in the area of debt collection,
- to determine whether the use of machine learning algorithms is reasonable and meaningful in the context of the business case,
- to derive the key opportunities and challenges through the implementation of different machine learning algorithms within the business case, and

- to describe how to deal with the implementation of machine learning algorithms, to identify the most successful one for the prediction of debt collection performance, and to identify the key success factors of the implementation.

The data set used in the empirical part of this research was provided by a debt collection company from Slovenia. Due to their desire for anonymity, the company's name cannot be disclosed. All sensitive information is excluded from the research.

This thesis is divided into seven chapters. In Chapters 1 and 2, we present the relevant background in the areas of machine learning and debt collection, respectively. The focus is on the methods used in the construction of the analytical model for the business case. Here, we use the description method and focus on books, professional articles and publicly available publications as the primary source of content. To gain more knowledge in the field of debt collection, and the key steps in the process, interviews with the employees of the company that provided the data were conducted. The purpose of the first two chapters is to equip the reader with the necessary background knowledge to be able to follow and understand the practical part of the thesis, as well as the key findings derived in this research. Chapter 3 provides an overview of the existing research on the use of machine learning algorithms in the area of debt collection.

The experimental part of the thesis is divided into two parts. First, Chapter 4 describes the business case, which outlines the problem, its background and the strategy used in the empirical part. Chapter 5 then presents the construction of the analytical models in the context of the business case. It captures everything from data understanding to the final construction of debt collection classification models. It is designed as step-by-step documentation in the process of model construction. To find the best possible solution, we apply different machine learning algorithms from more simple ones like logistic regression to more complex models, such as support vector machines and neural networks. For model creation, we use the CRISP-DM methodology, which according to Wirth and Hipp (2000), is the most widely-used open standard in this field, and is useful for planning, documentation, and communication.

The results of the evaluation of the different models constructed and their comparison are presented in Chapter 6. In this section, the method of comparative analysis is used to measure the performance of the various models. We complete the thesis with a discussion about the key findings of the research in Chapter 7.

1 MACHINE LEARNING

In this chapter, we first define machine learning and then classify machine learning algorithms into different categories based on various criteria (i.e., human supervision, the ability to learn on the fly and the approach to generalise). In Section 1.3, we present the most common challenges that can occur when experimenting with machine learning models,

together with some suggestions on how to tackle these problems. The challenges include the insufficient quantity of training data, nonrepresentative training data set, poor data quality, typical feature engineering steps, and problems of overfitting and underfitting. Section 1.4 first briefly describes all algorithms that are later used for the model construction, and then presents the ones that yield the most successful models (i.e., logistic regression, neural networks, support vector machines, and random forests) in more detail. The last section presents the metrics used for the evaluation of the algorithms in the empirical part of the thesis.

1.1 What is machine learning

A generally accepted definition of machine learning is believed to be set by the pioneer of machine learning Arthur Samuel (1959) in his paper "Some Studies in Machine Learning Using the Game of Checkers", who also coined the term *machine learning*. The definition says that "machine learning is the field of study that gives computers the ability to learn without being explicitly programmed" (Géron, 2017, p. 4). However, the definition cannot be found in the paper mentioned; it just may be an interpretation of Samuel's work by other authors. Another commonly recognised definition of machine learning was set by Thomas M. Mitchell (1997, p. 2), which says that "a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E".

1.2 Types of machine learning algorithms

Machine learning algorithms can be categorised according to the specific properties they possess. The most common criteria are human supervision, the ability to learn on the fly, and the approach to generalise. An algorithm can belong to multiple categories, as these criteria are not exclusive (Géron, 2017, p. 7).

1.2.1 Criterion: human supervision

Machine learning algorithms are distinguished by the type and the amount of supervision they receive in the training phase. An algorithm belongs to one of the four categories: supervised learning, unsupervised learning, semisupervised learning, or reinforcement learning (Géron, 2017, p. 8).

1.2.1.1 Supervised learning

In supervised learning, the training data set includes labels, which are the desired solutions. The data set is the collection of N labelled examples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$, where $\boldsymbol{x}_i$ stands for a feature vector. Each example is represented by D values called features. A feature is noted

as $x_i^{(j)}$, where $j = 1, \ldots, D$. Each feature is an independent variable that describes the example. For all examples in the data set, the *j*-th position in the feature vector contains the same kind of information. The label y_i can be an element belonging to a finite set of classes {1, 2, …, *C*}, a real number, or an even more complex structure like a vector, a matrix, a tree, or a graph (Burkov, 2019, pp. 1–2).

The most common supervised learning algorithms are:

- linear regression,
- logistic regression,
- *k*-nearest neighbours,
- decision trees and random forests,
- support vector machines, and
- neural networks.

The most typical example to demonstrate supervised learning is the classification of spam emails. The classification algorithm is trained with emails that are already labelled as spam or not spam. In this case, the label belongs to the finite set of classes {*spam, not_spam*}. Based on that information, the algorithm learns to distinguish between them (Géron, 2017, pp. 8–9).

1.2.1.2 Unsupervised learning

In the case of unsupervised learning, the training data set is a collection of unlabeled examples $\{\boldsymbol{x}_i\}_{i=1}^{N}$. The goal of an unsupervised learning algorithm is to create a model that takes a feature vector $\boldsymbol{x}$ as an input and transforms it into a value or another vector that can be used to solve a practical problem. The algorithm tries to make sense of the data set without any guidance. As an example, in dimensionality reduction, the output of a model is a feature vector with fewer features than the input $\boldsymbol{x}$. In anomaly detection, the output is a real number that indicates how the feature vector $\boldsymbol{x}$ is different from a typical instance in the data set. In clustering, the model returns the id of the cluster for each feature vector in the data set (Burkov, 2019, p. 2).

The most popular unsupervised algorithms are:

- clustering (k-means, hierarchical cluster analysis, expectation maximisation),
- visualisation and dimensionality reduction (principal component analysis, kernel principal component analysis, locally-linear embedding), and
- association rule learning (Apriori, Eclat) (Géron, 2017, pp. 10–12).

1.2.1.3 Semisupervised learning

Algorithms in this category can handle partially labelled data, which in practice means that a small part of the data is labelled, while the majority stays unlabelled. Most semisupervised learning algorithms consist of a mix of supervised learning and unsupervised learning. For example, deep belief networks (DBNs) are based on restricted Boltzmann machines (RBMs). RBMs are sequentially trained with unsupervised learning, but in later stages, the whole algorithm is calibrated with the use of supervised learning techniques (Géron, 2017, p. 13).

1.2.1.4 Reinforcement learning

Reinforcement learning is entirely different. In the context of reinforcement learning, the algorithm is called an agent. The agent is set in an environment where it can select and perform actions. Based on those actions, it receives positive or negative rewards in return. The agent has to learn by itself the best strategy (called a policy) to maximise the reward over time. The policy determines actions that the agent should choose in any given situation (Géron, 2017, p. 13).

Reinforcement learning is used to tackle unique kinds of challenges where decision making is sequential, and the goal is longterm. It is used in robotics, game playing, logistics, or resource management (Burkov, 2019, p. 3). One of the most known examples of reinforcement learning is DeepMind's AlphaGo program. The program made headlines in 2017 when it played the complex Chinese board game Go against the world champion Ke Jie, and it won. The winning policy was learned by analysing millions of games and then playing games against itself. During the match against Ke Jie, the training of the program was turned off, and the program just applied the policy that it learned beforehand (Géron, 2017, pp. 13–14).

1.2.2 Criterion: the ability to learn on the fly

This criterion classifies machine learning algorithms according to the ability to learn from a stream of incoming data incrementally. An algorithm can be either a batch learning algorithm or an online learning algorithm.

1.2.2.1 Batch learning

Batch learning algorithms are unable to learn incrementally and must be trained with all the data available, which is typically done offline. This is time-consuming and computationally intensive. The algorithm is trained before it is implemented into production. After the implementation, the algorithm applies what it has learned during the training and does not learn anymore. To further update the algorithm, it is not enough to just train the current algorithm on new data, but it has to be trained on the new, as well as the old data. After the

training, it has to be implemented into production again. Due to time and computation complexity, this kind of learning is not feasible for all machine learning algorithms. If the amount of data is too immense or the computational resources are limited, then this solution may prove itself to be very costly. In this case, the algorithm has to be trained using online learning (Géron, 2017, pp. 14–15).

1.2.2.2 Online learning

The online learning algorithm is incrementally trained by receiving data instances sequentially, either individually or by mini-batches (groups of small amounts of data). In this case, every learning step is fast, the computational aspect is nonintensive, and the algorithm can learn about new data without any interruptions, as it arrives. This kind of learning is very well suited for algorithms where the computational resources are scarce and for algorithms that need to adapt to changes swiftly or autonomously. An advantage of these algorithms is that when they are finished learning from a new batch of data, this data can be discarded, which can save up a lot of space. Online learning algorithms are also suitable for enormous amounts of data that generally do not fit on one computer's main memory. This kind of learning is called out-of-core learning and is usually done offline. The algorithm first loads a part of the data and runs a training step on the loaded data. Afterwards, it reruns these steps until all the data is processed (Géron, 2017, pp. 15–17).

An important parameter of online learning algorithms, called the learning rate, tells us how fast they can adapt to changing data. A high learning rate results in a fast adaptive algorithm that also tends to forget old data quickly. A slow learning rate results in an algorithm that adapts to changes in data slower and is less responsive to noise in the new data or sequences of nonrepresentative data. The performance of an online learning algorithm can decline if the data, which the algorithm is trained on, is bad. The bad data can result from a faulty sensor, or someone is intentionally feeding the algorithm bad data. An excellent example of the latter is if someone is spamming a search engine to rank higher in search results. To reduce such a risk, the algorithm has to be monitored closely. Furthermore, in case of bad data, the ability to learn has to be promptly switched off, and if possible, the algorithm has to be reverted to the previous state. In the case of performance issues, an anomaly detection algorithm can be used on the input data (Géron, 2017, pp. 15–17).

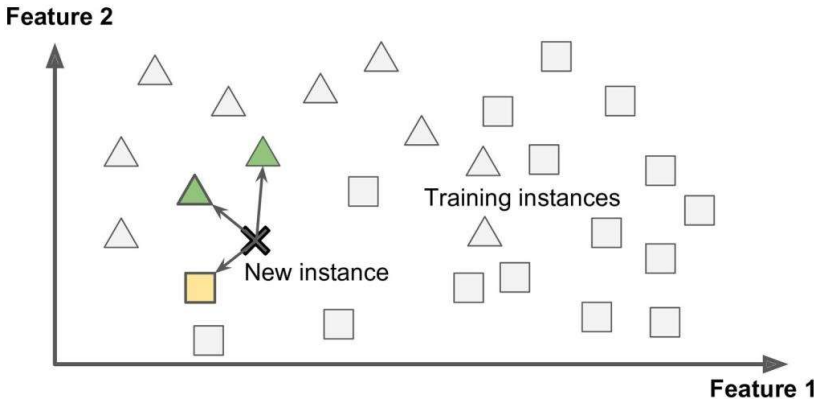
1.2.3 Criterion: the approach to generalise

Another way to divide machine learning algorithms is by how they generalise. The majority of machine learning tasks are predictive: when an algorithm receives an unseen example, it has to generalise based on the data that it was trained on. Having a good performance on training data is favourable, but it is not enough. More essential is that the algorithm performs well on the new instances. Algorithms can be divided into two categories regarding their generalisation: instance-based learning and model-based learning (Géron, 2017, p. 17).

1.2.3.1 Instance-based learning

This approach is the equivalent of learning by heart. Géron (2017, p. 17) displays this method of generalisation on a spam example. A machine learning algorithm would flag emails as spam only if they were identical to the ones that have already been flagged by other users. The method of comparison may also be extended with the use of a similarity measure (e.g., the number of words that the emails have in common). This way, not only the identical emails would get flagged but also emails that are similar enough to already flagged emails. In this case, the algorithm learns the examples by heart and then generalises to new cases using a similarity measure. In Figure 1, the new instance is compared to the three closest instances, which define how the new instance is labelled.

Figure 1: Instance-based learning

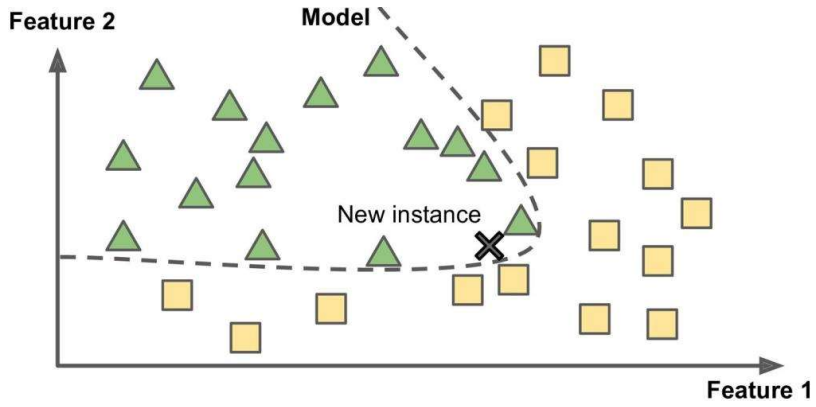


Source: Géron (2017, p. 17).

1.2.3.2 Model-based learning

On the other hand, it is possible to build a model on the training data, which is then used to make predictions.

Figure 2: Model-based learning



Source: Géron (2017, p. 18).

In Figure 2, a model calculated the boundary that is separating two data sets. To predict a label for a new instance, it only has to look at which side of the border the new instance lies on.

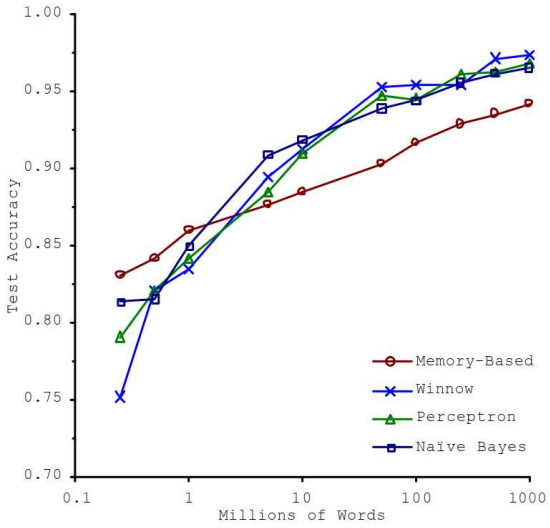
1.3 Main challenges of machine learning

Machine learning can improve efficiency in many areas as well as can save time and reduce the cost for businesses. Nevertheless, it comes with its own set of challenges. The field of machine learning is complex, and therefore the understanding of the process and different algorithms available is crucial. There are two primary sources of problems when implementing a machine learning algorithm. As Géron (2017, p. 22) wrote: "the two things that can go wrong are bad algorithm and bad data". Regarding the former, the developers are facing challenges like overfitting or underfitting to the training data. Problems regarding data include small quantity and poor quality of it, which furthermore leads to nonrepresentative training data and irrelevant feature selection.

1.3.1 Insufficient quantity of training data

Machine learning algorithms need a lot of data to work properly. To train an algorithm for solving a simple problem, it may take thousands of examples. To solve a more complex problem like image or speech recognition, it can take up to millions of examples to train an algorithm (Géron, 2017, p. 22).

Figure 3: The effect of data quantity on different algorithms



Source: Banko & Brill (2001).

In a study performed by Banko and Brill (2001), the effect of an extensive data set on machine learning for natural language disambiguation is demonstrated. The authors tested what happens when they train the existing methods with more data. This effect was tested

on different machine learning algorithms: naïve Bayes, perceptron, winnow, and a simple memory-based learner. The learning curves for the algorithms are shown in Figure 3.

They argue that "results suggest that we may want to reconsider the trade-off between spending time and money on algorithm development versus spending it on corpus development" (Banko & Brill, 2001).

1.3.2 Nonrepresentative training data

For the machine learning system to perform well, the data needs to be representative of the new instances. It is unlikely for a model that has been trained on a nonrepresentative training set to make accurate predictions. If the training set is too small, it can cause sampling noise, which means that the representative of the sample is a result of chance. On the other hand, if the sampling method is flawed, even a larger training set can be nonrepresentative (Géron, 2017, pp. 24–25).

One of the most known sampling bias happened during the 1936 US elections. In the election, Roosevelt was representing the democrats and Landon the republicans. The magazine Literary Digest conducted a large poll among telephone and magazine subscribers, which assured them that Landon would be elected as the new president, but instead Roosevelt won. The sample on which the magazine conducted the poll was not representative, which led to the incorrect assumption. People who could afford phones and magazine subscriptions in 1936 were not a valid cross-section of the voters. It turned out that the sample contained mostly republican voters (Huff & Geis, 1993, pp. 20–21).

1.3.3 Poor data quality

The quality of the data profoundly affects the performance quality of the machine learning algorithms. If the data contains a lot of outliers, errors, and noise, it gets harder for the algorithm to recognise patterns in the data. As a consequence, the predictions become less accurate. Data cleaning can be demanding and can take up quite some time, but the effort is often well worth the time. Typical examples of data cleaning are discarding the outliers, manually correcting the errors, and in case of missing data, deciding to ignore the whole attribute altogether or to fill in the missing values manually (e.g., with the median) (Géron, 2017, p. 25).

1.3.4 Feature engineering

Transformation of the raw data into a data set is called feature engineering. Due to the different ranges of features, the first step is usually to rescale the data. The rescaling ensures that all inputs are approximately in the same small range, with which we avoid problems, such as numerical overflow. Most common techniques for feature scaling are normalisation

and standardisation. Normalisation is a procedure of converting the actual range of a numerical feature's values to a standard range of values, generally on the scale between 0 and 1, or -1 and 1. On the other hand, standardisation is the process of rescaling numerical features to have properties of a standard normal distribution (i.e., the mean is zero, and the standard deviation from the mean is one) (Burkov, 2019, pp. 45–46).

The phrase "garbage in, garbage out" is often used to describe the importance of feature engineering. With the presence of irrelevant features, the performance of the machine learning system decreases. Therefore, it is crucial to come up with a good set of relevant features to train the system on (Géron, 2017, pp. 25–26). This can be a labour-intensive process that requires the analyst to possess some domain knowledge to come up with highly informative features with high predictive power (Burkov, 2019, pp. 43–44).

When building a predictive model, it is especially beneficial to reduce the number of input variables. This can be achieved by using a set of features that are obtained from the original input (i.e., dimensionality reduction) or selecting a subset of the most informative variables (i.e., feature selection). Applying feature selection is helpful for the following reasons. (1) The risk of overfitting is decreased, which leads to improved prediction performance. (2) The training time, as well as storage requirements, are reduced. (3) A smaller number of variables leads to a better understanding of the data and easier data visualisation (Guyon & Elisseeff, 2003).

The majority of algorithms (besides the ones that can perform automatic feature selection) can be put into two categories: wrapper methods and filter methods. Wrapper methods evaluate various models with different subsets of the input variables and choose a subset, which leads to the best model performance (John, Kohavi, & Pfleger, 1994). Their most significant disadvantage is that many models have to be evaluated, which leads to greater time complexity. Filter methods, on the other hand, are much more computationally efficient. Based on the evaluation of the input variables before training the model, filter methods select a subset of variables that will be put into the model. The downside of this approach is that if the number of variables chosen is too high, it can lead to collinearity problems (Kuhn & Johnson, 2013, pp. 490–499).

Another division of feature selection methods depends on how many features a method selects at once. Univariate methods consider each feature separately and rank them. In contrast, multivariate methods take into account dependencies among features and consider subsets of features together (Wang, Lei, Zeng, Tong, & Yan, 2013, p. 2). There are two aspects to feature selection, depending on the relevance of the feature and its redundancy. For observing the redundancy, multivariate analysis is used; while the relevance is inspected with the univariate methods (Jović, Brkić, & Bogunović, 2015, p. 1). In this thesis, three (statistical) univariate filter methods are used to select relevant subsets of features: mutual information, chi-square, and ANOVA F-test.

Mutual information (MI) measures the degree of relatedness between two variables by detecting any kind of relationship between them, while it is not sensitive to their sizes (Ross, 2014, p. 1). MI can be used to calculate the relatedness between any variable and a discrete (categorical) target variable. The output is a non-negative value, which equals zero if and only if the two variables are independent. In the case of a positive value, higher values mean higher dependency (Scikit-learn, 2020, p. 1984).

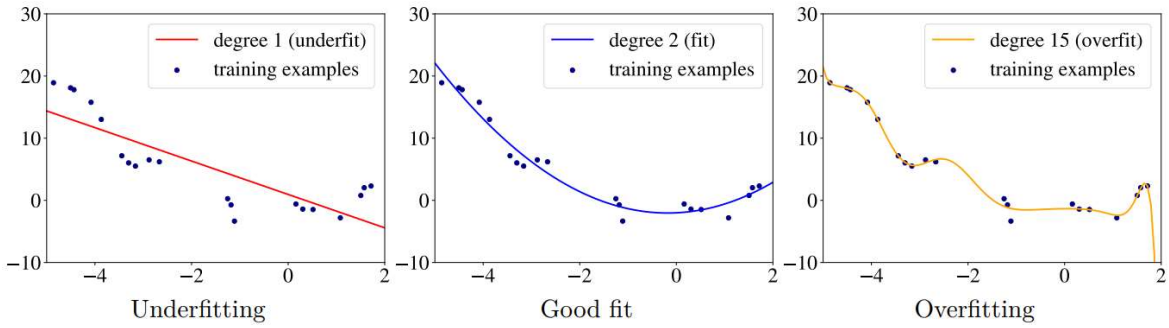
Chi-square test of independence can discover any significant relationship between two categorical variables. It is a hypothesis-testing test that uses the null hypothesis H_0 and the alternative hypothesis H_1, defined as follows. (H_0) there is no relationship between the two variables, and (H_1) there is some relationship between the two variables. After choosing the p-value threshold (usually 0.05 or 0.01), the null hypothesis is rejected if the selected p-value is significant (Wijaya, 2016).

Analysis of variance (ANOVA) determines the relationship between the numerical and categorical variables (e.g., between the numerical input variables and the categorical output in classification). A class of statistical tests for calculating the ratio between values of variance is called F-test (or F-statistic). Variance included could be from two distinct samples or the unexplained and explained variance by a statistical test, such as ANOVA. The ANOVA technique is a kind of F-statistic, called the ANOVA F-test. This test is often used for feature selection, where only a subset of features that are dependent on the target variable is used for the modelling (Brownlee, 2020).

1.3.5 Overfitting and underfitting

A common problem occurs when a model fits the training data too good or too loose. Figure 4 illustrates these problems, called underfitting and overfitting, on the same training examples.

Figure 4: Examples of underfitting, a good fit, and overfitting



Source: Burkov (2019, p. 51).

If a model predicts the labels of the training examples well, it has low bias. On the other hand, if a model performs poorly on the training data, it has high bias, or in other words, it

underfits. Underfitting is the inefficiency of a model to predict the labels of the training examples. There are two common reasons for underfitting: (a) the model is too simple for the data (this often happens when using a linear model, see Figure 4 – underfitting), or (b) selected features do not have sufficient predictive power. To tackle the first problem, one can use a more complex model, such as a quadratic or polynomial model, which can fit the data better than a straight line. For solving the second problem, it is crucial to use features with a higher predictive power, which can be done by gathering more data or with feature extraction (Burkov, 2019, p. 51).

On the contrary, overfitting occurs when the model predicts labels very well on the training set, but performs poorly on new data (see Figure 4 – overfitting). In statistics, overfitting is named a problem of high variance. Variance is an error of a model concerning its sensitivity to small fluctuations in the training data. This means that, if the data was sampled again, it would result in a completely different model. Frequent reasons for overfitting are (a) the model is too complex (this often happens when using deep neural networks), or (b) the number of features compared to the number of training examples is too big. The problem of overfitting can be tackled with different approaches: using a simpler model, reducing the dimensionality of the data (e.g., use of dimensionality reduction algorithms), adding more training data, or regularising the model (Burkov, 2019, pp. 51–52).

Regarding the second reason, if the data is high dimensional, but the number of training examples is low, even a linear regression algorithm can build a model that is trying to find complex relationships between features, and therefore overfit. Such a model would inherit all imperfections of the training data, for example, noise and sampling flaws due to the small size of the training set. Furthermore, its ability to perform well on other data than the one it was trained on is very poor (Burkov, 2019, pp. 51–52).

1.4 Machine learning algorithms

We are dealing with a binary classification task of predicting whether the debt will be successfully collected or not. There are many algorithms for solving classification problems, from simple ones like logistic regression to more complex ones, for example, neural networks and support vector machines. In the empirical part of the research, we apply different machine learning algorithms with default hyperparameters in order to determine the best-performing ones for further optimisation. In this section, we first briefly describe all of the algorithms used in the research. Then, the best-performing algorithms selected for optimisation are described in more detail.

Algorithms applied are logistic regression, stochastic gradient descent (SGD) linear classification, naïve Bayes, *k*-nearest neighbours (kNN), decision tree, random forest, support vector machines (SVM) with three different kernels (linear, polynomial, and radial basis function), and neural network. In total, eight algorithms are used to build ten different models (three models with different kernels are based on the support vector machines).

Logistic regression is a classification algorithm, which is used to estimate the probability that an instance belongs to a specific class. The logistic regression is similar to the linear regression in computing a weighted sum of the input features with a bias term. Instead of outputting a direct result of the computation like linear regression, the logistic regression outputs the logistic result of it, which is a probability between 0 and 1. If the computed probability is greater than 0.5, the logistic regression predicts that the instance belongs to the positive class labelled "1", or else it predicts that the instance belongs to the negative class labelled "0" (Géron, 2017, pp. 137–138).

The SGD is not an algorithm on its own. It is an optimisation technique used to fit linear classification algorithms under convex loss function, such as linear support vector machines or logistic regression. Applying SGD to an algorithms means that the gradient of the loss is estimated for one sample at a time, updating the model along the way with a decreasing strength schedule. Advantages of using SGD are higher efficiency and the ease of implementation. The algorithm's disadvantages are a high number of required hyperparameters and the sensitivity to feature scaling (Scikit-learn, 2020, pp. 275, 2091).

The naïve Bayes algorithm is called naïve because it assumes that all features are independent of each other given the class. Surprisingly, in practice, this algorithm often works very well, even when the assumption about conditional independence does not hold (Russell & Norvig, 2016, p. 499). The assumption makes the algorithm robust, which sometimes leads to outperforming other, more sophisticated machine learning algorithms. Furthermore, the naïve Bayes algorithm exposes the relationships between feature values and classes and provides an essential insight into the training data (Možina, Demšar, Kattan, & Zupan, 2004).

The k-nearest neighbours algorithm (kNN) is a non-parametric instance-based learning algorithm. After the model is built, the algorithm keeps the whole training set in memory. For classifying a new, previously unseen example, kNN looks at the example's immediate neighbourhood, which consists of k training examples that are closest to the new example. The algorithm then predicts the label that appears most often in this neighbourhood or the average label in case of classification and regression, respectively (Burkov, 2019, pp. 19, 34). kNN is a simple algorithm that is usually successful in classification problems with irregular decision boundaries. In spite of the algorithm's simplicity, kNN is successfully used for problems, such as handwritten digits and satellite image scenes (Scikit-learn, 2020, p. 284).

The decision tree algorithm can be referred to as an acyclic graph, which is used to make decisions. In each splitting node of the graph, a specific feature is inspected. If the value of that feature is below an explicit threshold, the left branch is followed; otherwise, the right branch is followed. For every split, the quality of the split is measured by minimising a criterion such as entropy or gini. The decision tree graph stops when a leaf node is reached. The leaf node determines the predicted class of an instance. The algorithm does not

guarantee an optimal solution, since the decision to split on each iteration does not depend on future splits. The performance of a decision tree algorithm can be improved using the techniques of back-propagation to search for the optimal decision tree and pruning to cut off branches that do not contribute enough to the error reduction (Burkov, 2019, pp. 27–30). The advantage of the decision trees is that they are straightforward to understand and to explain since they mirror human decision making. Decision trees can also be graphically displayed and are easily interpreted even by non-experts. Moreover, they require little data preparation compared to other algorithms, can handle both numerical and categorical data, and are suitable for multioutput problems. Decision trees also have some disadvantages. Generally, their predictive accuracy can lack behind other alternative classification and regression algorithms. Decision-tree learners can create overcomplex trees that do not generalise well (i.e., overfitting). Additionally, they can be non-robust, meaning that a small change in the data can cause a substantial change in the final estimated tree (Hastie, James, Tibshirani, & Witten, 2017, pp. 315–316; Scikit-learn, 2020, pp. 319–320).

An approach to increase the performance of simple learning algorithms, such as decision trees, is ensemble learning. Instead of trying to train one highly accurate model, ensemble learning focuses on training a large number of low-accuracy models and combining their predictions to obtain a high accuracy meta-model (Burkov, 2019, p. 83). Such an ensemble of decision tree models is called a random forest. Despite the simplicity of decision trees, the random forest is one of the most powerful machine learning algorithms available today. In general, the random forest algorithm trains a group of decision tree models. Each model is trained on a different subset of the training set. The final prediction is made by obtaining the predictions of each model. The class with the most votes then gets predicted (Géron, 2017, p. 183).

The support vector machine is a great method to try if no prior knowledge about a domain is present. It creates an $(n-1)$-dimensional hyperplane to separate examples that are represented with n features. Support vector machines (SVMs) have three appealing properties. (1) They generalise well because they construct a decision boundary that has the most significant possible distance to the examples. This boundary is called a maximum margin separator. (2) SVMs can embed the data into a higher-dimensional space by using a kernel trick. Usually, the data that cannot be linearly separated in the original space is without difficulty separable in a higher-dimensional space. (3) Lastly, SVMs integrate the benefits of both parametric and non-parametric methods. Therefore, they can represent complex functions and are at the same time resistant to the overfitting (Burkov, 2019, p. 4; Russell & Norvig, 2016, p. 744).

Artificial neural networks are mathematical models based on the human brain. They consist of neurons (i.e., nodes or units) that are connected by directed links. The function of a link from neuron i to neuron j is to transfer the activation value a_i from i to j. Each link has a corresponding weight $w_{i,j}$, which represents the strength of the connection. Each neuron j first calculates a weighted sum of its inputs and then applies an activation function g on the

computed sum. The activation function is usually nonlinear to ensure that a neural network can represent nonlinear functions. There are many different kinds of neural networks, for example, perceptron, feed-forward, and recurrent (Russell & Norvig, 2016, pp. 727–729). Neural networks perform very well on problems with a lot of features and diverse data. Because they can detect complex, nonlinear relationships between the input features and target variables, neural networks are vastly used even when the relationships between variables are not understood. The most significant advantage of neural networks is their successful performance when solving very complex problems. However, they have some disadvantages. First, they are often used as black boxes, which means that one cannot explain how they derived at their decisions. Second, they are very computationally expensive. Lastly, they are prone to overfitting, which can be solved by using preventive measures, such as cross-validation (Ciobanu & Vasilescu, 2013).

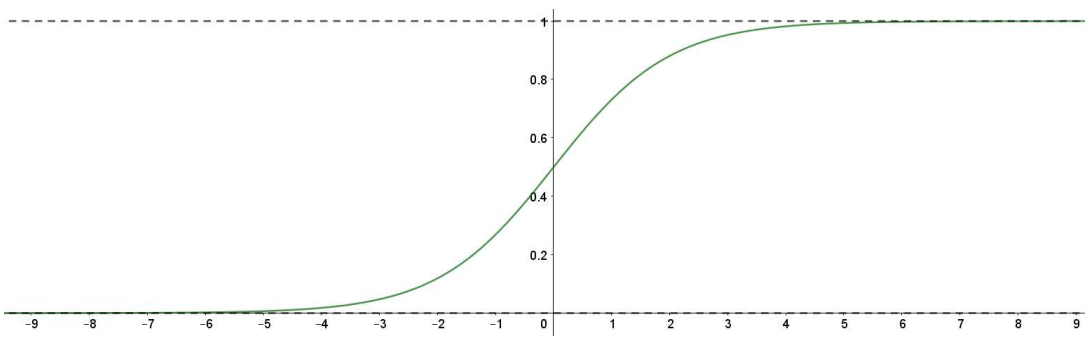
In the following sections, we describe the best-performing algorithms in more detail.

1.4.1 Logistic regression

Contrary to what the name suggests, logistic regression is one of the fundamental classification learning algorithms. Its mathematical formulation is similar to that of the linear regression. The goal of the logistic regression is to model the target variable as a linear function of input features, which is not apparent with the target variable consisting of only two classes when the linear combination of features can be any value from -∞ to ∞ (Burkov, 2019, p. 25).

Logistic regression is a binary classifier, which estimates the probability that an example belongs to a specific class c. It then predicts class c, if the probability is higher than 50%; otherwise, it predicts the opposite class. The probability is calculated as follows. First, a weighted sum of the input features is calculated, to which a bias term is added. This result is then passed to the sigmoid function shown in Figure 5, which returns the number on the interval (0, 1) (Géron, 2017, pp. 137–138).

Figure 5: Sigmoid function



Source: own work.

The sigmoid function is defined as:

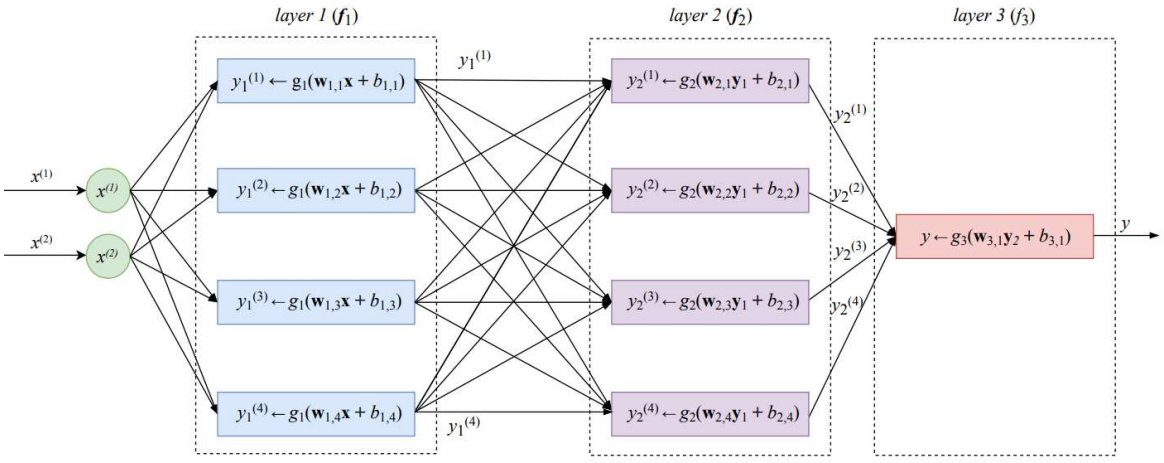
$$sigmoid(z) = \frac{1}{1 + e^{-z}} \quad (1)$$

Instead of minimising the average loss, logistic regression uses maximum likelihood as the optimisation criterion. This means that the model is trained by maximising the likelihood of the training set according to the model. The likelihood function comes from statistics and defines how likely it is for the example to belong to a specific class.

1.4.2 Neural networks

Inspired by human brains, artificial neural networks are powerful and scalable models, which can deal with complex machine learning tasks (Géron, 2017, p. 257). Similar to the brain structure, an artificial neural network consists of neurons that are organised in some way. One architecture of neural networks is the multilayer perceptron (MLP), shown in Figure 6, which consists of neurons, organised into multiple layers. In this particular case, the input layer has two neurons (leftmost green units), two hidden layers have four neurons each (blue units in the first hidden layer and violet units in the second hidden layer), and the output layer has one neuron (rightmost red unit). The input to this network is a two-dimensional feature vector $[x^{(1)}, x^{(2)}]$, and the output is a value y (Burkov, 2019, p. 62).

Figure 6: A multilayer perceptron with two hidden layers



Source: Burkov (2019, p. 63).

A neural network is a nested mathematical function $y = f_3(f_2(f_1(x))$, where each f_l represents one layer of a network and is defined as:

$$f_l(z) = g_l(W_l z + \boldsymbol{b_l}) \quad (2)$$

The neurons in two consecutive layers are connected with directed links. In the beginning, each link is assigned some random weight, and each neuron is assigned a random value called bias. The weights of edges that point to the neurons in layer l are stored in a matrix W_l. The biases for neurons in layer l are stored in a vector $\boldsymbol{b_l}$. The function g is called an activation function and is usually nonlinear to ensure that the network can approximate nonlinear functions. Examples of commonly used activation functions are TanH and ReLU, defined as (Burkov, 2019, pp. 61–64):

$$TanH(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

$$ReLU(z) = \begin{cases} 0 & if\ z < 0 \\ z & otherwise \end{cases} \tag{3}$$

The information flows through a neural network the following way. The neurons in the input layer simply forward the values of the input features (for the examples in the training set) to the neurons in the first hidden layer. The values of other neurons (rectangle units in Figure 6) are calculated as follows. First, a weighted sum of all inputs (values from the previous layer) is calculated, and the neuron's bias is added. Next, the activation function g is applied to this sum, resulting in the activation value of this particular neuron. The calculated output value of this neuron becomes an input value for neurons in the subsequent layer (Burkov, 2019, p. 62). The generalised version of this calculation represents a neural network with Equation (2).

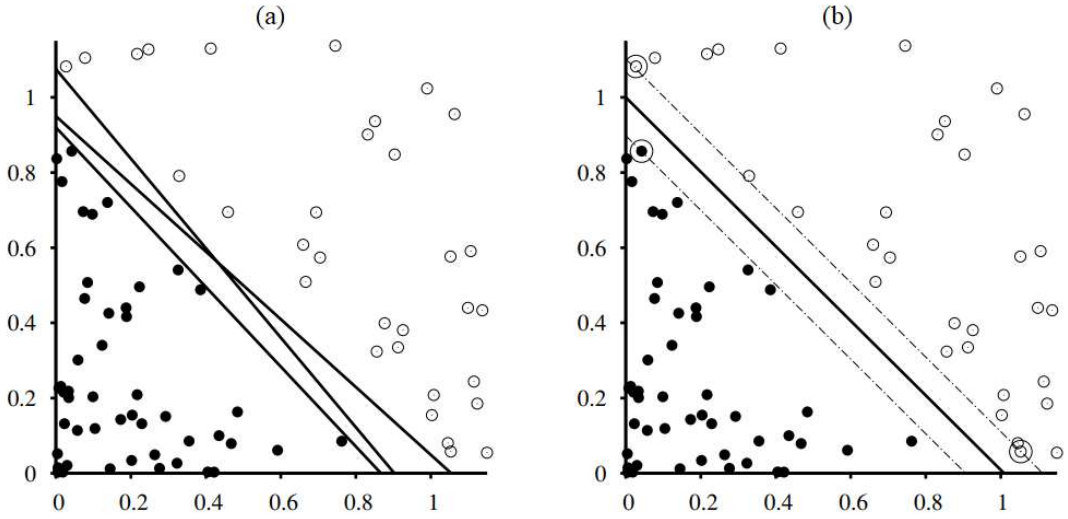
The last calculation in the neural network provides the output y. In case of using a neural network to solve a classification task, value y might differ from the actual class. For the network to learn how to approximate the function that represents the training data, the network uses some measure to calculate the loss. By using the back-propagation algorithm, a neural network is trained by updating the weights of the links connecting the neurons in a network (Russell & Norvig, 2016, pp. 733–735).

1.4.3 Support vector machines

Support vector machines (SVMs) build a hyper-plane or a set of hyper-planes in a high dimensional space, which separates examples. The goal is to construct a hyper-plane with the largest distance to the nearest representatives of any class (Scikit-learn, 2020, p. 263). Figure 7 illustrates a binary classification task for differentiating between black and white circles. There are multiple solutions to separate the two classes successfully; three linear candidates are presented in Figure 7 (a). But which one is the best? The lowest of the three lines correctly classifies all examples and therefore minimises loss, but it is very close to five black circles. SVM chooses to minimise expected generalisation loss rather than minimising expected empirical loss (Russell & Norvig, 2016, pp. 744–745).

Generalisation loss is minimised by selecting the separator that is farthest away from all examples. This separator is called the maximum margin separator and is represented by a heavy line in Figure 7 (b). Two dashed lines go through the nearest examples of both classes. The examples on the dashed lines (three circled points) are called the support vectors, and a width of the area that is bounded by dashed lines is called the margin (Russell & Norvig, 2016, p. 745).

Figure 7: SVM for a binary classification task



Source: Russell & Norvig (2016, p. 745).

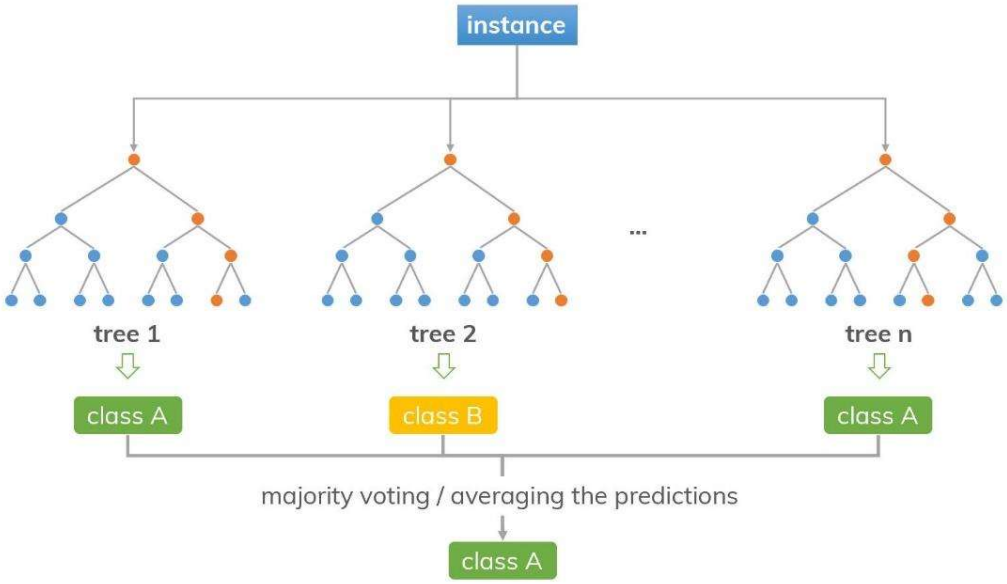
In general, for a binary problem, SVM plots data with n features in the n-dimensional space, and tries to separate examples with an $(n-1)$-dimensional line (i.e., hyperplane) (Burkov, 2019, p. 4). Sometimes, the examples cannot be separated by a hyperplane in their original space. However, if they are mapped into a space of higher dimensionality with so-called kernel trick, they can be separated. SVM uses kernel functions (or kernels) to work in higher dimensions efficiently without explicitly transforming the data. There exist a lot of different kernel functions that can be used for this implicit transformation; the most popular is RBF (radial basis function) kernel, which uses the squared Euclidean distance between two feature vectors (Burkov, 2019, pp. 32–34). Other common kernels are linear, polynomial, Gaussian RBF, and sigmoid (Géron, 2017, p. 164).

1.4.4 Random forest

A decision tree is a classifier that takes as an input a set of attribute values and outputs a decision in the form of a single value. It is determined by the sequence of tests on the input features. When a decision tree is built, on each step, the algorithm chooses one attribute for splitting. This process is iteratively repeated for constructing the subtrees. When the number of examples in a node is too small, or they all have the same label, the algorithm stops splitting and creates a leaf with a particular label, representing the majority class. For

deciding which attribute is the best to split on, different metrics can be used, for example, information gain or Gini impurity (Russell & Norvig, 2016, pp. 697–704).

Figure 8: Random forest with n decision trees



Source: Dinh (2019).

Random forest is an ensemble of decision trees, which means that a group of decision trees gets trained, each on a slightly different subset of the training set. Then, for predicting the class for one example (in case of classification; for the regression problem, the average is predicted), random forest computes the predictions of all classifiers (i.e., decision trees), and using the voting mechanism predicts the class with most votes (Géron, 2017, p. 183). For example, Figure 8 illustrates a random forest with *n* decision trees. When a prediction for a new instance is made, each simple decision tree makes its own prediction by following the path from the root node of a tree to its leaves. Each internal node presents a test on the input features, while leaves contain labels. For classification, the final prediction is then made as a majority vote of individual predictions. The illustrated random forest with *n* decision trees predicts the majority vote *class A* for this particular instance.

1.5 Evaluation metrics

When evaluating the classification model, one logically thinks of the accuracy, which is the most common performance metric. Accuracy tells us the ratio of correctly predicted examples to all made predictions and is useful in cases when errors in predicting different classes are equally important. Table 1 presents a confusion matrix for a binary problem. For each class, there exist examples that were classified correctly, that is, the number of true-positives (TP) and true-negatives (TN); and examples that were wrongly classified, that is, the number of false-positives (FP) and false-negatives (FN) (Burkov, 2019, pp. 55–56).

Table 1: Confusion matrix for a binary problem

		Predicted	
		Positive	Negative
True	Positive	TP	FN
	Negative	FP	TN

Source: own work.

The accuracy is then defined as:

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{4}$$

However, when dealing with an imbalanced data set, there exist other, more appropriate metrics, for example, precision and recall. Precision (a measure of exactness) is the ratio of correct positive predictions to the total number of positive predictions. Ratio (a measure of completeness) is the ratio of correct positive predictions to the total number of positives examples. They are defined as (Branco, Torgo, & Ribeiro, 2015, pp. 5–6):

$$precision = \frac{TP}{TP + FP} \tag{5}$$

$$recall = \frac{TP}{TP + FN} \tag{6}$$

In practice, it is often impossible to have both high precision and high recall, and we have to choose between them. For example, we want to have a high recall, so we come to piece with low precision (Burkov, 2019, p. 56). But sometimes it is preferable to combine both metrics into a single metric called f_1-score, which is defined as:

$$f_1 = 2 \times \frac{precision \times recall}{precision + recall} = \frac{TP}{TP + \frac{FN + FP}{2}} \tag{7}$$

The f_1-score balances the values of precision and recall metrics, as it represents their harmonic mean. Compared to the standard mean, the harmonic mean does not treat all values equally but gives more weight to low values. Consequently, the f_1-score will only be high if both recall and precision are high (Géron, 2017, p. 88).

2 DEBT COLLECTION

The purpose of this chapter is to describe the process of debt collection, which is often divided into two phases, that is, the prelegal and the legal phase.

It is imminent that some customers will have trouble fulfilling their obligations on time. The insolvency of the customers can have an impact on the company's liquidity and therefore, may slow down the growth of the company. The debtor's insolvency may also have far-reaching effects. As stated by Rinaldi & Sanchis-Arellano (2006), consumers insolvency does not only affect companies, but the increase of the debt-income ratio may also have an impact on macroeconomic and financial stability. To get a better picture of how much debt can accumulate over a year in a country, we can take a look at the data on private debt available on Eurostat. In the year 2018, Slovenian private debt reached 33,290 million euro, which represents 72.8% of the same year's GDP. This data includes liabilities held by the non-financial corporations, non-profit institutions serving households, and households themselves. Debt securities and loans were taken into account for the calculation of private debt. Although this may sound a lot, if we look at the whole picture, we notice that the debt is decreasing after the global economic crisis of 2008 (Eurostat, n.d.).

In the corporate sector, access to credit can enable businesses to grow or allow them to exist, but a massive accumulation of debt can also pose a high risk. Recently, the International Money Fund (IMF) raised concerns about the problem of corporate debt in their Global financial stability report (2019). Easy financial conditions and easy access to credit encouraged financial risk-taking, which lead to a sharp increase in corporate debt. "In a material economic slowdown scenario, half as severe as the global financial crisis, corporate debt-at-risk (debt owed by firms that are unable to cover their interest expenses with their earnings) could rise to $19 trillion—or nearly 40 per cent of total corporate debt in major economies—above crisis levels" (International Monetary Fund, 2019, p. ix). In other words, many companies would make themselves insolvent through risky credit.

Debt is not problematic only for the businesses that find themselves on the debtor's side, but also for companies that are affected by the loss of revenue. According to Intrum's (2019b) European Payment Report for the year 2019, the average European loss of revenue due to bad debt has increased from 1.69% in 2018 to 2.31% in 2019. The leading consequences of late payments reported by Slovenian companies were the loss of income and the company's growth limitation.

In many cases, debtors are ordinary people unable to repay the credit and to make ends meet. The findings of Intrum's (2019a, p. 4) European Consumer Report show that almost half of the European consumers surveyed say that the cost of living is increasing faster than the income. Many claim that the concerns and stress of the rising cost of living have negative effects on their wellbeing. One in four consumers claims that they need to borrow money to be able to pay bills, which is an increase from one in five in 2018.

To limit the risk and prevent the loss of revenue, companies must take preventive measures. In the event of the ineffectiveness of the measures, the companies must embark on debt management, that is, debt collection.

2.1 Process of debt collection

For each provision of goods and services, an invoice must be issued to the customer. The invoice must be paid until the due date. Until the payment is not paid, it is considered as a receivable. The Slovenian accounting standard defines receivables as the right to demand the payment of the debt from a person. It is a matter of matching one party's contractual right to receive money, with the other party's corresponding duty to fulfil the obligation (Slovenski računovodski standardi 2016, 2015). When the receivables are overdue, the process of debt collection begins.

Debt collection is the process, in which the creditor attempts to recover loans and credit that have not been repaid by the customer. The process can be handled internally by the creditor itself, or it can be sold to an external debt collection company. The latter may also be referred to as debt recovery (Fay, n.d.). Wejer-Kudełko and Łada (2018) argue that the process of debt collection is a multidimensional phenomenon and that the success of the debt collection is dependent on a variety of legal, economic, and psycho-sociological issues. There are typically two phases of the debt collection process. The first one is the prelegal phase, where the creditor kindly reminds the debtor of the repayment of overdue receivables (via email, text message, phone call, etc.). If the debtor does not repay the overdue receivables, a legal proceeding can be initiated to enforce the debt repayment legally. The legal proceedings can be very complex and longlasting. The legal process depends on the type of debt and its characteristics, as well as the debtor's legal status (i.e., whether the debtor is a natural or legal person). While the legal process is highly reliant on the local legislation, the prelegal process can be quite similar in many countries.

According to Prek and Rems (1999, p. 5), the efficiency of the debt collection is dependent on the knowledge of the people involved in business transactions. Furthermore, they indicate the importance of reliable and efficient systems, such as a payment transaction system and a legal system, which enables a timely legal recovery of liabilities or a way to secure them. Furthermore, Stanič (2012, p. 9) claims that the debt collection efficiency is also dependent on the age of debts submitted for recovery, as well as the available debtor's data and the country, in which the debtor is located. Last but not least, the quality of the collection process also affects the outcome.

2.1.1 The prelegal process

Prek and Rems (1999, p. 58) claim that non-payments or late payments are today the rule rather than the exception. This can be opposed in a variety of ways. To further elaborate on their perspective, there is no unique way on how to handle the prelegal process. Different companies and different authors each have their guidelines and techniques to manage the prelegal process. In general, it is a process where the creditor warns the debtor about the outstanding payment through various communication channels and tries to collect the outstanding payment without the need to start a legal process. A typical activity in the

prelegal process are reminders, which escalate in severity and express the urgency of repayment through different stages. If the debtor does not repay the debt in the prelegal phase, the creditor may continue to pursue the debt collection in court or may decide to terminate the collection process at this stage.

Debt repayment in the prelegal process is cheaper, faster, and more flexible compared to the legal process. There are no court or lawyer costs, the collection time frame is shorter, and it even allows further communication or continued cooperation with the debtor if the prelegal collection process is successful. That is why the debt repayment in this step is the desire for the creditors. The main advantage of the prelegal process is that it offers instalment repayment of the debt. This is especially beneficial in cases where the debtors are unable to pay the debt in a lump sum, as is the case for the judicial recovery in the legal process (Stanič, 2012, p. 9).

Sometimes important information about the debtor is missing. To gain the missing information, the debt collection party has to perform skip tracing, which is an essential part of the prelegal process. Skip tracing is the process of gathering missing information about the debtor, such as the debtor's contact information. The creditor can obtain information about the debtor in the prelegal phase before the commencement of court proceedings based on an authentic document (e.g., an invoice relating to the sale of goods and services) or after the initiation of the legal process upon receipt of a certificate of finality. The Personal Data Protection Act regulates the acquisition of personal data. Accordingly, certain restrictions apply to the processing and retrieval of data. It is, therefore, a good idea for the creditor to think about this early and adjust the contract in a way that allows him to obtain personal data with the consent of the client. This way, the creditor will be able to get information about the debtor before applying for enforcement, and will thus be able to assess the success of any enforcement proceedings in advance realistically. Otherwise, the creditor is forced to take the risk of the accuracy of the information at his disposal. There are different types of data queries based on the type of data:

- the query for identification data (e.g., birth date, registration or tax number),
- the query for address data (e.g., permanent, temporary and actual residence), and
- the query for information on assets (e.g., motor vehicles register, register of pledged movables assets, and land register) (Horvat & Guzej, 2010, pp. 53–58).

The missing information obtained is not only useful in the prelegal process to successfully collect the debt, but also in filing a motion for enforcement in court, as it requires a lot of debtor's personal information (Volk, 2003, p. 52). The cost of skip tracing is not negligible for the creditor. The research on the impact of skip tracing on the debt collection process dates back to the mid-20th century. Mitchner and Peterson (1957, pp. 527–528) explored the optimum pursuit duration and maximum expected profit with the use of a mathematical model for different types of delinquent loans. In their analysis, the pursuit duration for debt collection processes that include skip tracing is drastically shorter compared to the processes

where skip tracing is not performed. According to their estimate, the pursuit cost of debt collection is six times higher in processes involving skip tracing.

2.1.2 The legal process

The court has an obligation to provide legal protection by applying an abstract legal norm to the established factual situation and impose a legal consequence. When a debtor does not fulfil what was imposed on him by the court decision, it is necessary to enforce the court decision decisively. The creditor has to request the intervention of the state or its bodies. Only they are entitled to use force to establish the situation required by the court decision (Volk, 2003, p. 11).

Enforcement proceedings are defined as procedural actions that, with the help of state coercive means, should establish between the debtor and the creditor such a situation as the creditor has the right to demand based on the enforcement title or enable future enforcement of the creditor's claim (Volk, 2003, p. 11).

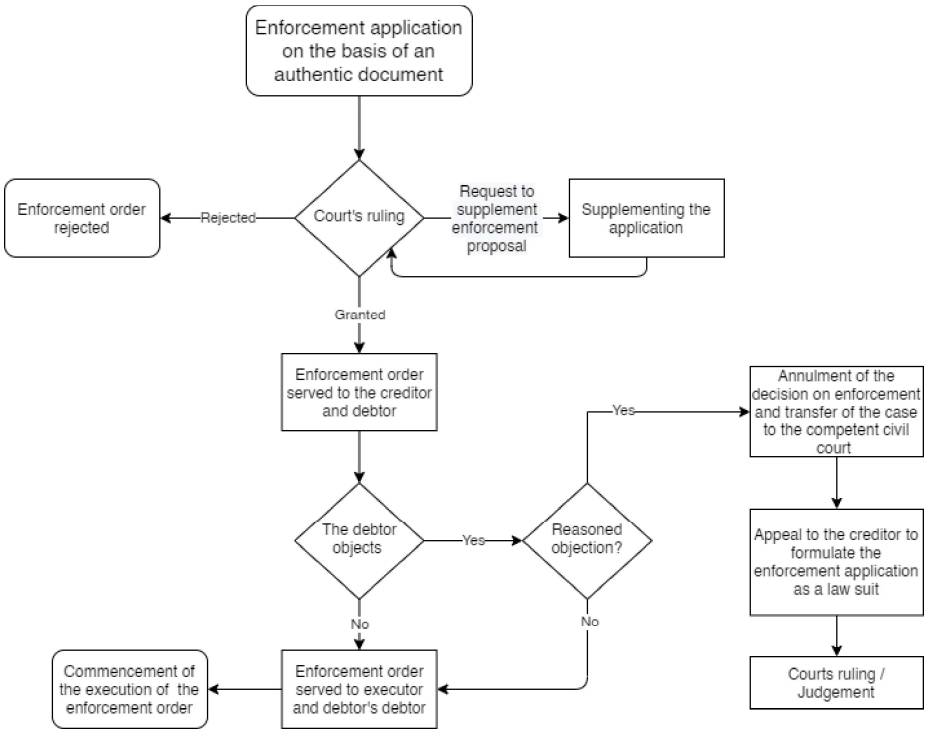
If the collecting party believes that the debtor will not voluntarily fulfil their obligation because they cannot or do not want to do so for various true or untrue reasons (e.g., waiting for the claim to be statute-barred, hoping the creditor will get tired collecting the debt, or thinking that the creditor will simply forget about the debt), they still have the possibility that the fulfilment of obligations is demanded. This can be achieved with a lawsuit, by the introduction of an enforcement procedure, or an insurance procedure if the right conditions are met (Prek & Rems, 1999, p. 34).

Throughout the legal proceedings, the creditor finances the debtor by paying court and other fees. The creditor also needs to provide professional assistance or representation, which can be costly. Assessing whether it is worthwhile to initiate a legal proceeding is, in essence, a business decision. It must be borne in mind that the costs of legal proceedings can be high, and they have to be paid in advance. At the same time, it is not sure if the debtor has the means to repay the debt and will not file for insolvency before the end of proceedings. Sometimes it is just not worth, based on the amount of debt, to initiate judicial debt collection (Prek & Rems, 1999, p. 34).

The creditor must initiate enforcement proceedings, which can be done in two ways. The first one is based on an authentic document (e.g., an invoice, a bill of exchange, an extract from the accounting book) and the second is based on an enforceable title (e.g., enforceable court decision and court settlement, directly enforceable notarial record). The difference between the two procedures is reflected in the method of service. In the case of enforcement based on an authentic document, the court's decision is served to the debtor and the creditor immediately, while the executor and the debtor's debtor (e.g., bank, employer) is served after the court's ruling has become final. In the case of an enforceable title, the decision is served to all parties immediately and simultaneously. The set of objectionable grounds is greater in

the case of enforcement based on an authentic document, and the debtor's objection itself suspends the enforcement (Lajevec, 2019).

Figure 9: Simplified enforcement process based on an authentic document



Source: Adapted from Lajevec (2019); Zakon o izvršbi in zavarovanju (ZIZ) (1998); Horvat & Guzej (2010).

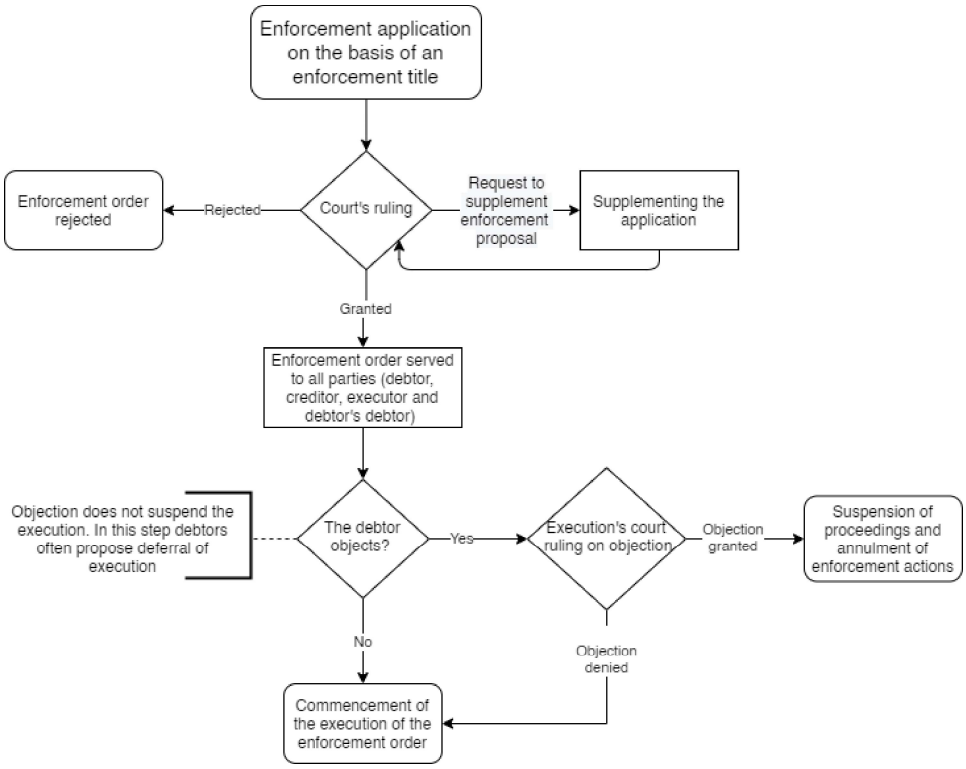
On the other hand, the set of objectionable grounds is smaller in the case of enforcement based on an enforceable title and the objection itself does not suspend the enforcement, as it has already been served to the debtor's debtor. The creditor himself chooses the means of enforcement, which relate to the method of settlement or the assets with which the creditor's claim is repaid. The subject of enforcement relates to every debtor's object or property or material right to which enforcement is permitted. The types of means of enforcement are divided into means to recover monetary and non-monetary claims as follows.

1. Execution means for monetary claims:
    - enforcement on movable assets,
    - enforcement on the debtor's monetary claim,
    - enforcement on salary and other remunerations - garnishment,
    - enforcement on debtor's funds,
    - enforcement on book-entry securities,
    - enforcement on shareholder's share, and
    - enforcement on real estate (i.e., foreclosure).

2. Execution means for non-monetary claims:

- forced eviction,
- an obligation to do or allow something,
- handing over a thing, and
- return of the worker to work (Lajevec, 2019).

Figure 10: Simplified enforcement process based on an enforceable title



Source: Adapted from Lajevec (2019); Zakon o izvršbi in zavarovanju (ZIZ) (1998).

The simplified typical legal collection processes for both methods of enforcement are outlined in Figure 9 and Figure 10.

The legal process is by many viewed as very complex and inefficient. Stanič (2012, p. 9) compares the prelegal and legal processes, emphasising the agility of the prelegal process through the directness and the immediacy of communication as advantages. During the enforcement, the debtor has the option to object, which may delay the proceedings considerably. Compared to the legal process, the debtor has little opportunity to object in the prelegal phase, where he is simply confronted with the facts by providing him with evidence substantiating the creditor's claim. Stanič argues that the legal process, that is, judicial debt collection, is rigorous and procedurally demanding.

The area of legal debt collection has often been the subject of criticism, mainly due to a massive case backlog. To address some of the issues, the legal debt collection process has undergone significant changes. Through adopting ever new regulations, an attempt was made to establish a system that ensures effective enforcement. Some believe that the

situation has improved in recent years, especially in the first stages of enforcement, mainly through the introduction of modern technology in court proceedings (Horvat & Guzej, 2010, p. 5).

On the other hand, Volk (2015, p. 9) argues that the constant changes in the legal process have had a detrimental effect on the understanding of the process and have raised several issues in practice. Some critical parts of the process remain insufficiently regulated.

Volk (2015, p. 27) continues that the field of judicial debt collection can be effectively regulated only by the legislator drafting a completely new law, which should be based on the modern and comparable legislation of other Europen countries. The constant supplementation, revision, and amendment of the current legislation, which is in place from 1998, is a dead end, especially because many of the new changes to the legislation end ingloriously with the annulment decisions of the Constitutional Court.

3 USE OF MACHINE LEARNING IN DEBT COLLECTION PROCESS

This chapter concludes the theoretical part of the thesis by providing an overview of the research conducted on the use of machine learning in the field of debt collection.

Machine learning has found its way in many different industries. It proves to be very useful in industries where large amounts of data are available. The more data there is, the more reliable results can be generated. A lot of data is created in the process of debt collection, which can then be used to predict the outcome of the debt collection process or to improve the effectiveness of the process.

Before the terms machine learning or data mining became mainstream, gathering information from data was mainly referred to as statistical modelling. One of the first studies about the use of statistical methods in the field of debt collection is a research study of the collection of defaulted loans by Mitchner and Peterson in 1957. The study refers to the Loan Adjustment Department (LAD) of the Bank of America, which works as a collection agency for the bank. The study includes a detailed description of the work in the field of debt collection, which is still very similar to the present-day work in the field of debt collection, particularly compared to the prelegal collection process. The most significant difference, of course, is the use of technology today, but the basics have mainly remained the same. The research focuses on three problems: (1) loan pursuit strategy, (2) number of cases assigned to one collector, and (3) the distribution of effort between paying and non-paying accounts. These three problems are closely related to the net profit of LAD, which is a function of these factors. The research mainly focuses on the loan pursuit strategy, which is presented as an analogy of a game of poker. The player needs to continue betting each round before the completion of a hand, and each round, new cards are dealt, which means that new information is available. The player has to decide whether to continue playing to remain in

the game or to throw in his hand. The same principle applies to debt collection, where the collector decides based on the information available to continue to pursue the collection process or to stop it. As the debt without any payments ages, its value to the collector declines, while the costs rise. After a certain time, based on the debt amount, it is better not to pursue the collection and to save the costs that would arise pursuing the collection further. Based on an optimal pursuit duration and the maximum expected net profit of debt, the authors constructed a statistical simulation. The simulation showed that following an optimal pursuit strategy could increase the profit by 33% compared to pursuing debt collection over the optimal pursuit duration. The increase of the profit is based on the effort reduction in the pursuit of debts that are more expensive to manage and the resulting reduction in costs.

The use of machine learning in the debt collection industry is often used to predict the probability of a debt being repaid, that is, classifying it as good or bad debt. One study, comparing neural networks and traditional statistical techniques, dates back to the mid-nineties. Desai, Crook, and Overstreet (1996) explored the ability of neural networks and modular neural networks alongside traditional statistical techniques, such as logistic regression and discriminant analysis, in building credit scoring models. At the time, it was thought that traditional methods such as linear discriminant analysis and logistic regression were less successful than neural networks. However, neural networks were still perceived as work in progress and as lacking robustness. The problem was the continuous validity over time and a wide range of conditions. The models, which were promising on paper, collapsed after the deployment. The results of the study show that in terms of efficiency of bad loan identification as the criterion, the neural networks performed a little better than the traditional techniques. On the other hand, if the criterion was the identification of good and bad loans, the performance of logistic regression models was comparable to that of the neural networks.

In 2009, a statewide novel approach for tax collection optimisation was launched on a state level by the New York State Department of Taxation and Finance. The solution is based on data analytics and optimisation through the framework of constrained Markov Decision Process (MDP). The model tries to answer the following questions: (1) which debtors should be approached, (2) which of the available collection actions should be taken onto them, (3) who should make those actions, and (4) when to take them. The answers to these questions depend on several factors, for example, available demographic information about the debtor, amount of debt, resources available, etc. (Abe et al., 2010). The system is built in such a way that it optimises collection to maximise long-term returns while considering complex dependencies among resources, business needs, and legal constraints. The authors claim that the solution provides an unprecedented level of decision automation while optimising the collection. The system was launched in December 2009. The annual increase in revenue was about 8% at 83 million dollars. It was estimated that the expected additional tax revenue over three years after deployment would add up between 120 and 150 million dollars, exceeding the target value of 99 million dollars. Even though the solution provided

promising results, it cannot handle every case. The more complex cases are separated from the general ones and are assigned to a field agent. The good news is that the average age of a case assigned to a field agent has decreased by almost 10 per cent (Miller et al., 2012).

Unlike a financial company that offers loans, a debt collection company has little information at their disposal about the debtor. Data that is handed over includes information about the debt, occasionally lacking even the debtor contact data, which they have to obtain themselves. As discussed in Section 1.3.4, lack of relevant features can substantially decrease the performance of a machine learning algorithm. The focus of the following research is to address the problem of missing debtor's data regarding bad debt classification. The University of Louisville carried out the research with the focal point of the debt collection in the healthcare industry. Bad debt presents a significant issue for the health care industry in the USA. Unpaid bills and bad debt significantly contribute to the rising cost of healthcare. The established process where hospitals hand over debts to collection agencies is becoming increasingly ineffective. Hospitals end up paying between 30 and 50 per cent of the recovered bad debt revenue to the outside agencies. The situation led to a trend where hospitals were taking harsh legal actions towards the debtors. In some cases, the situation escalated in the arrest or even imprisonment of the debtors (Zurada & Lonial, 2011).

The healthcare institution that provided the data for the research was able to recover only about 7% of debt by non-paying patients. A reliable distinction of good and bad debt would allow the institution to focus primarily on good debt and save administrative expenses on debts considered as bad. To obtain this information, five different machine learning models were tested (memory-based reasoning, neural networks, logistic regression, decision trees, and an ensemble model consisting of the latter three). Since the healthcare institution does not have access to other financial or demographic information, the only variables included in the research were the debtor's age and gender, the injury diagnosis code, and the amount of the claim. The best-performing models were neural networks alongside with logistic regression and the ensemble model. The neural networks were able to classify almost 35% of unknown cases as good cases, potentially yielding about 420,000 dollars in additional revenue (Zurada & Lonial, 2011).

Nowadays, the data regarding the collection is stored in databases and is managed by programs, which facilitate the collectors' work. The debt collection process often consists of predetermined time-based steps that are automatically executed, generally on a one-size-fits-all policy. Typically, they are defined as timelines that perform activities based on matching conditions in each particular collection process. The next case shows how to use machine learning to make the schedule of each process more flexible and tailor it based on the next best activity to maximise the collection. Research carried out by van de Geer, Wang, & Bhulai (2018) specialises in data-driven scheduling of outbound calls. The idea behind this is that each day, only a limited number of calls can be made. Therefore, each day, an algorithm selects the debtors to call that day. The algorithm determines the likelihood with which a debtor is going to repay its debt. The likelihood is then multiplied with the size of

the debt obtaining an approximation of the expected value of a debtor considering its current state. Based on this value, the marginal value of a phone call is determined for each debtor. The debtors with the highest marginal values are prioritised for a phone call. Because the outcome of a call is uncertain, the extent to which a call will benefit the collection process is hard to determine. It depends on features like time since the previous call, outcome of the previous call, time of the month, amount of debt, and the persuasiveness of the agent calling. The machine learning algorithm selected was gradient boosted decision trees (GBDT). The comparison of the incumbent policy (IP) and the GBDT-optimised calling policy (GOCP) showed that GOCP collected more debt in less time with fewer resources. GOCP was able to collect 14% more outstanding debt than IP. The average number of days until complete repayment dropped from 22.2 to 20.3 days. The number of outgoing calls was reduced by almost 22%. In total, there was a 47% increase in the monetary amount collected per call made by GOCP compared to IP.

A good example of what can be achieved with the use of machine learning to gain insight into data illustrates the partnership between the debt collection company EOS KSI Česká Republika and Tibco, a provider of advanced analytics software solutions. According to them, insight into data is the key to success. Past debt collection processes can be used to gain a deeper understanding of the collection process itself. One can learn which debts are most likely to become delinquent, which delinquent debts are likely to be collected and which will be a waste of resources, when to seek payments, and what are the best means to collect the debt for a particular customer. It is impossible to get insight into data without efficient data mining and analysis tools (Statistica, 2017, p. 3).

Collecting a debt can often be like walking on thin ice. Businesses cannot survive if their customers do not pay their bills. However, if companies attempt to collect debts from delinquent customers in an aggressive manner, then the customer is usually lost forever. The key is to find the right path without hurting long-term viability. This problem can be tackled through gathering more insight into the data to support the decision making for each particular debt about whether or not to go to court, in which phase of the process should the case be especially focused on, or to skip a standard collection step. In the case of EOS KSI Česká Republika, this means that using data such as type of the client, the nature of the debt, region, and available contact information, will allow the company to forecast whether a particular debt collection process will be successful. Furthermore, they could predict what modification to the process should be undertaken to improve the chances of success. Previously, decisions like these were made based on the ‘gut instinct’ and were not supported by data (Statistica, 2017).

With the investment in extracting information from data, the company gained the ability to drive the collection process based on the payment probability, debtor characteristics, and their behaviour. In practice, this means that the collection process is not predetermined but varies according to the factors stated above. For each case, an algorithm determines what the next best activity to maximise the collection is. According to EOS KSI Česká Republika, the

solution allowed them to focus only on the debts that are most likely to be paid, which notably reduced the administrative and call centre costs. They claim that such a solution saves them the time that would be spent on cases with a low likelihood of successful collection, additionally to 52 hours saved each month due to tasks that were automated (Statistica, 2017).

4 BUSINESS CASE

This research is carried out in collaboration with a debt collection company from Slovenia. Due to their desire for anonymity, the company's name cannot be disclosed. This company is one of the leaders in the field of debt collection in the country. It provides the services of debt purchase and managing debt collection for other clients as an external service. The company is aware of the power the data holds and the potential of insight it provides for further development of the company. Given the fact that efficient and safe data management is of paramount importance in the field of debt collection, the company regularly invests in information technology and standardisation. In light of finding new solutions and ways to improve the efficiency of recovery, the company is also exploring the use of the latest technology.

It has long been known that the value of debt decreases over time. Mitchner and Peterson (1957) said that as the debt ages without any payments, the value of the debt declines while the costs rise. After a certain amount of time, it is better not to pursue the collection and save the costs.

Debt collection companies usually make money in two ways. First, other companies outsource the process of debt collection to a debt collection agency. They sign a contract and agree on the commission, which is usually dependent on the debt collection performance. The second way these companies make money is through the purchase of an overdue or outstanding debt. The purchase value of the debt depends on the likelihood of the debt being collected. In the case of outstanding debt, the debt purchase value can be as high as 85% of the initial debt value (Stanič, 2012, pp. 8–11). The profit in the second case equals the collected amount minus the debt purchase cost and the operational expenses to collect the debt. The ability to successfully predict the outcome of the debt collection process enables the company to determine debts that are more likely to be successfully collected. Shifting the focus from debts that are not likely to be collected to the ones that are, can have a positive effect on the collected amount, while at the same time, it reduces the operational cost of collection for debts that are not likely to be collected. Therefore, increasing profit and becoming more competitive in the market.

It is not easy to predict how a debt collection process will turn out. Without the use of advanced algorithms, companies have to consider other ways to determine which debts have a higher likelihood of success than others. According to the company that provided the data,

the current process is quite complex. It consists of assessing the likelihood through available data such as the age of the debt (older debts are harder to collect), the amount of the debt (larger debts are harder to collect) and other factors, for example, whether the debt is secured and what collateral is used to back up the debt.

This research serves as a pilot study to explore the possibility of using machine learning to enhance the debt collection process by developing different machine learning models to predict the debt collection outcome.

The business case aims to determine how successful are machine learning algorithms at predicting the outcome of debt collection in two time points. Firstly, on data that is typically available before or at the start of the process of debt collection, and secondly, one month after the start of the debt collection process when more data is available.

Models constructed in the scope of this research are not expected to predict the debt collection process perfectly. After all, the machine learning models are only as good as the data, which they are trained on. Nevertheless, this research can be an indicator of whether it is worthwhile considering machine learning algorithms for debt collection outcome prediction and only the first step towards broader use of machine learning algorithms in the company.

Upon debt take over, the debt and debtor data usually covers only the essential information necessary for the debt collection process. This data typically includes information about the debtor (e.g., name, address, telephone, and birthdate) and information about the debt (e.g., amount of debt, debt date, and debt due date). The lack of relevant data poses a threat of ineffective outcome prediction. Therefore, in the scope of the business case, two sets of models are constructed. The first set of machine learning models is based on the data that is available at the time of debt take over and predicts the outcome at the start of the debt collection process. In anticipation that the first set of models performs worse due to the lack of relevant features, the second set of models is constructed. These models predict the outcome based on data that is available one month into the debt collection process. At this point, more data is available for the models to base their predictions on. Added data includes information about communication, steps undertaken to collect the debt, payment behaviour of the debtor, and much more.

Debt collection processes that include court proceedings have different timelines and can be very distinct from those that do not include legal proceedings. The number of processes that include legal proceedings is lower, but these processes can be much more complicated. Therefore, it was decided (together with the company) to reduce the complexity and the risk of reduced performance by not mixing both types of processes and to take only non-litigation processes into account.

One of the concerns in this research is a rather small data set of 13,250 debt collection processes, which could affect the prediction performance of the models. Banko and Brill

(2001) have demonstrated the importance of data quantity for machine learning models accuracy, stating that it may be wiser to invest time and money into getting more data than into algorithm development.

The most common metric to evaluate classification models is accuracy, which represents the overall predictive performance by dividing the number of correct predictions with the number of total predictions, defined with Equation (4). However, when the data (target variable distribution) is imbalanced, accuracy may not be the best metric to use. In our case, the target variable can be considered as imbalanced. Its distribution is approximately 1:2 between the negative and the positive class. The problem of using accuracy for imbalanced data sets is that considering a simple model, which always predicts the majority class, the model would have an immediate accuracy of 0.67. However, the model would be unable to predict any examples of the negative class correctly. Branco, Torgo, and Ribeiro (2015, pp. 2–7) argue that in the case of an imbalanced data set, using standard metrics can lead to sub-optimal performance of classification models. To address this issue, they propose to make the algorithms focus on the rare events with the use of special-purpose evaluation metrics, which are biased towards the performance of models on rare events.

In the process of building and evaluating different models, we focus on metrics precision, recall and f_1-score, which are defined with Equations (5), (6), and (7), respectively. The latter is the weighted average of precision and recall. Our decision of metric selection is based on their focus. "Precision is the ratio of correct positive predictions to the overall number of positive predictions … Recall is the ratio of correct positive predictions to the overall number of positive examples" (Burkov, 2019, p. 55). This means that high precision will minimise false-positive predictions, while high recall will minimise false-negative predictions.

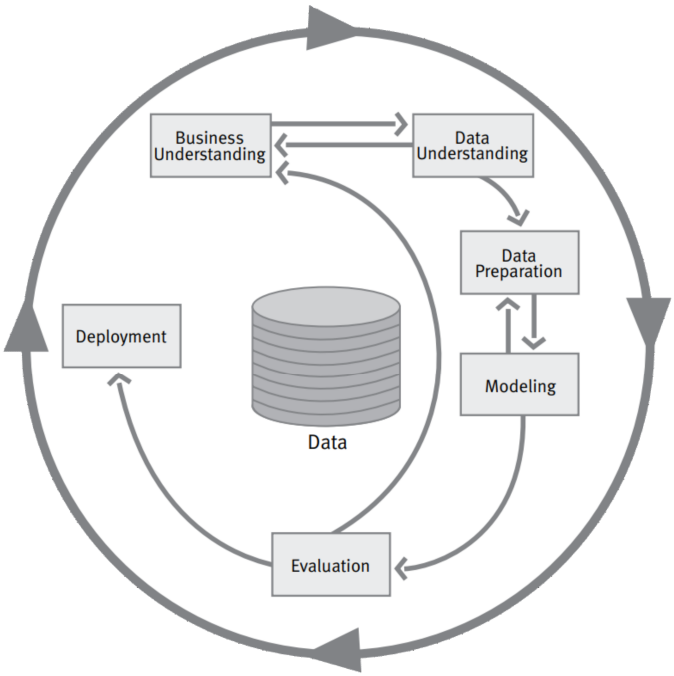
Focusing on high precision means a low number of cases where an unsuccessful debt is predicted as successful. On the other hand, focusing on high recall means minimising the number of cases where a successful debt is predicted as unsuccessful. The latter is more crucial to the company. The cost of a false-positive prediction is lower than the cost of a false-negative prediction. Classifying a successful debt as unsuccessful could lead to a situation where the company does not pursue the collection of a debt that is likely to be collected. Therefore, they lose the whole amount of the debt that they could collect if they continued the collection. On the contrary, predicting an unsuccessful debt as successful would only result in the operational cost increase of pursuing collection for a case that will likely not be successfully collected.

The best metric based solely on minimising the false-negative predictions is the recall metric. However, focusing only on high recall could lead to a situation where the model correctly predicts only the majority class (i.e., successful). Precision and recall are inversely proportional, and the strive for a high recall could mean low precision. Using this strategy, we could end up with a model, which achieves a recall of one by predicting every case as

successful. If the model predicts only one class, it cannot produce any false-negative predictions, only false-positives. The outcome would be the same as assuming that every debt collection process will be successful, and this is not optimal. To avoid such a situation, we use the f_1-score as the primary metric, which strives towards a low number of both false-negative and false-positive predictions. For the comparison of final models, we use the precision-recall curve together with the area under curve (AUC) score, which is the integral of the precision-recall curve.

The primary integrated development environment (IDE) used in this research is Anaconda alongside PyCharm. One of the main libraries used to construct machine learning models is scikit-learn, which is one of the most sophisticated libraries in the field of machine learning. The authors of the library say that "Scikit-learn is a Python module integrating a wide range of state-of-the-art machine learning algorithms for medium-scale supervised and unsupervised problems" (Pedregosa et al., 2011). To construct the neural network model, the Keras library is used. Keras is a deep learning application programming interface running on top of TensorFlow. With almost 400,000 individual users as of 2020, it is one of the most used libraries across the research community and industry (Keras, n.d.-a, n.d.-b).

Figure 11: CRISP-DM project life cycle model



Source: Chapman et al. (2000, p. 10).

For building the models, the cross-industry process for data mining (CRISP-DM) methodology is applied. CRISP-DM methodology provides an analytical approach to plan a data mining project. The life cycle of a machine learning project consists of six phases: business understanding, data understanding, data preparation, modelling, evaluation, and deployment. The sequence of the phases can be flexible since the outcome of a phase

determines the next phase or the next activity. The CRISP-DM project life cycle model is shown in Figure 11 (Chapman et al., 2000, pp. 10–11).

This chapter serves as the phase of business understanding. Phases data understanding, data preparation, and modelling are included in the following chapter, while the evaluation phase is covered in Chapter 6. The final development phase of the CRISP-DM model is out of the scope of this thesis.

5 MODEL CONSTRUCTION

This chapter describes the path from the business case towards fully functional machine learning models that predict the outcome of debt collection processes.

The chapter is divided into two parts. The first part focuses on the first set of models predicting the debt collection at the time of taking over the debt, that is, at the very beginning of the debt collection process. The second part centres around the second set of models predicting the outcome one month into the debt collection process.

We build different machine learning models to predict the outcome of the debt in the two time points. They are described in detail in Sections 5.1 and 5.2, respectively. Both sets of models are presented as follows. First, the data used for modelling is described. Then the process of data preparation is presented, which includes data cleaning, feature scaling, and feature selection. Lastly, the modelling is described through model selection and model optimisation.

5.1 The first set of models – before the start of the debt collection process

In this section, we focus on building models that predict the debt collection outcome at the very beginning of the debt collection process.

5.1.1 Data understanding

This section covers the description of the data used in the first set of machine learning models.

Table 2 shows the information about the features included in the first set of models. The data set includes thirteen features, of which nine are quantitative, and the rest are qualitative. The financial features, that is, *main_claim*, *costs*, *interests*, and *pbi_payments*, contain values submitted by the client and exclude all subsequent accrued costs and interests after the start of the collection process. The target variable is the categorical feature *outcome*. The value 0 represents an unsuccessful debt collection, and 1 represents a successful debt collection.

Table 2: Feature description for the first set of models

Feature	Type	Description
account_type	binary	Is the debtor a person or a company (B2B / B2C)?
from_partner	binary	Is the debt handed over by a partner company (1 - yes / 0 - no)?
skd_first_level	categorical	Creditors' first level for the standard classification of activities.
main_claim	continuous	Value of main claim [€].
costs	continuous	Value of costs [€].
interests	continuous	Value of interests [€].
pbi_payments	continuous	Payments paid to the creditor before debt take over [€].
debtors	discrete	The number of debtors at the time of debt take over.
addresses	discrete	The number of addresses at the time of debt take over.
phones	discrete	The number of phones at the time of debt take over.
emails	discrete	The number of emails at the time of debt take over.
dpd	discrete	Days past the due date of the debt.
outcome	categorical	The outcome of the debt collection process (0 / 1).

Source: debt collection company.

Having a look at the mean and standard deviation values of financial information in Table 3, one can notice that there are quite a few outliers present in the data. Together with the company that provided data, we discussed different possibilities to reduce the number of outliers without drastically reducing the number of examples in the data set. In the end, we decided to remove only the most noticeable outliers in each of the financial features. The debt collection company set the cut-off point based on their experience and domain knowledge to reduce outliers and preserve the diversity that is normal for each of the features. We also set a new condition to exclude all debts, whose total sum of all financial features included in the analysis does not exceed one euro. This measure was taken into account as a precaution since there are some examples with very low amounts of total debt. For example, the minimal value of the *main_claim* feature could be as low as 0.01€.

Table 3: Numerical feature information for the first set of models

Feature	Count	Mean	Std	Min	25%	50%	75%	Max
main_claim	13,250	459.25	1,950.84	0.01	30.70	81.52	257.20	44,731.66
costs	13,250	10.50	18.49	0.00	0.00	0.83	13.54	471.91
interests	13,250	4.68	80.30	0.00	0.00	0.00	0.00	6,621.61
pbi_payments	13,250	12.76	335.14	0.00	0.00	0.00	0.00	19,507.89
debtors	13,250	1.00	0.78	1.00	1.00	1.00	1.00	2
addresses	13,250	1.00	0.30	0.00	1.00	1.00	1.00	7
phones	13,250	0.71	0.86	0.00	0.00	1.00	1.00	13
emails	13,250	0.01	0.40	0.00	0.00	0.00	0.00	7
dpd	13,250	212.94	292.21	-238.00	25.00	117.00	254.00	2,800

Source: own work.

Table 4 shows the result of the action undertaken to reduce the number of outliers. The number of examples decreased by 182 examples to 13,068. Looking at the results, one can see that by dropping a bit more than 1% of examples, the standard deviation decreases significantly. In the case of the *main_claim* feature, it almost halved, it decreased by five times for the *interests* feature and even decreased by almost nine times for the *pbi_payments* feature.

Table 4: Numerical feature information for the first set of models after outlier removal

Feature	Count	Mean	Std	Min	25%	50%	75%	Max
main_claim	13,068	354.8	1,071.37	0.93	31.1	81.5	257.2	14,706.27
costs	13,068	10.14	16.31	0	0	0.83	12	89.4
interests	13,068	2.58	16.19	0	0	0	0	227.98
pbi_payments	13,068	2.79	42.7	0	0	0	0	1252.7
debtors	13,068	1	0.08	1	1	1	1	2
addresses	13,068	1	0.3	0	1	1	1	7
phones	13,068	0.7	0.86	0	0	1	1	13
emails	13,068	0.1	0.39	0	0	0	0	7
dpd	13,068	215.2	292.49	-238	27	118	160	2,800

Source: own work.

Although the removal of outliers has a significant impact on the standard deviation of the data, it remains relatively large. Given that the data set in this business case is already rather small, a joint decision was made to proceed with the current data set.

We continue with a more detailed description of the features grouped by their type into qualitative and quantitative, starting with qualitative features. The *account_type* feature tells us whether the debtor is a legal entity or a person. For about 70% of the examples in our data set, a debtor is a person. The *skd_first_level* feature represents the creditor's business activity according to the statistical classification of economic activities. In our case, the feature holds information about the most general classification level and has fifteen distinct values. The majority of examples in the data set belongs to creditors, which business activity is either information and communication or activities, auxiliary to financial services and insurance activities. The binary feature *from_partner* tells us whether the debt is handed over by a partner company, which is the case in about 13% of examples. Last but not least, categorical feature *outcome* is the target variable, which contains information on whether the debt was successfully collected. The distribution of unsuccessful and successful outcomes is imbalanced with the ratio of approximately 1:2, respectfully.

We are continuing with quantitative features. The *debtors* feature represents the number of debtors. In all but 79 examples, there is only one debtor for each debt collection process. As expected, the address (feature *addresses*) is the most common contact information that is available at the time of the debt take over. In more than 95% of instances, at least one address

information is available. The information availability rate decreases rapidly to only a little more than half of the debts having a phone number (feature *phones*) associated with it. However, only about 8% of all instances have at least one email (feature *emails*) contact available.

The *dpd* feature (i.e., days past the due date of the debt) can be referred to as the debt's age. These are the days between the due date of the debt and the take over date. The lower the number, the greater the likelihood for a successful debt collection. As seen in Table 4, the value can also be negative, which indicates that some of the debts in the data set were still outstanding at the time of taking over the debt.

The *main_claim* feature represents the original amount the debtor owes. The *costs* feature reflects the accumulated amount of costs in the process of collection. Costs often also contain penalties resulting from non-payment, but not all creditors charge costs. The i*nterests* feature represents the accumulated creditor's interest until the debt take over. As is the case with the costs, not all creditors charge interests. The main claim represents more than 96% of the total debt from our data set, while costs represent about 3% and interests less than 1%. The *pbi_payments* feature tells us the amount of the debt that has already been settled before the take over date. The sum of all payments represents just under 1% of total debt. The problem with the payments is that not every creditor reports them. Some creditors hand over the entire debt history along with information about payments made, while others hand over only the debt amount reduced by the sum of the payments. The inconsistent reporting of those payments reduces the usefulness of the *pbi_payments* feature, which we determine later in Section 5.1.2.3.

5.1.2 Data preparation

This section describes the steps of data cleaning, feature scaling, and feature selection for the first set of models.

5.1.2.1 Data cleaning

This step normally covers the handling of missing values, outliers, and data transformation. There are no missing values, and we have already taken care of outliers. Therefore, in this section, we describe the preparation steps undertaken to prepare the data for training models. The first thing to do in terms of data cleaning is to take care of the encoding of categorical variables. Values of the *account_type* feature are encoded following the rules B2B = 0 and B2C = 1. One-hot encoding of the *skd_first_level* feature is done by replacing the existing feature with fifteen so-called dummy variables. In each of the data set instances, only one of the new fifteen variables is positive (with value one), representing the value of the initial feature, while others are set to zero. Normally, when the feature has n distinct values, $n - 1$

dummy variables are created. However, we decide to create as many new variables as there are distinct values so that we can assess all of them in the step of feature selection.

At this point, the data is split into the training set and the test set with the ratio of 80:20, stratified according to the class distribution of the target variable *outcome*. All models will be trained and tested on the training set using cross-validation to reduce model overfitting. The test set will only be used for the final model evaluation in Chapter 6.

5.1.2.2 Feature scaling

Some of the machine learning algorithms do not perform well when the numerical input data is not scaled. The issue is that different features have different value distributions and are therefore hard to compare, for example, features *dpd* and *debtors* have very different scales, the former ranging from -238 to 2,800 and the latter ranging from 1 to 2. Therefore, feature scaling is often regarded as one of the most critical steps. The dilemma of choosing the right scaling technique often arises here. The main options are to use normalisation (commonly also referred to as min-max scaling) and standardisation.

As there is no definitive answer to which technique to use, we test both and evaluate their impact on the models' performance. Therefore, we discuss this topic further in Section 5.1.3.1, where we compare the performance of models trained on both standardised and normalised data.

5.1.2.3 Feature selection

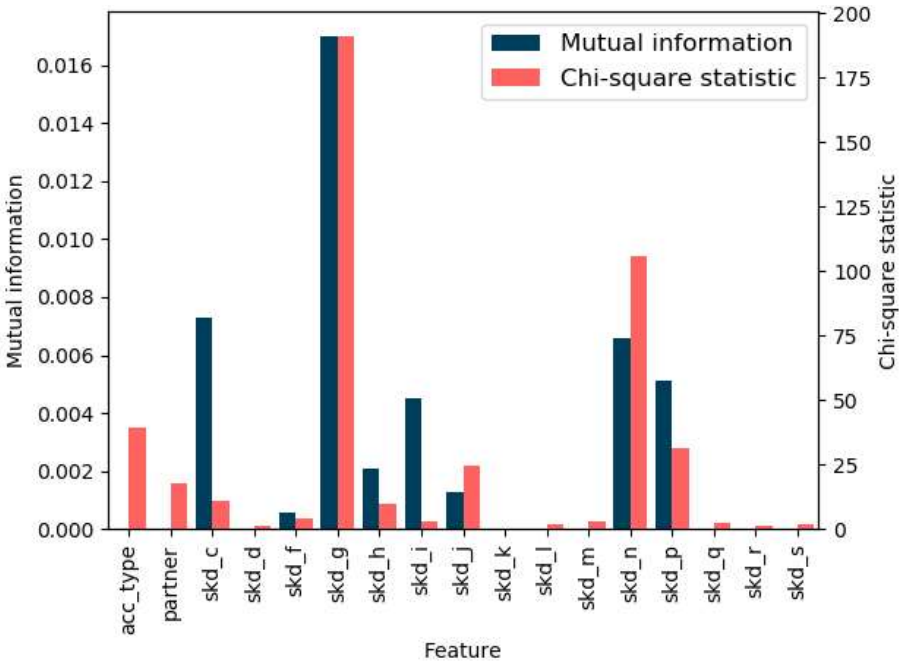
In this section, we take a look at the importance of each feature in the data set. First, we assess the importance of the qualitative features, followed by the analysis of the quantitative features.

Starting with qualitative features, we look at two standard analyses of univariate statistical methods for feature selection where the input data and output data are categorical. The first method is the chi-square test, which is a statistical hypothesis test to determine the features, which are most likely to be independent of the target variable, and therefore irrelevant for the classification. The second analysis is mutual information, which measures the amount of shared information between input features and the target variable.

Figure 12 plots the mutual information and chi-square statistic scores. Both techniques are placed in one graph to provide an easier overview. Note that they each have independent scales. In both cases, the higher the value, the better. Features *account_type* and *from_partner* are denoted as *acc_type* and *partner,* respectively. The remaining features are dummy variables of the original feature *skd_first_level*, which represent the business activity of the creditor. Interpreting the results, it is clear that both of the techniques consider some of the features more significant than others. Interestingly, the most crucial categorical feature

considered by both techniques is *skd_g*. In this particular case, *skd_g* refers to the field of 'wholesale and retail trade, repair of motor vehicles and motorcycles'.

Figure 12: Mutual information and chi-square test for categorical features



Source: own work.

Table 5: Feature p-values for chi-square test

p-value ≤ 0.05		p-value > 0.05	
Feature	**p-value**	**Feature**	**p-value**
skd_g	0.000000	skd_i	0.076513
skd_n	0.000000	skd_m	0.084273
acc_type	0.000000	skd_q	0.119998
skd_p	0.000000	skd_l	0.150380
skd_j	0.000001	skd_s	0.195274
partner	0.000025	skd_r	0.261567
skd_c	0.000942	skd_d	0.273555
skd_h	0.001825	skd_k	0.898045
skd_f	0.041961		

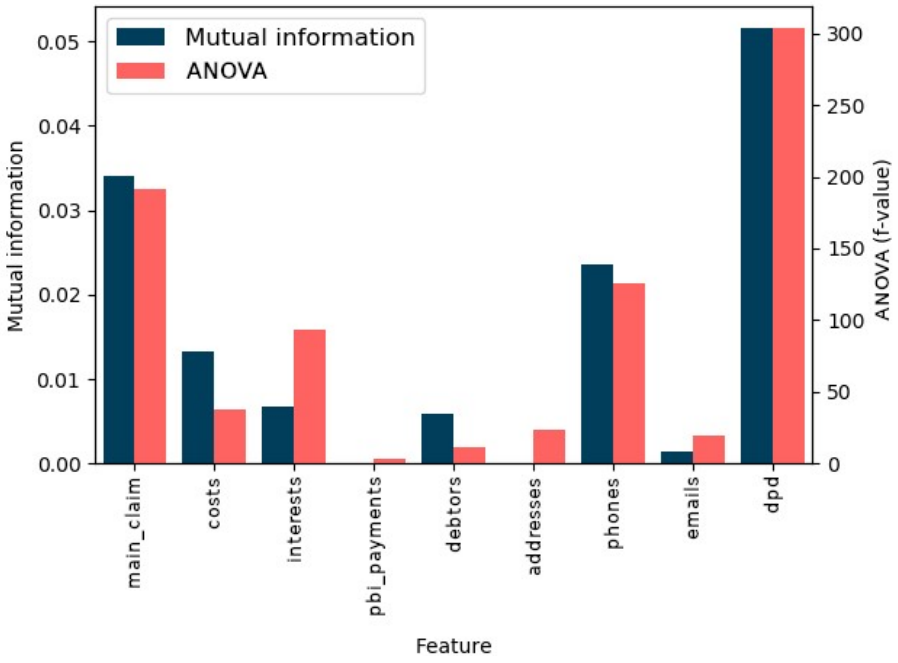
Source: own work.

For some features, the results of the two techniques contradict each other. One such feature is *acc_type*, which chi-square considers as the third most crucial feature among all categorical features. In contrast, according to the results of mutual information, this feature is not important at all. It is safe to say that a feature is irrelevant for classification if both techniques measure it as independent of the target variable. Therefore, we remove such features before continuing to the model selection. Looking at the results, we decide to discard

the following seven features: *skd_d*, *skd_k*, *skd_l*, *skd_m*, *skd_q*, *skd_r*, and *skd_s*. Table 5 shows that all of these features also have the p-values of chi-square statistic above the standard significance threshold $\alpha = 0.05$. The only feature that has a p-value over the threshold of $\alpha = 0.05$, and we still consider for classification, is *skd_i*. Despite being treated as independent by chi-square test, it is ranked as more promising by mutual information.

Proceeding to quantitative features, we again look at two tests. This time, we measure mutual information and the analysis of variance (ANOVA), which tests if the means of two or more groups significantly differ from each other. The results are shown in Figure 13. Note again that both techniques have independent scales. Mutual information is typically used with categorical or ordinal variables. According to Ross (2014), it can also be adopted and used in cases where the input features are numerical, and the target variable is categorical.

Figure 13: Mutual information and ANOVA test for numerical features



Source: own work.

As expected, both ANOVA and mutual information consider features *dpd* (debt's age) and *main_claim* as the most informative numerical features. The next best feature is *phones*, which represents the number of debtor's phones available. Features *pbi_payments*, *debtors*, *addresses,* and *emails* are by both techniques ranked as less informative when compared to others.

Considering ANOVA and mutual information results in Figure 13, as well as the features' p-values for the ANOVA in Table 6, we decide only to remove the *pbi_payments* feature. It is the only feature with a p-value above the standard significance threshold $\alpha = 0.05$.

Table 6: Feature p-values for ANOVA test

Feature	p-value
dpd	0.000000
main_claim	0.000000
phones	0.000000
interests	0.000000
costs	0.000000
addresses	0.000001
emails	0.000010
debtors	0.000658
pbi_payments	0.065932

Source: own work.

Through the step of feature selection, we identify eight features, for which we assume that are irrelevant for the task of classification, in total. This means reducing the size of the input variables from 26 to 18 features.

5.1.3 Modelling

This section is divided into two parts. First, we build different machine learning models in order to obtain the best-performing ones. Then, we optimise the best-performing models by tuning their hyperparameters.

5.1.3.1 Model selection

The goal of this step is to find the most promising classification algorithms. We do this by training ten different base classification models and evaluate their performance to find out the best-performing ones and then tune them further for even better performance in Section 5.1.3.2.

The classification algorithms used are support vector machines (SVM) with different kernels (i.e., radial basis function (rbf), polynomial (poly) and linear), decision trees, random forests, stochastic gradient descent (SGD) classifier, logistic regression, *k*-nearest neighbours (kNN), naïve Bayes, and neural networks. Each model is trained twice: once on the standardised training data and once on the normalised training data. These results determine the best-performing feature scaling technique (i.e., standardisation or normalisation).

Four standard metrics are calculated for each model: accuracy, precision, recall, and the f_1-score. Results, displayed in Table 7, are generated using stratified 5-fold cross-validation to avoid overfitting. Metrics for each of the base classification models are computed for standardised and normalised training data. Classifiers are ordered by the primary evaluation metric f_1-score on the standardised data.

Table 7: Performance evaluation of base classifiers on standardised and normalised data

	Standardised data				Normalised data			
Classifier	**Accuracy**	**Precision**	**Recall**	**F_1-score**	**Accuracy**	**Precision**	**Recall**	**F_1-score**
SVM (rbf)	0.72	0.73	0.93	0.82	0.71	0.72	0.93	0.81
neural network	0.73	0.75	0.89	0.82	0.72	0.75	0.89	0.81
SVM (linear)	0.68	0.68	0.99	0.81	0.68	0.68	1	0.81
SVM (poly)	0.71	0.72	0.94	0.81	0.7	0.71	0.95	0.81
logistic regression	0.69	0.7	0.94	0.8	0.69	0.7	0.94	0.8
SGD classifier	0.67	0.69	0.94	0.8	0.68	0.68	0.99	0.81
naïve Bayes	0.67	0.69	0.91	0.79	0.64	0.69	0.91	0.79
random forest	0.71	0.77	0.82	0.79	0.72	0.77	0.82	0.79
kNN	0.7	0.76	0.81	0.79	0.71	0.77	0.81	0.79
decision tree	0.67	0.76	0.74	0.75	0.67	0.76	0.74	0.75

Source: own work.

We observe the differences in the models' performance using the feature scaling techniques normalisation and standardisation. In our case, there are no significant differences between these two feature scaling methods. Due to a marginal increase in performance of *SVM (rbf)* and *neural network* models when using standardised data compared to normalised data, we decide for standardisation over normalisation.

Considering the results, we decide to shortlist the three best-performing models based on the f_1-metric, while also considering the recall and precision scores. We select *SVM (rbf)* as the main support vector machine model alongside with *neural network* and *logistic regression*.

Looking at the results of the model *SVM (linear),* we see that the model has the highest recall of 0.99 while having the lowest precision value (i.e., 0.68). Due to the high recall, its f_1-score is also relatively high.

Taking a look at the confusion matrix for the model *SVM (linear)* in Table 8, it becomes clear that this model mainly predicts the positive outcome for every instance. Only 1% of all examples were predicted as unsuccessful, and a third of these cases were false-negatives. This is an example of a situation described in the business case that we would like to avoid.

Table 8: Confusion matrix for the model SVM (linear)

		Predicted	
		Unsuccessful	Successful
True	Unsuccessful	86	3321
	Successful	38	7009

Source: own work.

Every model presented in Table 7, except for a *neural network,* is built using the scikit-learn library. In each case, the default settings of a model are applied to facilitate raw performance comparison. The *neural network* model is built using the Keras library, and it is a feed-forward neural network with one hidden layer. The hidden layer has thirteen neurons, according to a rule of thumb provided by Heaton (2008, p. 159), which says that "the number of hidden neurons should be 2/3 the size of the input layer, plus the size of the output layer." The model uses Adam optimiser with its default values and binary cross-entropy for calculating the loss, which is minimised during the training. The activation function of the hidden layer is a rectified linear unit (ReLU), while the output layer uses a sigmoid activation function. Both activation functions are defined with Equations (3) and (1), respectively.

5.1.3.2 Model optimisation

In this section, we optimise the three best-performing models by tuning their hyperparameters. Since we only use one model based on support vector machines, we abbreviate it as *SVM*.

A machine learning model has configuration parameters and hyperparameters. Parameters can be estimated or learned from the data and are internal to the model, while hyperparameters are external to the model and are often specified by the practitioner. Hyperparameter values are often determined by rules of thumb or can be searched for the best value by trial and error (Brownlee, 2017). We approach the optimisation of hyperparameters with the technique of trial and error to determine the best hyperparameters. For each of the three models, we choose parameters for the optimisation as follows.

For the *logistic regression* model, we consider the norm of penalisation (*penalty*), the inverse of regularisation strength (*C*), the optimisation problem algorithm (*solver*) and the maximum number of iterations for the solver to converge (*max_iter*) as the tuning hyperparameters (Scikit-learn, 2020, pp. 2071–2073).

In the case of the *SVM* model, that is, support vector machine with the rbf kernel, we select the regularisation parameter (*C*) and the kernel coefficient (*gamma*) for tuning (Scikit-learn, 2020, pp. 2571–2572). Other kernels for SVM algorithm were already tested and discarded in Section 5.1.3.1.

Regarding the *neural network* model, we tune the number of neurons in the hidden layer (*neurons*), the rate with which the weights are updated (*learning_rate*), the number of cycles the model goes through the data set while training (*epoch*), and the number of patterns shown to the model before the weights are updated (*batch_size*) (Brownlee, 2016).

Table 9 shows hyperparameter combinations that are considered for each of the models. The model optimisation is carried out using the grid-search method with the combination of stratified 5-fold cross-validation. This means that for each model, every combination of the

hyperparameters is fitted and evaluated five times on different data sets. The evaluation metric is the average f_1-score over five folds.

Table 9: Hyperparameter combinations for optimisation

Logistic regression	
penalty	l1, l2, elasticnet
C	0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000
solver	liblinear, lbfgs, newton-cg, sag, saga
max_iter	100, 1000, 2500, 5000, 10000
Total combinations	600
SVM	
C	0.001, 0.01, 0.1, 1, 10, 100, 1000
gamma	0.0001, 0.001, 0.01, 0.05, 0.1, scale, auto
Total combinations	49
Neural network	
neurons	4, 8, 12, 16, 20, 24, 28, 32, 36
learning_rate	0.001, 0.01, 0.1, 0.2, 0.3
epoch	50, 75, 100, 125
batch_size	10, 20, 40, 60, 80, 100, 120
Total combinations	1260

Source: own work.

The hyperparameter results for the grid-search are as follows:

- logistic regression: *penalty* = l2, *C* = 0.001, *solver* = lbfgs, *max_iter* = 100;
- SVM: *C* = 10, *gamma* = auto; and
- neural network: *neurons* = 20, *learning_rate* = 0.01, *epoch* = 100, *batch_size* = 40.

Table 10: Performance cross-validation on training data after optimisation

Model	**Accuracy**	**Precision**	**Recall**	**F_1-score**
SVM	0.73	0.74	0.93	0.82
neural network	0.73	0.75	0.89	0.82
logistic regression	0.68	0.68	0.98	0.81

Source: own work.

Table 10 shows the performance of each model trained with the best hyperparameter values. The models are evaluated on the training set using stratified 5-fold cross-validation. The results indicate that the search for the best hyperparameters did not result in a significant performance increase (compare with Table 7). Half of the metric values remain the same. The precision of the *SVM* model slightly increased, the performance of the *neural network* model stayed the same, and recall of the *logistic regression* model increased at the expense

of lower precision. All models except the logistic regression retain the same evaluation performance value of the f_1-score metric.

5.2 The second set of models – one month into the debt collection process

In this section, we focus on building models that predict the debt collection outcome one month into the debt collection process.

5.2.1 Data understanding

The second set of models evaluates the performance of the debt collection one month after the debt collection process has started. The data set used in this case is the extension of the one that is used in the first set of models. It consists of 27 features and 13,068 examples. Thirteen features are the same as in the first set of models. The additional fourteen features shown in Table 11 are limited to activities performed within the first month since the start of the debt collection process. Features covered in the first set of models are not interpreted again.

Table 11: Feature description of additional features

Feature	Type	Description
returned_mail	binary	Does the sent mail return (1 - yes / 0 - no)?
skip_tracing	binary	Is the skip tracing performed (1 - yes / 0 - no)?
outgoing_action	discrete	The number of outgoing actions (SMS, fax, email, special letter).
incoming_action	discrete	The number of incoming actions (SMS, fax, email, special letter, visits).
outgoing_call	discrete	The number of outgoing calls made.
incoming_call	discrete	The number of incoming calls received.
payments	continuous	Sum of payments made.
payment_plan	binary	Did the debtor agree to a payment plan (1 - yes / 0 - no)?
plan_pay_ratio	continuous	The ratio of payments made and payments promised in the payment plan.
bankruptcy	binary	Did the debtor declare bankruptcy (1 - yes / 0 - no)?
letters_s1	discrete	The number of letters of the first seriousness degree sent.
letters_s2	discrete	The number of letters of the second seriousness degree sent.
letters_s3	discrete	The number of letters of the third seriousness degree sent.
letters_s4	discrete	The number of letters of the fourth seriousness degree sent.

Source: debt collection company.

The whole data set consists of six binary, two categorical, thirteen discrete, and six continuous features. As in the first set, the target variable is the categorical feature *outcome* that contains the result of the debt collection process (successful or unsuccessful). First, we describe the binary features and then numerical features in more detail.

The binary feature *returned_mail* tracks whether the letter sent reaches the debtor. If at least one letter returns, the value of the feature is 1. The ability to reach the debtor by mail is essential for a successful outcome. In our data set, the company experienced returning mail in 5% of all cases. The set of possible activities to perform at the start of the debt collection process depends on the data handed over by the creditor. For example, if the creditor does not provide the debtor's phone number, the debt collection company cannot reach the debtor over the phone. From the description of the variables used in the previous set of models (see Section 5.1.1), we know that the phone number is available in just over half of the cases at the beginning of the debt collection. In the case of missing contact information, the company has to perform skip tracing, which is the search for the debtor's contact information through inquiries by government agencies and other accessible databases. The frequency of skip tracing performed is the highest at the start of the collection process. This is reflected by the *skip_tracing* feature, which indicates that almost half of the cases were subject of skip tracing.

The *payment_plan* feature contains information on whether the debtor agreed for a payment plan. A payment plan is an agreement between the debt collection company and the debtor to repay the debt in instalments. In over a third of all cases, an agreement for a payment plan was reached. The *bankruptcy* feature contains information on whether the debtor declared bankruptcy. In the event of debtor's bankruptcy, the collection process has to be suspended, and the chances of repayment are reduced. In only 64 instances of the data set, the debtor declared bankruptcy.

Table 12: Numerical feature information for the additional features

Feature	count	mean	std	min	25%	50%	75%	max
outgoing_action	13,068	0.36	0.74	0	0	0	1	11
incoming_action	13,068	0.29	0.73	0	0	0	0	15
outgoing_call	13,068	0.75	1	0	0	0	1	9
incoming_call	13,068	0.38	0.73	0	0	0	1	10
payments	13,068	77.37	284.24	0	0	6.46	59.92	9,637.21
plan_pay_ratio	13,068	0.18	0.38	0	0	0	0	1
letters_s1	13,068	0.87	0.44	0	1	1	1	1
letters_s2	13,068	0.03	0.17	0	0	0	0	2
letters_s3	13,068	0.08	0.27	0	0	0	0	2
letters_s4	13,068	0.001	0.03	0	0	0	0	2

Source: own work.

Table 12 describes the numerical features of the second set of models in more detail. No feature contains any missing values (see column *count*). Features *outgoing_action* and *incoming_action* count the number of actions based on the direction of the activity. The *outgoing_action* feature covers activities performed by the debt collection company, while actions initiated by the debtor are included in the *incoming_action* feature. There are slightly

more outgoing than ingoing actions made per case. Calls are collected as a separate feature (and are not included in general actions) since they are the most common activity performed by debt collection companies. We again distinguish between the direction of the call made. Calls performed by the debt collection company are included in the *outgoing_call* feature, while the calls made by the debtor are counted in the *incoming_call* feature. Calls, where the connection was not established, are not included. Looking at the ratio, we notice that for every incoming call received, there are almost two outgoing calls made.

The *payments* feature represents the total amount of payments made by the debtor in the selected period; the average sum of payments in the first month of the collection is 77€, which represents about 22% of the average main claim. Feature *plan_pay_ratio* is linked to the binary feature *payment_plan* and represents the ratio of agreed payments the debtor has settled. Looking at the cases where a payment agreement was reached, it becomes clear that many debtors do not keep their promise as they only pay a little over half of the agreed sum.

Last but not least, features *letters_s1*, *letters_s2*, *letters_s3*, *letters_s4* all include the number of letters sent according to the severity level of the letter. The suffixes *s1* and *s4* denote the letters with the minimum and maximum degree of severity, respectively. As seen in Table 12, letters with the first degree of severity are the most common ones and represent almost 90% of all letters sent in the first month of the debt collection process.

5.2.2 Data preparation

This section describes the steps of data cleaning, feature scaling, and feature selection for the second set of models.

5.2.2.1 Data cleaning

The additional features do not represent any further work regarding the cleaning of the data. Therefore, we perform the same procedure as when cleaning the data for the previous set of models (see Section 5.1.2.1). The process consists of encoding features *account_type* and *skd_first_level*, after which the data set contains 41 features. One of the features, that is, *outcome*, represents a target variable, while the rest are input variables. To keep the same examples in both sets of models, we do not remove any additional examples that represent outliers.

The data set is again split into the training set and the test set with the ratio of 80:20, stratified according to the target variable *outcome*. All models will be trained and evaluated on the training set using 5-fold cross-validation until final versions of models are selected. Only then, the models will be evaluated on the test set.

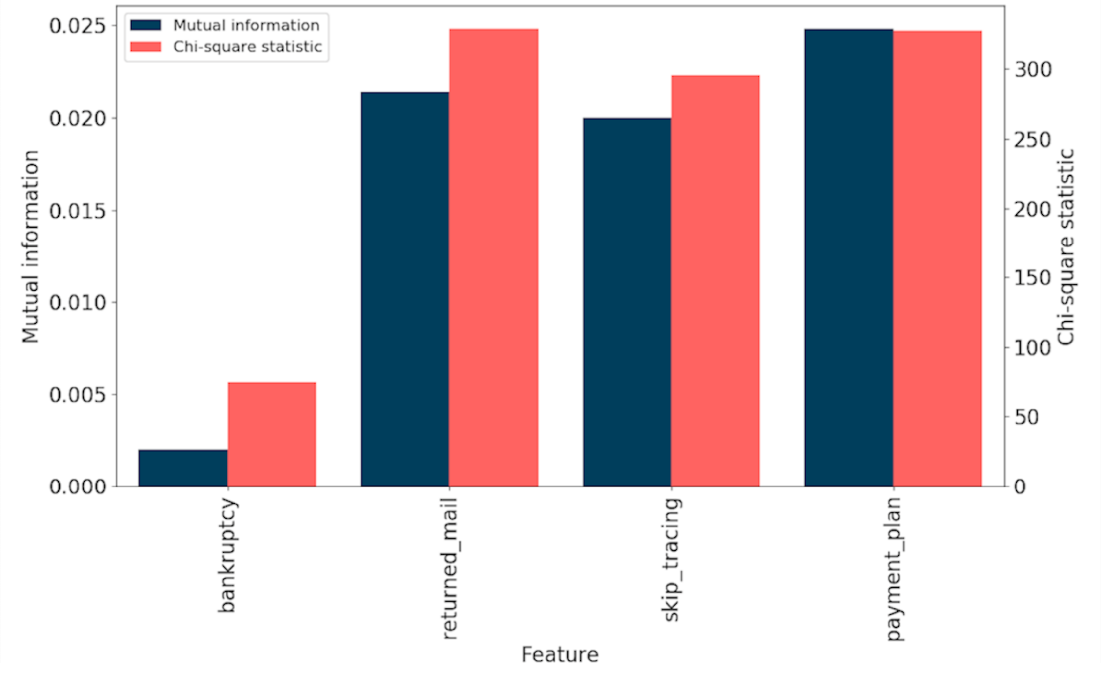
5.2.2.2 Feature scaling

Based on the results in the model selection step in the previous set of models (see Section 5.1.3.1), where there were no significant differences between the use of standardisation and normalisation as the scaling techniques, we decide to continue using only standardisation.

5.2.2.3 Feature selection

In this section, we look at the importance of the additional features for the second set of models. Because chi-square statistic, mutual information, and ANOVA are all univariate statistical tests, there is no need to test features that have already been tested. Therefore, for features that appear in the first set of models, we take into account the results from the previous feature selection analysis found in Section 5.1.2.3. For the rest of the features, tests are performed in two parts. The first part covers the categorical (i.e., binary) features using a combination of the chi-square statistical test and mutual information. The numerical features are tested in the second part using a combination of the ANOVA test and mutual information. In both Figure 14 and Figure 15, the scales used in plots are independent of each other.

Figure 14: Mutual information and chi-square test for categorical features



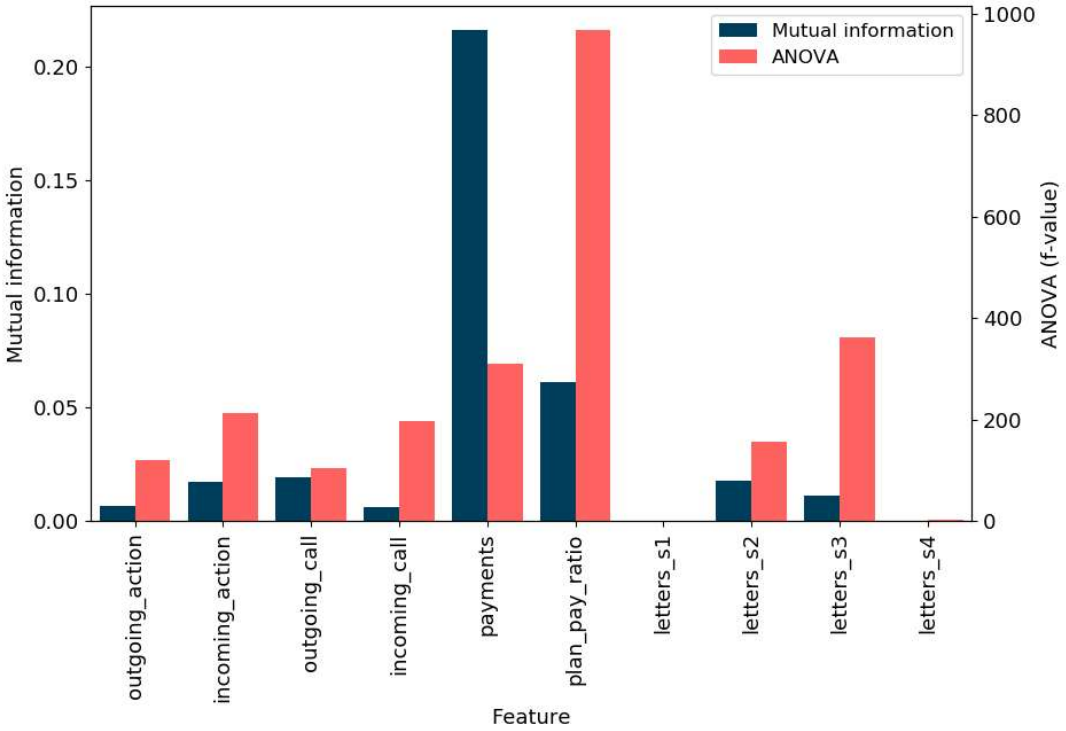
Source: own work.

Results in Figure 14 suggest that features *returned_mail*, *skip_tracing* and *payment_plan* are all important to a similar degree by both chi-square statistic and mutual information. Compared to other features, *bankruptcy* seems to be the least relevant for the classification

performance. Nevertheless, we decide to keep all features because they all have a p-value of the chi-square statistic below the standard significance threshold $\alpha = 0.05$.

Results of the ANOVA and mutual information analyses are displayed in Figure 15. Unsurprisingly, the continuous features reflecting the payment behaviour of the debtor (i.e., *payments* and *plan_pay_ratio*) are considered the most informative. Interestingly, each method has its own opinion as to which of these two features is more important. Features on the left side of the bar chart reflecting activities and calls are considered to be somewhat important. On the other hand, features regarding the second and third severity degree of letters are more important than the first- and the fourth-degree letters.

Figure 15: Mutual information and ANOVA test for numerical features



Source: own work.

Only the *letters_s1* feature has the p-value of the ANOVA analysis above the threshold value $\alpha = 0.05$, while the p-value of the *letters_s4* feature is on the threshold border. We decide to discard both features, that is, *letters_s1* and *letters_s4*. Additional motivation to discard the *letters_s4* feature is the fact that there are only eleven observations of letters with the fourth degree of severity being sent in the whole data set. After the feature selection step, we end up with a data set consisting of 30 features excluding the target variable.

5.2.3 Modelling

This section is divided into two parts. First, we build different machine learning models in order to obtain the best-performing ones. Then, we optimise the best-performing models by tuning their hyperparameters.

5.2.3.1 Model selection

After the addition of new features, the list of the best-performing models drastically changed. The results of the model selection analysis are shown in Table 13. There has been a significant boost in the performance of all models. The new best-performing model is *random forest*, which was not considered as a top-performing model in the first set of models. Interestingly, the *decision tree* model performs much better than before and is among the better models. Models *decision tree* and *random forest* especially benefited from the additional features.

Table 13: Performance evaluation of base classifiers

Classifier	Accuracy	Precision	Recall	F_1-score
random forest	0.83	0.89	0.86	0.88
neural network	0.81	0.86	0.86	0.86
SVM (rbf)	0.8	0.82	0.89	0.86
decision tree	0.8	0.85	0.85	0.85
logistic regression	0.78	0.82	0.87	0.84
SVM (linear)	0.77	0.8	0.88	0.84
SVM (poly)	0.77	0.78	0.92	0.84
SGD classifier	0.76	0.81	0.84	0.83
kNN	0.77	0.83	0.84	0.83
naïve Bayes	0.73	0.75	0.92	0.82

Source: own work.

Based on the obtained results, we eliminate *logistic regression* from the set of models for optimisation. For further analysis, we select the three best-performing models, which are *random forest, neural network,* and *SVM (rbf)*.

5.2.3.2 Model optimisation

In this section, we optimise the three best-performing models by tuning their hyperparameters. Because we only use one model based on support vector machines, we denote it as *SVM*.

The domains of hyperparameters for models *SVM* and *neural network* are the same as in the first set of models. The only difference is in the number of neurons in the hidden layer for

the *neural network* model. The hyperparameter domain of *neurons* is adjusted to [4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60], reflecting the rule of thumb set by Heaton (2008): "the number of hidden neurons should be less than twice the size of the input layer." The total number of combinations tested for the *neural network* model is 2,100.

Table 14 shows the optimisation domain of hyperparameters for the *random forest* model. Hyperparameters considered for optimisation are the number of trees in the forest (*n_estimators*), the number of features to consider when looking for the best split (*max_features*), the maximum depth of the tree (*max_depth*), the minimum number of samples required to split an internal node (*min_samples_split*), the minimum number of samples required to be at a leaf node (*min_samples_leaf*), and the function to measure the quality of a split (*criterion*) (Scikit-learn, 2020, pp. 546–548).

Table 14: Random forest hyperparameter combinations for optimisation

Random forest	
n_estimators	100, 250, 400, 550, 700, 850, 1000, 1150, 1300, 1450
max_features	auto, log2
max_depth	5, 20, 35, 50, 65, 80, 95, None
min_samples_split	2, 5, 10, 20
min_samples_leaf	1, 2, 5, 10
criterion	gini, entropy
Total combinations	5120

Source: own work.

The results of the grid-search for the best hyperparameters for each model are as follows:

- random forest: *n_estimators* = 1300, *max_features* = auto, *max_depth* = 20, *min_samples_split* = 2, *min_samples leaf* = 2, *criterion* = entropy;
- SVM: *C* = 10, *gamma* = auto; and
- neural network: *neurons* = 44, *learning_rate* = 0.001, *epochs* = 100, *batch_size* = 20.

Table 15: Performance cross-validation on training data after optimisation

Model	**Accuracy**	**Precision**	**Recall**	**F_1-score**
random forest	0.84	0.91	0.85	0.88
neural network	0.82	0.87	0.86	0.86
SVM	0.8	0.84	0.88	0.86

Source: own work.

Table 15 shows the performance of the models with the best hyperparameters learned during optimisation. The evaluation is performed on the training set using stratified 5-fold cross-validation. The results reveal a slight improvement in accuracy and precision metrics

compared to results in Table 13. The main f_1-score metric stayed the same, while the balance of precision and recall has slightly shifted. The increase in precision and decrease in the recall is notable with almost all models.

6 MODEL EVALUATION AND COMPARISON

In this chapter, we evaluate the performance of the models from both sets on the test data. The models in question are the best-performing models selected and optimised in Chapter 5. Each model is trained on the whole training set with the best hyperparameters found during optimisation and then evaluated on the test set. For each model, we calculate the following metrics: accuracy, precision, recall, and f_1-score. The latter is also the primary evaluation metric. Additionally, we display the model performance with the precision-recall curve and compute the AUC score, which is the integral of the precision-recall curve.

The first set of models predicts the debt collection outcome at the beginning of the debt collection process, where the features are limited. The top-performing models for this problem are based on support vector machine with rbf kernel, neural network and logistic regression. The second set of models predicts the debt collection outcome one month into the process. The best-performing models are based on support vector machine with rbf kernel, neural network and random forest.

Table 16 shows the prediction performance for the first set of models, predicting the outcome at the beginning of the debt collection process. The metrics indicate that the performance of the models on the test set is comparable to the performance on the training set (see Table 10). The *SVM* model has the same metric values on the test and the training set. The precision of the *neural network* model increased, while its recall decreased. Surprisingly, the results suggest that the performance of the *logistic regression* model is slightly better on the test set than the training set.

Table 16: Performance evaluation of the first set of models on test data

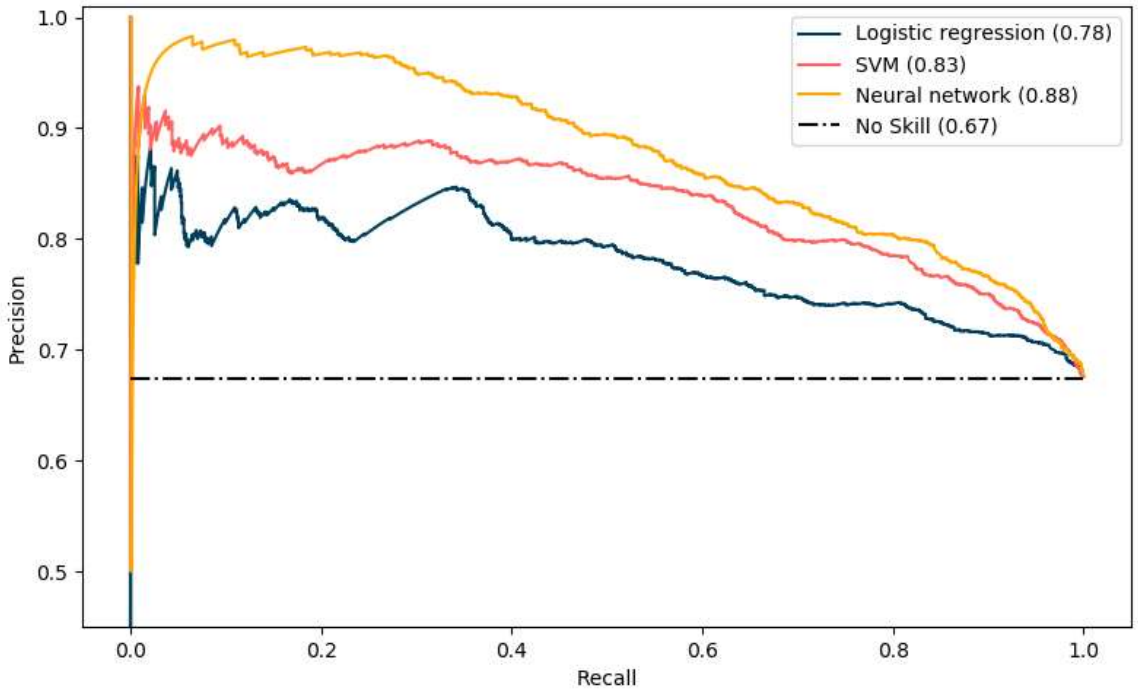
Model	Accuracy	Precision	Recall	F_1-score
SVM	0.73	0.74	0.93	0.82
neural network	0.74	0.77	0.88	0.82
logistic regression	0.69	0.69	0.99	0.81

Source: own work.

The precision-recall curve displayed in Figure 16 is a plot of the recall on the x-axis and precision on the y-axis where each line represents one model. The dash-dotted line represents a model with no skill, which is a model that always predicts the majority class, that is, the successful debt collection. The score next to the name of the model in the legend is the AUC score. The higher the AUC score, the better.

We see that it is harder for the models to have a high precision value than the recall value. This is expected because it is easier for a model to predict the positive class since there are twice as many positive outcomes in the data set as negative. The more a model predicts the positive class, the higher the recall and the lower the precision until it reaches the precision of the no skill model (i.e., 0.67). Ideally, we would like for a model to have a line as close to the right upper corner as possible. We see that the *neural network* model outperforms the other two at any threshold. Therefore, the *neural network* model would be the best in this scenario. The exact precision and recall performance value of the model is dependent on the threshold value selected. Determining the best threshold value depends on the precision-recall trade-off and which of these metrics is preferred in a specific business case.

Figure 16: Precision-recall curves of models in the first set



Source: own work.

Table 17: Performance evaluation of the second set of models on test data

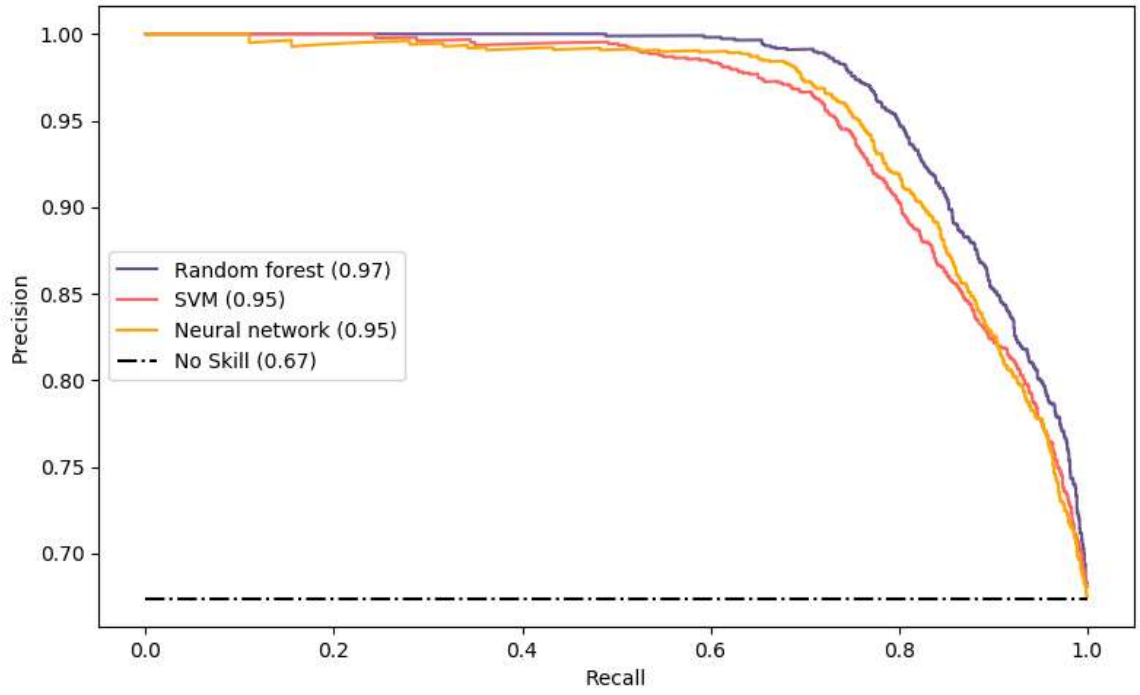
Model	Accuracy	Precision	Recall	F_1-score
random forest	0.84	0.91	0.85	0.88
neural network	0.81	0.84	0.89	0.86
SVM	0.8	0.84	0.88	0.86

Source: own work.

Performance results for the second set of models predicting the outcome of the collection process one month after the start are shown in Table 17. As it was the case with the first set of models, these results are also comparable to the ones on the training set (see Table 15).

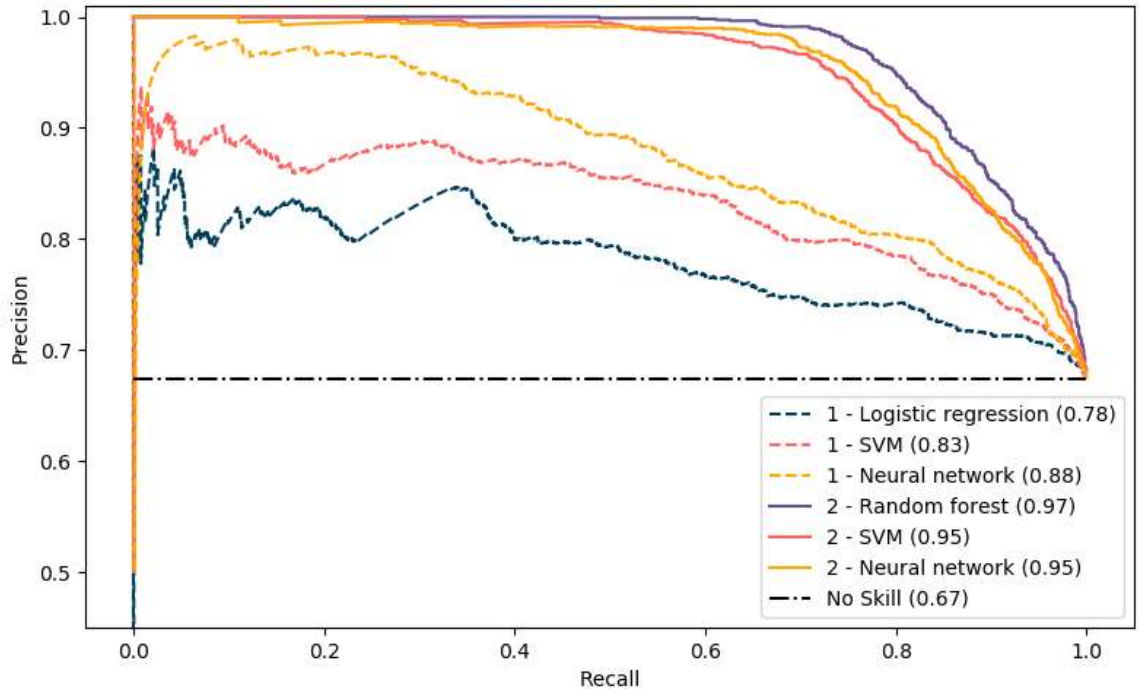
There is a slight difference in the precision and recall values for the *neural network* model, where the precision decreased and the recall increased.

Figure 17: Precision-recall curves of models in the second set



Source: own work.

Figure 18: Precision-recall curves of all models



Source: own work.

To determine the best-performing model, we look at the precision-recall curve shown in Figure 17. The *random forest* model outperforms the other two models noticeably. Models *SVM* and *neural network* are about equally effective in predictions, with the same AUC score.

In order to compare the prediction performance at the beginning of the debt collection process with the prediction performance after a month, Figure 18 combines the precision-recall curves for both sets of models.

The performance of models predicting the outcome after a month of debt collection is superior to the performance of models predicting the outcome at the start of the debt collection process. The additional features used by the second set of models have a substantial impact on the models' performance. The results of the second set of models prove a relatively good prediction performance, while the obtained results for the first set of models raise concerns about the sensibility of predicting debt collection outcome at the beginning of the debt collection process.

The poor performance of models in the first set can be a sign of high bias (i.e., underfitting). As discussed earlier, underfitting is the inability of a model to predict the labels correctly. In our case, the underfitting possibly originates from the lack of the prediction power of the features used. In order to solve this issue, it is crucial to use features with more predictive power.

7 DISCUSSION

Obtained results reveal a significant gap in the prediction performance between the models in the first set predicting the outcome of the debt collection at the start of the process compared to the models in the second set predicting the outcome a month into the debt collection process.

Models in the first set struggle at predicting the debt collection outcome at the beginning of the collection process. Although the models have relatively high f_1-score values, this is mainly due to very high recall values, which are easier for models to achieve (especially noticeable in the case of the final *logistic regression* model). This shortcoming is primarily due to the inability to predict the minority class correctly, which is captured by the low precision values. We believe that this shortcoming stems from the lack of relevant features, which causes the models to underfit.

On the contrary, the second set of models is predicting relatively well, which is noticeable in the higher precision values resulting in higher f_1-scores. The *random forest* model is the only model that has a higher precision value than recall value, which makes it the most promising model to predict the debt outcome one month after the debt collection process has started since it is the best model for predicting the minority class.

At the beginning of the thesis, we defined four research goals. The first goal was to review the relevant literature on the use of machine learning in the area of debt collection. There exist many studies that combine these two fields. The purpose of machine learning application varies among the studies. The main focal point is the optimisation of the debt collection process through the application of machine learning algorithms on different segments of the process, for example, determining the optimal debt pursuit duration, calculating the probability of repayment, or tailoring the debt collection process through predicting the next best activity to maximise the collection. Furthermore, the literature review revealed that some studies also encountered the problem of lacking relevant features for debt classification. One such study of bad debt prediction in the healthcare industry was carried out by the University of Lousiville.

Within the scope of the second goal, we researched whether the use of machine learning algorithms is reasonable and meaningful in the context of the business case. The results obtained in the empirical part of the research indicate that the use of machine learning algorithms is reasonable and meaningful for the prediction of the debt collection process. However, there must be enough relevant features available for the models to predict the debt collection outcome successfully. The models in the first set have performance difficulties, especially in predicting the minority class. All models of the first set suffer from underfitting, which is a consequence of the lack of relevant features available. Nevertheless, the models of the second set perform well and are relatively successful in predicting the debt collection outcome. These models offer a promising starting point for further research.

The third research goal was to derive the key opportunities and challenges through the implementation of machine learning algorithms within the business case. The main challenge of the business case is to successfully predict the collection outcome at the beginning of the debt collection process. Currently, with the data provided by the creditors at the start of the debt collection process, it is not possible to reliably predict the debt collection outcome at that time. This issue could be resolved by including more relevant features. However, this is easier said than done. At the start of the debt collection process, the data about the debt and the debtor is scarce. An attempt to improve the prediction performance would be to include the data that is already available, for example, whether the debtor is already recorded in the company's database, and use data related to the collection processes found. Nonetheless, this would be only beneficial for the cases where the debtor has multiple debts, and the company possesses information about them. The only other way to improve the prediction performance of models in the first set would be to include more examples in the data set. The second set of models present an opportunity for a successful debt collection outcome prediction. The additional features included in these models provided the necessary prediction power needed for better prediction performance. However, the trade-off of these models is that the additional features are not available immediately at the start of the collection process. Therefore, the time of the debt outcome

prediction is placed after the start of the debt collection process, that is, when the data becomes available.

The last research goal was to describe how to deal with the implementation of machine learning algorithms, to identify the most successful one for the prediction of debt collection performance, and to identify the key success factors of the implementation. The detailed description of how to deal with the implementation of a machine learning algorithm is quite extensive and can be found in Chapter 5.

The best-performing model from the first set is the *neural network* model. While it has the same f_1-score as the final *SVM* model, it outperforms other models in the prediction of the minority class, which is also reflected in the highest AUC score of 0.88. Even though the *neural network* model has the best results from the first set of models, we do not believe that any model from the first set performs well enough to use them in practice. On the other hand, models from the second set are performing significantly better than the models from the first set. The best-performing model in the second set is *random forest* with the AUC score of 0.97. It outperforms both *SVM* and *neural network* models. It has the highest precision, which means that it is the most successful in predicting the minority class.

We identify two key success factors that were crucial for the successful implementation of machine learning algorithms for predicting debt collection outcome. We believe that the most crucial factor is to understand the data and to keep the business objective in mind. The most important decision regarding both the data and the business objective is the selection of the evaluation metric. Based on the data, the decision is affected by the distribution of the target variable. The business objective also affects the decision based on the purpose of the implementation. Due to the imbalanced distribution of the target variable, we rely on metrics precision and recall instead of the accuracy. A high precision minimises the number of false-positive predictions, while a high recall minimises the number of false-negative predictions. In our case, a false-positive prediction is cheaper for the company than a false-negative prediction. The latter would mean that the company abandons debt collection activities or writes off a collectable debt and thereby loses the whole debt amount. On the other hand, a false-positive prediction means that the company continues to perform collection activities on a case that is probably not worthwhile pursuing, which results in higher operational costs. However, the gravitation towards high recall without high precision can be a double-edged sword. A model can achieve high recall by only predicting the majority class, which makes a model useless. To strive for both high precision and high recall, we use the f_1-score, which is the harmonic mean of both of the two metrics.

The second key success factor is the use of a structured approach in order to plan the implementation of machine learning algorithms, such as the CRISP-DM methodology. It provides an overview of the entire process of building machine learning models, from business understanding to the model deployment. Moreover, it provides detailed information on each step of the implementation of the algorithms. Therefore, by using this methodology,

many issues can be caught early on, and the scenario of the final model severely underperforming can be prevented.

The debt collection company that provided the data is satisfied with the results of the research. From their point of view, this was a pilot study to explore the possibility of using machine learning to enhance the debt collection process by predicting the debt collection outcome. The results of the first set of models confirmed their assumption that the available data at the start of the debt collection lacks the predictive power to predict the debt collection performance successfully. The results of the second set of models, especially the *random forest* model, seem promising to the company. Currently, the insight gained can be used by the agents as additional information in deciding whether to abandon the debt collection process in individual cases. The company is convinced that the outcome prediction during the process of debt collection itself proved to be effective, that the obtained results are promising, and that this research represents a good starting point for further research on this topic. They believe that with even more features and a more extensive data set, they can further improve the prediction performance to the extent that the model will be able to independently and reliably predict the outcome of the debt collection process.

CONCLUSION

This research aimed to identify opportunities and challenges of machine learning implementation to predict debt collection performance. Outstanding debts pose a risk to the companies that own the debt since it lowers their revenue. The ability to predict the outcome of a case allows for the selective allocation of resources to cases that are considered to have a successful outcome. Therefore, operational costs are saved on cases where the outcome is considered to be unsuccessful. Alternatively, the company can write such debts off.

To provide a comprehensive overview of the background, we first described the field of machine learning and the process of debt collection. The theoretical part is concluded with a review of the application of machine learning algorithms in the area of debt collection.

The empirical part of the thesis is based on a business case provided by a debt collection company from Slovenia. The business case aims to determine the reasonableness of the use of machine learning algorithms to predict the debt collection outcome. Unlike a financial company that offers loans, a debt collection company has little information about the debtor at their disposal. To address the potential lack of relevant features for a successful model prediction at the beginning of the collection process, we decided to perform the prediction in two points in time of the debt collection process. The first prediction time point is at the beginning of the debt collection process, where only the data provided by the creditor is available. The second prediction time point is one month after the debt collection process has started. At the time of the second prediction, additional data is available, which is the product of actions taken in the debt collection process so far.

The results obtained in the empirical part of the research reveal a significant difference in the models' prediction performance between the two time points. Models predicting at the first time point struggle to successfully predict the debt collection outcome and show signs of underfitting. Due to these issues, the reasonableness of the prediction at the first time point is questionable. On the contrary, models predicting at the second time point are performing much better. The *random forest* model proved to be the most successful at the debt outcome prediction with an f_1-score of 0.88.

All set goals of the research have been achieved. The main contribution of the master's thesis is the step-by-step guide on how to successfully implement machine learning algorithms for predicting the debt collection outcome. The results confirm that it is possible to successfully predict the debt collection outcome using machine learning even in the case of the lack of relevant features at the very beginning of the collection process. By shifting the prediction point from the beginning of the debt collection process to a point during the process itself, it is possible to compensate for the lack of features and significantly improve the prediction performance.

One of the constraints of the research is a rather small data set with features that have limited prediction power resulting in performance issues, especially in the first set of models. Also, as the legal process of debt collection drastically differs from the prelegal one primarily due to slow judicial processes and their complexity, the former was excluded from the research.

Further research on this topic could include the exploration of the most appropriate time for debt outcome prediction, as the second time point at one month into the process of debt collection was set arbitrarily. Possibly, there exists a better time point for the prediction. Furthermore, a series of models can be built, which would predict the outcome at various stages of the debt collection process to assess whether the debt is still worthwhile pursuing.

REFERENCE LIST

1. Abe, N., Kowalczyk, M., Domick, M., Gardinier, T., Melville, P., Pendus, C., … Cooley, B. R. (2010). Optimizing debt collections using constrained reinforcement learning. *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM Press.
2. Banko, M., & Brill, E. (2001). Scaling to Very Very Large Corpora for Natural Language Disambiguation. *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, 26–33. Stroudsburg, PA, USA: Association for Computational Linguistics.
3. Branco, P., Torgo, L., & Ribeiro, R. P. (2015). A Survey of Predictive Modelling under Imbalanced Distributions. *CoRR*, *abs/1505.0*.
4. Brownlee, J. (2016). *How to Grid Search Hyperparameters for Deep Learning Models in Python With Keras*. Retrieved August 2, 2020, from

https://machinelearningmastery.com/grid-search-hyperparameters-deep-learning-models-python-keras/
5. Brownlee, J. (2017). *What is the Difference Between a Parameter and a Hyperparameter?* Retrieved July 31, 2020, from https://machinelearningmastery.com/difference-between-a-parameter-and-a-hyperparameter/
6. Brownlee, J. (2020). *How to Perform Feature Selection With Numerical Input Data*. Retrieved August 4, 2020, from https://machinelearningmastery.com/feature-selection-with-numerical-input-data/
7. Brynjolfsson, E., & McAfee, A. (2017). The Business of Artificial Intelligence. *Harvard Business Review*, 20.
8. Burkov, A. (2019). *The Hundred-Page Machine Learning Book* (1st ed.). Kindle Direct Publishing.
9. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). CRISP-DM 1.0 Step-by-step data mining guide. In *SPSS inc* (Vol. 78).
10. Ciobanu, D., & Vasilescu, M. (2013). Advantages and Disadvantages of Using Neural Networks for Predictions. *Ovidius University Annals: Economic Sciences Series*, *13*(1), 444–449.
11. Desai, V. S., Crook, J. N., & Overstreet, G. A. (1996). A comparison of neural networks and linear scoring models in the credit union environment. *European Journal of Operational Research*, *95*(1), 24–37.
12. Dinh, A.-T. (2019). Random Forest. Retrieved August 2, 2020, from https://dinhanhthi.com/random-forest
13. Eurostat. (n.d.). *Private sector debt, consolidated - % of GDP*. Retrieved March 29, 2020, from https://ec.europa.eu/eurostat/databrowser/view/tipspd20/default/table?lang=en
14. Fay, B. (n.d.). *What is Debt Recovery?* Retrieved March 26, 2020, from https://www.debt.org/advice/recovery/
15. Géron, A. (2017). *Hands-on machine learning with Scikit-Learn and TensorFlow concepts, tools, and techniques to build intelligent systems*. Sebastopol, CA: O'Reilly Media.
16. Guyon, I., & Elisseeff, A. (2003). An Introduction of Variable and Feature Selection. *Journal of Machine Learning Research*, *3*, 1157–1182.
17. Hao, K. (2019). *Google's AI can now translate your speech while keeping your voice*. Retrieved July 12, 2019, from MIT Technology Review website: https://www.technologyreview.com/s/613559/google-ai-language-translation/
18. Hastie, T., James, G. M., Tibshirani, R., & Witten, D. (2017). *An introduction to statistical learning: with applications in R*. Springer : Springer Science+Business Media.
19. Heaton, J. (2008). *Introduction to Neural Networks with Java* (2nd ed.). Heaton Research, Inc.
20. Horvat, M., & Guzej, N. (2010). *Izvršba z verodostojno listino v teoriji in praksi* (1st ed.). Maribor: De Vesta.

21. Huff, D., & Geis, I. (1993). *How to lie with statistics*. New York; London: W.W. Norton & Co.
22. International Monetary Fund. (2019). *Global Financial Stability Report*.
23. Intrum. (2019a). *European Consumer Payment Report 2019*.
24. Intrum. (2019b). *European Payment Report 2019*.
25. John, G. H., Kohavi, R., & Pfleger, K. (1994). Irrelevant Features and the Subset Selection Problem. *Proceedings of the Eleventh International Conference on International Conference on Machine Learning*, 121–129. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
26. Jović, A., Brkić, K., & Bogunović, N. (2015). A review of feature selection methods with applications. *38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 1200–1205.
27. Keras. (n.d.-a). *About Keras*. Retrieved July 25, 2020, from https://keras.io/about/
28. Keras. (n.d.-b). *Why choose Keras?* Retrieved July 25, 2020, from https://keras.io/why_keras/
29. Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling*. New York, NY: Springer.
30. Lajevec, M. (2019). *Praktični seminar: Pravila izvršbe v praksi*. Forum Akademija.
31. Lee, P., Stewart, D., & Calugar-Pop, C. (2018). *Technology, Media and Telecommunications Predictions 2018*.
32. McFadden, C. (2019). *Neural Networks Are Being Used to Help Predict Road Traffic More Accurately*. Retrieved July 12, 2019, from http://interestingengineering.com/neural-networks-are-being-used-to-help-predict-road-traffic-more-accurately
33. Mejia, N. (2019). *AI-Based Fraud Detection in Banking – Current Applications and Trends*. Retrieved July 12, 2019, from https://emerj.com/ai-sector-overviews/artificial-intelligence-fraud-banking/
34. Miller, G., Weatherwax, M., Gardinier, T., Abe, N., Melville, P., Pendus, C., … Cooley, B. R. (2012). Tax Collections Optimization for New York State. *Interfaces*, *42*, 74–84.
35. Mitchell, T. M. (1997). *Machine Learning* (1st ed.). New York, NY, USA: McGraw-Hill, Inc.
36. Mitchner, M., & Peterson, R. P. (1957). An Operations-Research Study of the Collection of Defaulted Loans. *Operations Research*, *5*(4).
37. Možina, M., Demšar, J., Kattan, M., & Zupan, B. (2004). Nomograms for Visualization of Naive Bayesian Classifier. In J.-F. Boulicaut, F. Esposito, F. Giannotti, & D. Pedreschi (Eds.), *Knowledge Discovery in Databases: PKDD 2004* (pp. 337–348). Berlin, Heidelberg: Springer Berlin Heidelberg.
38. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.
39. Prek, M., & Rems, M. (1999). *Učinkovita izterjava dolgov*. Ljubljana: Primath.
40. Qingchen, W., Geer, R. van de, & Bhulai, S. (2018). *Data Driven Debt Collection Using Machine Learning and Predictive Analytics*. Retrieved June 9, 2019, from

https://www.datascience.com/blog/data-driven-debt-collection-machine-learning-predictive-analytics
41. Rinaldi, L., & Sanchis-Arellano, A. (2006). *Household Debt Sustainability What Explains Household non-performing loans? An empirical analysis* (No. 570).
42. Ross, B. (2014). Mutual Information between Discrete and Continuous Data Sets. *PloS One*, *9*, e87357.
43. Russell, S., & Norvig, P. (2016). *Artificial Intelligence: A Modern Approach* (3rd ed.). Prentice Hall Press.
44. Samuel, A. L. (1959). *Some Studies in Machine Learning Using the Game of Checkers*. *IBM Journal of Research and Development*, *3*, 210–229.
45. Scikit-learn. (2020). *Scikit-learn user guide - Release 0.23.2*.
46. Sentance, R. (2019). *15 examples of artificial intelligence in marketing*. Retrieved July 12, 2019, from https://econsultancy.com/15-examples-of-artificial-intelligence-in-marketing/
47. *Slovenski računovodski standardi 2016*. (2015). Retrieved March 16, 2020, from Pravno-informacijski sistem Republike Slovenije website: http://www.pisrs.si/Pis.web/pregledPredpisa?id=DRUG4192#
48. Stanič, K. (2012). Upravljanje terjatev. In *Poslovanje in izterjava v EU*. Gospodarska zbornica Slovenije.
49. Statistica. (2017). *Debt collector predicts payment outcomes using data analytics*. Retrieved April 12, 2020, from https://www.tibco.com/sites/tibco/files/resources/eos-group-uses-advanced-analytics-and-automation-to-predict-debt.pdf
50. van de Geer, R., Wang, Q., & Bhulai, S. (2018). Data-Driven Consumer Debt Collection via Machine Learning and Approximate Dynamic Programming. *SSRN Electronic Journal*, 1–32.
51. Volk, D. (2003). *Izvršba : izterjava denarnih terjatev in vse kar morate vedeti o sodnem postopku : priročnik*. Ljubljana: DZS.
52. Volk, D. (2015). *Izvršba - Izterjava denarnih terjatev*. Maribor: Poslovna založba MB.
53. Walker, J. (2019). *Artificial Intelligence Applications for Lending and Loan Management*. Retrieved July 12, 2019, from https://emerj.com/ai-sector-overviews/artificial-intelligence-applications-lending-loan-management/
54. Wang, L., Lei, Y., Zeng, Y., Tong, L., & Yan, B. (2013). Principal Feature Analysis: A Multivariate Feature Selection Method for fMRI Data. *Computational and Mathematical Methods in Medicine*, *2013*, 1–7.
55. Wejer-Kudełko, M., & Łada, M. (2018). Success factors and barriers to the effective debt collection process. *Prace Naukowe Uniwersytetu Ekonomicznego We Wrocławiu*.
56. Wijaya, C. Y. (2016). *Categorical Feature Selection by Chi-Square*. Retrieved August 2, 2020, from https://towardsdatascience.com/categorical-feature-selection-via-chi-square-fc558b09de43
57. Wirth, R., & Hipp, J. (2000). *CRISP-DM: Towards a Standard Process Model for Data Mining*.

58. *Zakon o izvršbi in zavarovanju (ZIZ)*. (1998). Retrieved June 14, 2020, from Pravno-informacijski sistem Republike Slovenije website: http://www.pisrs.si/Pis.web/pregledPredpisa?id=ZAKO1008
59. Zurada, J., & Lonial, S. (2011). Comparison Of The Performance Of Several Data Mining Methods For Bad Debt Recovery In The Healthcare Industry. *Journal of Applied Business Research (JABR)*, *21*(2).

APPENDICES

Appendix 1: Povzetek (Summary in the Slovene language)

V magistrski nalogi raziskujemo priložnosti in izzive uporabe metod strojnega učenja za napovedovanje uspešnosti izterjave dolgov. Z uspešnim napovedovanjem rezultatov izterjave lahko podjetje selektivno dodeli sredstva dolgovom, ki imajo večjo verjetnost poplačila. Podjetje tako prihrani operativne stroške izterjave dolgov, ki imajo nizko verjetnost poplačila, ali pa take dolgove celo odpiše. Pri nastanku magistrske naloge je sodelovalo slovensko podjetje, ki se ukvarja z izterjavo dolgov. Raziskava podjetju predstavlja pilotno študijo uporabe strojnega učenja za napovedovanje izterjave dolgov in optimizacijo procesa izterjave. Podatki, uporabljeni v empiričnem delu, so last tega podjetja. Glavna ovira raziskave izvira iz dejstva, da podjetja, ki se ukvarjajo z izterjavo dolgov, velikokrat od upnikov prejmejo zgolj najnujnejše podatke za opravljanje storitve izterjave. Zato smo se odločiti napovedovati izid izterjave v dveh časovnih točkah. Prvič ob predaji dolgov, ko so na voljo zgolj osnovni finančni in morebitni kontaktni podatki, ki so predani s strani upnikov, in drugič po preteku enega meseca od začetka izterjave. V drugem primeru so vključeni tudi podatki, ki so bili generirani med samim procesom izterjave.

Cilji raziskave so: (1) pregledati relevantno literaturo o uporabi strojnega učenja na področju izterjave dolgov, (2) ugotoviti, ali je uporaba algoritmov strojnega učenja smiselna v okviru poslovnega primera, (3) določiti ključne priložnosti in izzive skozi implementacijo različnih algoritmov strojnega učenja in (4) opisati sam proces implementacije algoritmov strojnega učenja, identificirati najuspešnejši model za napovedovanje dolgov in prepoznati ključne dejavnike implementacije.

Empirični del naloge je ločen na dva dela. V prvem delu zgradimo različne modele strojnega učenja, ki napovedujejo izid izterjave ob predaji dolgov, medtem ko v drugem delu napovedujemo izid po preteku enega meseca od začetka izterjave. Za obe časovni točki napovedovanja opišemo sledeče korake: razumevanje podatkov, pripravo podatkov, modeliranje in ocenitev modelov. Za gradnjo modelov je uporabljena metodologija CRISP-DM.

Najobetavnejši modeli iz prve časovne točke temeljijo na metodi podpornih vektorjev, logistični regresiji in nevronskih mrežah. Slednji model najuspešnejše napoveduje izid izterjave. Zaradi pomanjkanja relevantnih podatkov ob predaji dolgov, vsi modeli kažejo znake nezadostnega prilagajanja podatkom (angl. *underfitting*). Ker imajo vsi modeli težave z uspešnim napovedovanjem, se sprašujemo o smotrnosti napovedovanja izida izterjave v tej časovni točki.

Napovedovanje izterjave po preteku enega meseca od začetka izterjave se je izkazalo za boljšo rešitev. Najučinkovitejši modeli v tej časovni točki temeljijo na naključnih gozdovih, nevronskih mrežah ter metodi podpornih vektorjev. Kot najuspešnejši se je izkazal model naključnih gozdov.

Doseženi so bili vsi štirje cilji raziskave. (1) V okviru prvega cilja smo v teoretičnem delu naloge predstavili več raziskav, katerih namen je optimizacija procesa izterjave z uporabo algoritmov strojnega učenja. (2) Rezultati empiričnega dela kažejo na to, da je uporaba algoritmov strojnega učenja smiselna za napovedovanje izterjave dolgov, če imamo primerne podatke. V nasprotnem primeru lahko pride do težav pri napovedovanju, kar smo izkusili pri napovedovanju uspešnosti izterjave dolgov v prvi časovni točki. (3) Glavni izziv raziskave je predstavljalo uspešno napovedovanje izterjave na začetku procesa izterjave, kjer so na voljo zgolj osnovni podatki. Glavno priložnost za uspešno napovedovanje predstavlja druga časovna točka, kjer se napovedovanje izvaja en mesec po začetku izterjave. (4) Celoten proces gradnje modelov je zajet v empiričnem delu naloge. Izmed vseh zgrajenih modelov se je najbolje izkazal model naključnih gozdov iz druge časovne točke, ki je dosegel vrednost f_1-ocene 0,88. V teku gradnje modelov smo identificirali dva ključna dejavnika uspeha. Prvi zajema dobro poznavanje podatkov in osredotočanje na poslovni cilj. Obojno je bistvenega pomena pri izbiri mer za ocenitev modelov. Izbira napačne mere lahko povzroči, da učinkovitost napovedovanja modelov ni optimalna. Drugi ključni dejavnik je uporaba strukturiranega pristopa za načrtovanje implementacije algoritmov strojnega učenja, kot je metodologija CRISP-DM, ki ponuja jasen pregled nad celotnim procesom implementacije. Z uporabo standardnega pristopa zmanjšamo možnost napak pri gradnji modelov in se tako izognemo situaciji neuspešnih modelov zaradi napak v postopku gradnje.

V podjetju, ki je posredovalo podatke za to raziskavo, so z rezultati zadovoljni. Rezultati iz prve časovne točke so potrdili domnevo, da z razpoložljivimi podatki ne bo mogoče učinkovito napovedovati izida izterjave ob prevzemu dolgov. Rezultate iz druge časovne točke ocenjujejo kot obetavne. Zaposleni v podjetju lahko uporabijo napoved modela naključnih gozdov kot dodatno informacijo pri odločanju, ali v posameznih primerih opustiti postopek izterjave dolgov ali ne. Verjamejo, da se lahko z razširitvijo obsega podatkov in več primeri uspešnost napovedovanja dodatno izboljša do te mere, da bo model lahko neodvisno in zanesljivo napovedal izid postopka izterjave.