TALLINN UNIVERSITY OF TECHNOLOGY

School of Business and Governance

Sina Ansari Fard

Risk Prediction for Loan Applications By Machine Learning Algorithms

Bachelor's thesis

Programme International Business Administration, Specialisation Finance

Supervisor: Pavlo Illiashenko

Tallinn 2023

I hereby declare that I have compiled the thesis independently and all works, important standpoints and data by other authors have been properly referenced and the same paper has not been previously presented for grading.

The document length is 8875 words from the introduction to the end of conclusion.

Sina Ansari Fard
17.04.2023

TABLE OF CONTENTS

ABSTRACT

Today, Machine Learning, as one of the growing technologies worldwide, has had made its own path to various sectors of industries including peer to peer lending companies and banks. While various models have been used for this purpose, emerging machine learning algoritthms have challenged the performance of existing classical models that are used for predicting the probablity of default.

Therefore, finding out which Machine Learning algorithm to use for the same purpose; predicting a more accurate probability of default than exisiting classical models has gained importance in P2P lending industy.

This paper uses the data from Bondora annual public reports to evaluate several Machine Learning algorithms used to predict the risks associated with peer to peer loans againts the traditional methods. In general, it has been demonstrated that machine learning algorithms perform better than conventional models at tasks involving loan risk prediction. It is crucial to remember that the effectiveness of machine learning algorithms depends on the quality and quantity of the training data, the algorithm and hyperparameters chosen, and other factors.

After implementing nine Machine Learning algorithms, the Random Forest algorithm and Gradient Boosting algorihm had higher accuracy compared to other machine learning algorithms. Thus, these two algorithms are suggested to be used by P2P lending companies to decide whether to assign a loan to a borrower or not.

Keywords: Machine Learning algorithms, Loan Risk Prediction

1. INTRODUCTION

Financial technology is largely regarded as one of the most significant breakthroughs in the financial industry, and it continues to grow at a rapid pace (Lee and Shin 2018). In definition, financial technology, or FinTech, is the application of technology to the design and delivery of financial goods and services. It has a direct effect on financial institutions, regulators, customers, and merchants across a wide range of industries (Leong et al. 2017). According to Lee and Shin (2018), peer-to-peer (P2P) lending is one of the six newborn growing FinTech business models. Simply put, P2P lending is the act of lending money to indiviuals or small and mid-size enterprises via online platforms that connects lenders and borrowers. One of the hot topics in this field is risk assessment of applicants. A P2P lending company, in order to make sure the client will be able to pay back the loan in agreed duration, assesses the risk of each applicant individually (Hsueh et al. 2017).

The rapid growth of P2P lending companies has made it impractical to perform a sufficient risk assessmet before assigning a loan and has led the companies to use different models in order to handle ths issue (Gahlaut et al. 2017). Also, as Duchamp (2016) believes, in financial industry including banks and P2P lending companies, the act of big data implementation was way far below the expected pace of data volume growth.

Therefore, In P2P lending, lenders might be unable to receive clear and sufficient information regarding their loan applicants. Besides, the systems they use for credit scoring are immature. As a result, their risk management and decision making based on the available information and system they use may not be efficiently realistic (Hsueh et al. 2017).

In fact, a major problem in P2P lending companies is their Risk Assessment methods. In other words, credit risk assesment has become a challenge for these companies (Suryono et al. 2019). Therefore, the possibility of utilizing more accurate tools for this purpose has gained importance.

Unlike traditional data, Big Data refers to big expanding data collections that encompass a variety of formats, including structured, unstructured, and semi-structured data. Big Data is complex, making the use of strong technologies and smart algorithms more necessary than ever. As a result, typical static Business Intelligence tools are no longer effective in Big Data applications. Big data has significant prospects and transformative potential for a variety of industries; nevertheless, it also presents unprecedented obstacles in utilizing such enormous and expanding volumes of data. Thus, properly-developed algorithms and efficient data mining methods are needed to get accurate results, keep a close eye on changes in various fields, and forecast upcoming observations (Mueller and Massaron 2021).

Data mining methods are important to discover interesting patterns and to extract value hidden in such huge datasets and streams. However, traditional data mining techniques such as association mining, clustering and classification lack of efficiency, scalability and accuracy when applied to such Big Data sets in a dynamic environment. Enterprises, however, require a reliable Financial Distress Predict (FDP) system in the current uncertain economic climate. Such a system is essential for risk management improvement and lending decisions by banks (Sun et al. 2017).

All being said, Today, P2P lending companies suffer from lack of efficient risk assessment models. At the same time, few sufficient solutions are suggested to the problem. One of the main solutions is Credit Risk and Loan Performance Prediction using Machine Learning Algorithms (Suryono et al. 2019).

In this study, these research questions are aimed to be answered.

1. Are ML algorithms more accurate than classical prediction models?
2. Which ML models are most efficient in loan applications' risk assessment for P2P lending companies?

In order to reduce the existing possibility of failure, this research tries to suggest a series of efficient and more accurate data mining models based on Machine Learning algorithms. These models will alleviate and improve prediction of either the applicants will be able to repay on time or not. In other words, these models examine the abiliy of the customers in repaying credit loans within the adjusted timeline by classifying the loan receivers as 'high risk or 'low risk. The term ‘low risk’ states that the loan receiver has an acceptable score and there has been no problematic payment records. On the other hand, the phrase ‘high tisk’ suggests the opposite, that the applicant has a bad credit score or there were records for delayed payments or past defaults. Because, as described by Gahlaut (2017), this approach eventually enables P2P lending companies to assign good credit that generates profit in their year to year revenue.

To answer the abovementioned questions, a historical data from Bondora (a P2P lending company) has been collected. This comprehensive dataset includes details about applicants that were granted a loan, including the calculated Probability of Default (PD) for each applicant which was measured by classical models.

A set of Machine Learning algorithms are aimed to be applied to this dataset. These algorithms include Adaptive Boosting model, Gradient Boosting Model, Logistics Reggression model, K-nearest Neighbors model, Naïve Bayes model, Decision Tree model, Random Forest model, Support Vector Machine model and Artificial Nueral Network model. All data analysis process is done in Python programming language. Recently, Python has gained populairty among programming languages for data science due to its well-performing and convinient libraries, namely Scikit-learn, which is used for applying Machine Learning algorithms to datasets.

These algorithms are aimed to measure the Probibility of Default (PD) of the loans. After claculating the PD, each model will be ranked by the most accurate ones to least. Meanwhile, the results can be compared with the calculated PD by classical models.

Section 2 contains related research work. Section 3 gives the overall description about the data used in this study alonside with brief explanations about each algorithms. Section 4 discusses the empricial approach regarding this study, including implementing each algorithm and preprocessing of data.

2. LITERATURE REVIEW

This chapter of the thesis is divided into three sections. The first section will look into an overview of Probability of Default (PD) in previous studies. Followed by second chapter that describes Machine Learning algorithms and finally the third section provides a summary of Machine Learning algorithms that can be applied for calculating Probibility of Default.

Numerous research have examined related challenges within the context of data mining in the banking and insurance analytics sector. This study tries to suggest a more comprehensive model with higher accuracy compared to previous approaches.

For instance, for loan prediction, Hassan and Abraham (2013) used supervised neural network models, which they developed after studying a German bank credit dataset with 20 variables. A thorough evaluation of all models was conducted, as well as the determination of their relative accuracy percentages. This study demonstrated that a machine learning algorithm can be applied for prediction of loan applications success rate in P2P lending companies.

Moving forward, to estimate loan risk, Jin et al (2015) used a data-driven methodology and data mining, comparing models including decision trees, support vector machines, and neural networks. In order to show the superior prediction, they used a 10-fold cross-validation strategy together with a high average percent hit ratio. An examination of the cumulative lift curve is utilized to determine the quality. The Support Vector Machine produced the best outcomes. As Jin et al. (2015) used the same approach, but adding two more algorithms to the study, namely Support Vector Machine and Decision Tree, the emphasize on possibility of different results from different algorithms gained more attention.

Support vector machine (SVM) was utilized by Hsueh et al. (2017) to categorize a bank credit dataset, and they came to the conclusion that SVM is more effective for credit rating as the quantity of data samples or other selection features rise.

In order to identify corporate defaulters, Ramakrishnan et al. (2015) ensemble model was based on the well-known boosting approach known as Adaptive boosting. This model was evaluated alongside different other classifiers, including Decision Trees, Support Vector Machines, Artificial Neural Networks, and Logistic Regression, and they discovered that their model performed significantly better than the other classifiers.

In the end, it can be claimed that the efficiency of each model is greatly dependent on the data preparation. Meanwhile, any model may perform poorly on one dataset but well on another. Although some models, such as Adaptive Boosting, are considered to have slightly more accuracy than others, such as Decision Trees, in broad terms. This reality necessitates testing each model on a given dataset to be able to determine which model fits the data the best.

All being said, this thesis takes a complementary but distinct approach from others in numerous respects. This study used machine learning algorithms to forecast a customer's potential in repaying credit loans within the due date, categorizing them as a 'low risk' or a 'high risk' customer. Second, unlike other approaches, this study implemented a broad range of existing Machine Learning models that are applicable to the dataset. This approach enables a more informed selection about which model to use for the same objective based on the findings and accuracy level of the model.

2.1 Probability of Default

Probability of Default (PD) measures how likely a borrower is to be unable to fulfill his or her contractual commitments and to go into default. Even though default doesn't always result in losses right away, it can raise the possibility of bankruptcy and eventual losses. In other words, for banks and lending companies, Default is a concern. P2P lending companies must first determine the likelihood that the borrower will default over a specific time horizon in order to accurately estimate credit risk (Bandyopadhyay 2016).

For predicting Probability of Default, there are numurous models utilized regularly, which can be divided into categories below;

- Discriminant Analysis (DA)
- Linear Regression Models eg. Logistic Regression
- Inductive Models

These models are also known as credit scoring models (Resti and Sironi 2007).
This study summarises the advantages and disadvantages of the first two types of the credit scoring models.

A specific case of linear regression models is deemed to be logistic regression. However, generic regression models' normality assumptions are broken by the binary answer variable. According to a logistic regression model, the fitted probability of the occurrence is a linear function of the observed values of the explanatory variables that are available. This method's main benefit is that it can generate a straightforward probabilistic classification formula. The limitations of Logistic Regression include its inability to adequately address issues with non-linear and interaction effects of explanatory factors (Han et al. 2001).

Meanwhile, Fisher's rule, also referred to as discriminant analysis, is a different method used to analyze response variables that have binary results. DA is a variant of logistic regression and is predicated on the idea that the explanatory variables are distributed as a multivariate normal distribution with a shared variance-covariance matrix for every type of outcome variable. Fisher's rule aims to reduce distances within every category while increasing distances between different categories. The benefits and drawbacks of DA are equivalent to those of LR (Hand et al. 2001).

2.2 Machine Learning

Machine learning has emerged as a critical component in a variety of financial system functions, ranging from the approval of advances to the monitoring of assets and resource allocation (Mathur 2018).

To make learning easier, one can divide tasks into groups based on how they work together with the environment (Shalev-Shwarts and Ben-David 2014). Simply put, learning is the ability to adapt to new situations in response to external stimuli while still recalling the majority of one's previous experiences (Bonaccorso 2017).

As described by Mitchell (1997), in computer science, machine learning is a discipline that seeks to learn from data in order to enhance performance at multiple tasks such as prediction.

Thus, the primary goal of machine learning is to investigate, develop, and improve mathematical models that can be trained once or repeatedly on context-related data provided by a generic environment in order to predict the future and make decisions without complete knowledge of all relevant factors (Bonaccorso 2017).

To determine whether and which machine learning method(s) to employ, it is necessary to initially define the research problem. According to the framework developed by Hernán et al. (2019), there are three essential research tasks in the field of data science: description, prediction, and causal inference. The application of machine learning is capable of accomplishing all three tasks (Jiang et al. 2020). The field of Machine Learning can be broken down into three subsets: supervised learning, unsupervised learning, and reinforcement learning (Oussous et al. 2018).

In this paper, supervised learning algorithms and models are used to fulfill the task of prediction.

2.2.1. Supervised Learning

There are two types of successful machine learning algorithms: those that make decisions and those that learn from past examples. In the process known as supervised learning, an algorithm is given a set of inputs and desired outputs, and then searches for a way to generate the desired output given the input. Specifically, the method can generate a result for a previously unseen input without human intervention.

It is possible for an algorithm to learn from example data and associated target responses, which can be positive numbers or text labels, such as classes or tags, in order to predict the correct response when new examples are presented. This strategy, which emphasizes the need for repetition, is analogous to human learning under the supervision of a teacher. It is the responsibility of the teacher to provide examples for the student to remember, and it is the responsibility of the student to derive general laws from the specific examples provided. It is necessary to distinguish between regression problems, which target a numerical value, and classification problems, which target a qualitative variable (e.g., a class or a tag) (Mueller and Massaron 2021).

There are two fundamental types of supervised machine learning tasks: classification and regression issues. Classification issues are the most prevalent. The objective of classification is to predict a class label, which is a selection from a specified list of potential labels or categories. The purpose of regression tasks is to predict a continuous number, or a floating-point number in programming terms or a real number in mathematics. Test set evaluation is the only way to determine whether or not an algorithm will perform adequately on new data. However, the complexity of our model will increase our ability to predict using the training data (Müller and Guido 2016).

Accordingly, the term "supervised learning" is mainly used in prediction tasks because the objective is to predict a specific expected outcome, which aligns with the study' objective of classifying the loan applicant's future behavior based on their provided data and profile.

2.2.2. Training Set and Test Set

It is widely accepted that the most crucial stage of constructing any model is choosing the samples from which to train or build it. For the vast majority of standard chemometric modeling procedures, the data set must be split into a training or learning set and a test set.

Unluckily, the data that were used to create the model prevent us from evaluating it. This is due to the fact that the model can always recall the whole training set and, as a result, can predict the label with accuracy at any point throughout the training set. This "remembering" is at odds with how broadly applicable our approach can be. In other words, it is impossible to promise how well the model will perform when given new data. We expose the model to fresh, labeled data that it has never seen before in order to assess its performance. In most cases, this is accomplished by halving previously acquired labeled data. A portion of the data used to build our machine learning model is known as the training data or training set. The test set refers to the remaining data that will be used to assess how well the model performed (Müller and Guido 2016, pp. 28).

2.2.3 Classification

Classification is the mathematical simulation of a function that converts a set of input variables (X) into distinct output variables (Y). It involves labeling a number of input data variables with class labels. In simple terms, it is the classification of updated data into sub-classes based on previously established labels. In machine learning, classification algorithms are used to predict results and manage labeled datasets. As a broad field of study, machine learning focuses primarily on categorization and clustering. We can train our machine learning model based on the structured or unstructured data we have, as well as the type of output we require, which basically is the distinct values such as True or false or simple groupings (Isik et al. 2007).

Thus, classification is the process of identifying a function that facilitates the classification of a given dataset into distinct classes based on a variety of parameters. After being trained on a training dataset, a computer program classifies the data variables (Pandimurugan et al. 2022).

Here is the summory of classifiers that are used in this study.

Table 1. Types of Classifiers

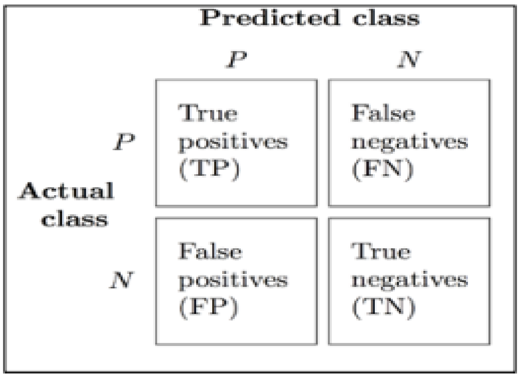
Classifier Name	**Classifier Type**
Decision Tree	Ensemble
Gradient Boosting	Ensemble
Random Forest	Ensemble
K-nearest Neighbors	Instance-based
Logistic Regression	Linear
Artificial Neural Networks	Non-linear
Support Vector Machine	Linear
Adaptive Boosting	Ensemble
Naïve Bayes	Linear

Source: Pandimurugan et al. 2022

2.2.4 Evaluation of models

In order to make sure the models that are used are sufficient and have reliable results for our study, few metrics are used to evaluate the models.

One of the most common metrics that is used to evaluate a machine learning model is Confusion Matrix. Confusion matrix is a matrix that summarizes the performance of a learning algorithm. The confusion matrix is a simple square matrix containing the counts of a classifier's True positive (TP), True negative (TN), False positive (FP), and False negative (FN) predictions, as depicted in the figure below.



		Predicted class	
		P	*N*
Actual class	*P*	True positives (TP)	False negatives (FN)
	N	False positives (FP)	True negatives (TN)

Figure 1. Confusion Matrix

Source: Raschka and Mirjalili 2017

As defined by Raschka and Mirjalili (2017), those divisions in a confusion matrix that have been accurately predicted to occur are true positives. True negatives, on the other hand, represent classes that are likely to fail. False positives and false negatives indicate the model's inability to accurately predict. Thus, true positives (TP) and true negatives (TN) reflect the accuracy of the model, whereas false positives (FP) and false negatives (FN) reveal its flaws. Both prediction error (ERR) and accuracy (ACC) provide a count of misclassified samples. Therefore:

$$ACC = \frac{TP+TN}{TP+TN+FP+FN} = 1 - ERR \qquad (1)$$

Also, Müller and Guido (2016) describe that the confusion matrix can be summed up in a number of different ways. Precision and recall are the most frequently used. Precision quantifies the proportion of predicted positive samples that actually are positive, known as true positives. When the goal is to reduce the number of false positives, this metric is used as a performance indicator. Recall quantifies the proportion of positive samples obtained from positive predictions. Recall is used as a performance metric when it is crucial to identify all positive samples, or when it is necessary to prevent false negatives.

$$Precision = \frac{TP}{TP+FP} \qquad (2)$$

$$Recall = \frac{TP}{TP+FN} \qquad (3)$$

Indeed, precision and recall are essential metrics. However, focusing solely on one will not reveal the entire picture. One way to summarize them is the F1-score, which is defined as the harmonic mean of precision and recall. Because it takes precision and recall into account, it may be a more suitable metric for the classification strategy than accuracy. The following is the definition of the F1-score (Müller and Guido 2016, pp. 283).

$$F1 - score = 2\frac{precision.recall}{precison+recall} \quad (4)$$

All being said, higher amount of these four metrics; accuracy, precision, recall and f1-score shows that the model performance is higher. In this study, all four metrics for each model are presented.

Another metric for evalution of a Machine Learning model performance is ROC analysis. The performance of a binary classification system as the discrimination threshold is changed is depicted graphically by the ROC (Receiver Operating Characteristic) curve. At different threshold values, the curve compares the true positive rate (TPR) and false positive rate (FPR). The ratio of positive cases that are correctly classified as positive is known as the TPR, whereas the ratio of negative cases that are wrongly classified as positive is known as the FPR. With values ranging from 0.5 (random guessing) to 1 (perfect classification), the area under the ROC curve (AUC), which can be applied as a single performance indicator, describes the entire performance of the classifier. When evaluating classifiers and deciding on the best threshold for a given problem, the ROC curve is a valuable tool (Fawcett 2006).

2.4 ML Algorithms for PD

There are eight models that have been used in this paper. In this section, each model is described briefly.

Naïve Bayes model

Using the information contained in the feature vector, Bayesian classifiers assign the most plausible class to a given example characterized by its own feature vector. When learning such classifiers, it is advantageous to assume that features are independent of the class for which they are employed. The formula offers a more accurate explanation.

$$P(X|C) = \prod_{i=1}^{n} P(Xi|C)\,,\ X = (X_1, X_2, \dots, X_n) \qquad (5)$$

In this formula, 'X' represents a vector of observed random variables, denoted as feature. C represents the class of unobserved random variables (Rish 2001).

A straightforward and effective technique for predictive modeling is naive bayes. This model measures the probability of each class and the conditional probability for each class given each x value. These two forms of probabilities can be derived from the training data directly by the model. Once determined, the probability model can be applied to Bayes theorem to produce predictions for new data. It is typical to assume a Gaussian distribution (bell curve) when the data is real valued so that one can quickly estimate these probabilities. Since the model presumes that each input variable is independent, the name ‘naïve’ Bayes is given to this model (Shobha and Rangaswamy 2018).

The naive Bayes classifier has the advantage of only requiring a small amount of training data to form an impression of the classification variables. The correct classification is determined using the Bayesian approach if the correct category is more probable than the others. It is not necessary to accurately estimate the probabilities of each category. In other words, the aggregate classifier is capable of overlooking significant flaws in the underlying naive probability model (Pandey and Pal 2011).

Meanwhile, Zhang (2004) believes that despite the fact that naive Bayes is a good classifier, it is a relatively poor estimator.

Logistic Regression model

Logistic regression is a quantitative modeling technique used to establish a relationship between numerous X_s and a binary dependent variable, such as "D" (Kleinbaum et al. 2002). It is one of the most widely used statistical and data mining techniques for the classification and analysis of boolean and scaled response datasets by academics and researchers.

The majority of linear classification models are binary-only and cannot be extended to multiclassification. The one-versus-the-rest approach is one of the most common ways to circumvent this problem and extend binary classifications into multiclass classification techniques. In the one-vs.-rest method, a binary model is learned for each class in an attempt to differentiate it from all other classes, resulting in the same number of binary models as classes. In order to make a prediction, all binary classifiers are applied to a test point. As the prediction, the class label corresponding to the classifier with the highest score on its single class is returned (Müller and Guido 2016, pp. 63).

As described by Kleinbaum et al. (2002), in order to be able to obtain the logistic model from the logistic function, we define 'Z' as the linear sum α plus β_1 times X_1 , plus β_2 times X_2, and so on to β_k times X_k, where the X_s are independent variables of interest and α and the β are constant terms that represent unknown parameters.

$$Z = \alpha + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k \qquad (6)$$

Consequently, Z is a composite index that combines the X_s.

When the linear sum expression for Z is substituted on the right-hand side, the model formula becomes:

$$P(X) = \frac{1}{1+e^{-(\alpha+\sum \beta i X i)}} \ , \ X = (X_1, X_2, \ldots, X_k) \qquad (7)$$

Inherent advantages of logistic regression include the ability to produce probabilities and its applicability to multi-class classification problems, among others. Another benefit is that the majority of Logistic Regression model analysis procedures are based on the same principles as those used in linear regression. In addition, Logistic Regression is applicable to the vast majority of unconstrained optimization strategies (Maalouf 2011).

As soon as logistic regression is implemented, the model's complexity is significantly reduced, particularly when no or few interaction terms and variable transformations are employed. In this situation, there is less of a problem with overfitting. Overfitting refers to the process of developing a model that is disproportionately complex in relation to the amount of data collected to date. Overfitting occurs when a model is overfit to the characteristics of a training set, resulting in a model that performs well on the training set but cannot generalize to new data. Applying an overfitting model to additional datasets is therefore not recommended (Dreiseitl and Ohno-Machado 2002).

Neural Networks model

A rising corpus of research emphasizes the application of neural network classification models, specifically backpropagation neural networks (BPN), in consumer loan applications in regards to the issue of classification accuracy in loan applications. Neural networks are complex hardware or software created in a digital environment to resemble the functions of the human brain. Neural networks are nonlinear models that categorize items based on how well they can spot data patterns. (Malhotra and Malhotra 2003).

The most basic form of neural network is the perceptron. This neural network includes a single input layer and a single output node, as shown in Figure 2. The weights w1 through W_d are present along the edges from the input to the output, where they are multiplied and summed at the output node, which is the last node. The entire value is then transformed into a label for a particular class using the sign function. The activation function depends on the sign function in several ways. An essential component of building a neural network is choosing an activation function (Aggarwal 2018, pp. 5-11).

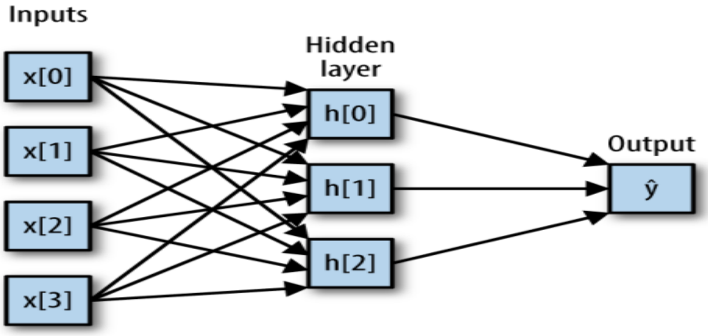


Figure 2. Artifical Neural Networks Algorithm, Perceptron

Source: Müller and Guido (2016), pp. 106.

According to Müller and Guido (2016), there are two coefficients to be learned for this model; one for any input and any hidden unit which make up the hidden layer, and one for each hidden unit and the output. These coefficients are also known as weights. The entire equation for estimating in the case of regression for the small neural network shown in Figure 2 using a tanh nonlinearity would be:

$$h_0 = \tanh\,(W_{00}X_0+W_{10}X_1 + W_{20}X_2 + W_{30}X_3)$$
$$h_1 = \tanh\,(W_{01}X_0+W_{11}X_1 + W_{21}X_2 + W_{31}X_3) \qquad (8)$$
$$h_2 = \tanh\,(W_{02}X_0+W_{12}X_1 + W_{22}X_2 + W_{32}X_3)$$

Which is followed by calculation of the value of ŷ through the formula below.

$$\hat{y}= V_0h_0 + V_1h_1 + V_2h_2 \qquad (9)$$

In this formula, ‘W’ represents the weights among the input ‘X’ and the hidden layer ‘h’, while ŷ represents the weights among the hidden layer ‘h’ and the dependent variable.

The input features are represented by ‘X’, and the weights ‘V’ and ‘W’ are learned from data. The output of the computation is represented by ‘ŷ’, while intermediate computations are represented by ‘h’. The number of nodes in the hidden layer is a crucial parameter that the user must select for the system to function properly (Müller and Guido 2016, pp. 106-107).

Neural network algorithms are more sensitive to overfitting than logistic regression because they are more flexible than logistic regression. The size of a network can be reduced by reducing the number of variables and hidden neurons within the network, as well as by pruning the network following training. Alternately, one could stipulate that the model's output must be suitably smooth. This can be achieved through regularization, or weight decay in the context of neural networks. As its name suggests, weight decay is a procedure similar to logistic regression's shrinkage in that it limits the magnitude of the weights. Weight loss and reduction soften the edges of decision limits. In contrast to unrestricted decision borders, sufficiently smooth decision borders are less adaptable to the specifics of a data set and are therefore incapable of adjusting to the specifics of a data set (Maalouf 2011).

In this research, overfitting is not an issue because the feature scaling is implemented to the dataset prior to the model implementation.

Support Vector Machine model

Support Vector Machine (SVM) has newly emerged as the superior neural network for dealing with classification and forecasting problems due to its exceptional characteristics. Due to the superior qualities of generalization performance and global optimum, this is the case. Vapnik developed SVMs in 1995 as a cutting-edge artificial neural network (ANN) based on statistical learning. In recent years, it has attracted a great deal of interest from a variety of research communities due, among other things, to its exceptional ability to tackle classification problems and its novel approach to improving the generalization property of Artificial Neural Networks (Li et al. 2006).

The Support Vector Machine (SVM) model is a machine learning technology that facilitates the classification of diverse features with the corresponding labels. With the aid of the training dataset, an SVM algorithm can generate a model that can be used to assign appropriate labels to data from the testing set. The Binary linear classifier is also known as such due to the fact that it only requires two types of output to function. Support Vector Machine (SVM) is a machine that generates a hyperplane or set of hyperplanes in a high- or infinite-dimensional space for classification and regression tasks. Examples are plotted in space so that distinct classes appear to be separated by a gap in the center of the plot. The term hyperplanes is used to describe these separated spaces. Each hyperplane contains samples from the same category or label as the one preceding it. SVM algorithms build models that distribute new examples from the testing set of data into one of two categories based on examples from the training set that have been assigned to one of the two groups, as determined by the training set. The SVM model partitioned the space into two hyperplanes, with each hyperplane containing one of the categories A and one of the categories B. The hyperplane with the greatest distance to the nearest training data point of any class achieves a good separation in this situation, according to intuition (Kothari and Patel 2015).

Figure 3 illustrates the Support Vector Machine algorithm in its most fundamental configuration.

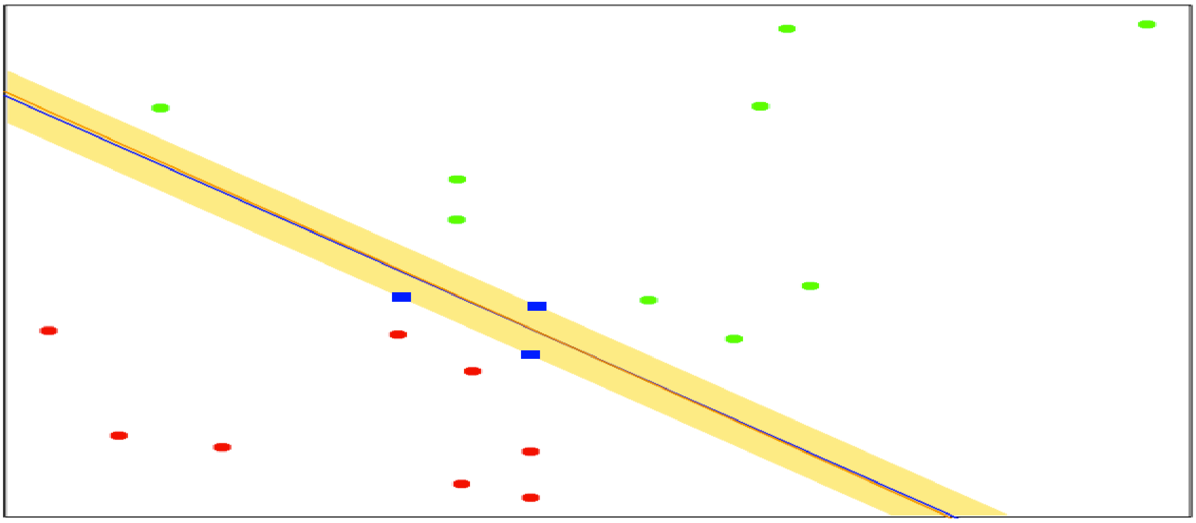


Figure 3. Support Vector Machine Model

Source: Hastie et al. 2009, pp. 134

The shaded area denotes the greatest separation between the two groups. The ideal separation hyperplane that is shown as a blue line, cuts the area in half at its center, and there are three highlighted support points that are all on the area's margin. (Hastie et al. 2009, pp. 134).

Decision Tree model

Decision Trees are commonly used models for a variety of tasks, including categorization and regression. They are taught a hierarchy of if-else questions that ultimately leads to the decision-making process (Müller and Guido 2016, pp. 70).

A Decision Tree is a tree-based technique in which each path beginning at the root is characterized by data-separating sequences and continues until a Boolean result is reached at the leaf node. This hierarchical illustration of knowledge links contains nodes and links, which is what makes it so effective. When classification is achieved through the use of relations, nodes denote purposes. Utilizing a sequential model that effectively and cohesively unifies a series of the fundamental tests, each test involves the comparison of a numeric feature to a threshold value. The conceptual principles in the neural network of connections between nodes are significantly simpler to generate than the network's numerical weights. Decision Tree is predominantly employed for grouping purposes. In addition, Decision Tree is a frequently used classification model in Data Mining. Each tree is composed of nodes and branches that are interconnected. Each subset defines a value that a node can use to determine its classification status, with nodes representing the features within a classification category. Due to the ease with which they can be analyzed and the precision with which they can be applied to multiple data formats, decision trees have found use in numerous fields (Charbuty and Abdulazeez 2021).

Consequently, a Decision Tree only poses a single question and, based on the answer (yes or no), further divides the tree into subtrees by severing the branches into subtrees. A decision tree is a model of a thinking human mind that simulates the decision-making strategy of the mind. Being easy to understand and visualize, having good performance on large datasets of information while having shorter data preparation process and being limited to one output per attribute are considered as advantages of this model. On the other hand, having potentially complex structure and possibility of duplication in the tree are counted as disadvantages of the Decision Tree model (Pandimurugan et al. 2022).

Random Forest model

As we have seen, one of the primary drawbacks of decision trees is their tendency to overfit the training data. Random forests are one strategy for addressing this difficulty (Müller and Guido 2016, pp. 83).

The Random Forest is comprised of numerous decision trees, each of which will be at full maturity; therefore, there will be no need for pruning. The greater the number of decision trees, the more precise the outcome will be, without overfitting. The overall estimate will be calculated using the random forest technique, which has the advantage of automatically selecting features (Lin et al. 2017).

The Random Forest algorithm is a classification method that utilizes multiple decision trees simultaneously while training the model on the data. It works as a meta estimator by constructing numerous trees on various sub-data sets to perform its task. The final output of the process is determined by the mode of all the classes that have been evaluated. The accuracy of a forest's predicted outcome is directly proportional to the number of trees it contains. Therefore, a higher number of trees in the forest results in greater accuracy. The forest algorithm improves its predictive power and precision by taking the average of all the decision trees within the forest. A random forest classification is a collection of many decision trees that are grown from the training data and serve as the fundamental learners in the classification. Thus, a random forest is composed of trees that exploit correlations between the various properties of the data points or experiments to deal with non-linearity. The dataset remains unchanged, and only a subset of the complete dataset is used to train the model each time to teach the model new information. The advantages of this model are numorous, namely minimizing of overfitting problem, accuracy improvement by reduction of variance in the data, being able to handle non-linear parameters efficiently. In contrast, having a complex structure and requiring longer training period are cons to this model (Pandimurugan et al. 2022).

Adaptive Boosting model

Boosting is a machine learning technique based on the concept of developing a highly accurate prediction rule by combining numerous somewhat weak and inaccurate rules. Freund and Schapire's AdaBoost algorithm was the first practical boosting algorithm and remains one of the most commonly used and studied, with multiple applications. Boosting is a strategy that improves the performance of machine learning algorithms by combining "weak learners" to get a highly accurate classifier or a better match for the training set. Weak learners are classifiers with a low rate of precision. In other words, they lack statistical significance and cannot be relied upon (Schapire 1990).

Studies have shown that the adaptive boosting algorithm combined with the decision-tree method is a precise learning technique. AdaBoost, like other ensemble algorithms, creates a collection of classifiers and then uses a voting mechanism to classify test samples. AdaBoost generates multiple classifiers successively by directing the underlying learning algorithm to focus on training examples that were incorrectly classified by the previous classifiers. Therefore, Adaptive Boosting is expected to be more accurate than the initial model without boosting (Margineantu and Dietterich 1997).

Gradient Boosting model

Gradient boosting builds regression models by adding simple parameterized functions (known as base learners) sequentially to fit the current residuals, which are calculated using the gradient of the loss function being minimized. The base learners are fit to the pseudo-residuals using least squares at each iteration. To improve the accuracy and speed of gradient boosting, randomization is incorporated into the process. This involves randomly selecting a subset of the training data at each iteration and using it to fit the base learner and compute the model update. This randomized approach improves the robustness of the model against overfitting by reducing the capacity of the base learner (Friedman 2002).

K-nearest Neighbours model

Instance-based learning techniques do not create a universal, precise description of the target function during training. Instead, they store the training examples and defer generalization beyond these examples until a new instance requires classification. Whenever a new query instance is encountered, the method analyzes its relationship with the previously stored examples to determine the value of the target function for the new instance (Mitchell 1997, pp. 230).

To construct this model, it is sufficient to store the training set. When generating a forecast for a new data point, the algorithm locates the training set point that is most comparable to the new data point. Then, the new data point is labeled with the label of this training point. The "k" in "k-nearest neighbors" indicates that instead of relying on the single nearest neighbor to the new data point, any fixed number k of neighbors in the training set can be taken into account, such as the nearest three or five neighbors. Finally, the majority class among these neighbors can be used to generate a prediction. Digging into pros and cons of the model, this model can be applied to any dataset thanks to its simplicity. However, this model is sensitive to outliers. Also, finding optimal ‘k’ can be challenging sometimes (Müller and Guido 2016, pp. 21).

3 DATA AND METHOLODOGY

This chapter focuses on the data that was used for the study. Starting from basic and general information regarding the dataset, this chapter continues on describing how the data was prepared for model implementation. Afterwards, this chapter takes a deeper look on the dataset from statistical perspectives and finally explains how the dataset was divided for training the algorithms.

3.1 Data

The dataset has been obtained from Bondora P2P lending company website, public reports. The dataset is a high-dimensional dataset accumulated through years 2009 till 2022.

The dataset is in Comma Separated Values (CSV) format. The comma-separated values (CSV) format is a widely used text file format that contains multiple records (one per line), and each field is delimited by a comma.

The dataset is structured with 112 columns and 203559 rows. Each row represents a specific loan applicant. There are 112 attributes for each loan applicant.

3.2 Data Pre-processing

This section explains how loan default prediction models are constructed.

Data cleansing is a vital step in machine learning. It is an essential component of developing a machine learning model. It is the process of ensuring that the data set contains no irrelevant or incorrect information. Data cleaning, also known as data cleansing or scrubbing, is the process of detecting and removing errors and inconsistencies from data in order to improve data quality (Bayraktar et al. 2018; Rahm and Do 2000).

Dirty data may cause our model to produce unsatisfactory results, so cleaning the data may resolve this issue by modifying the unclean data. This procedure is performed for each model utilized in this study.

Main steps of the data cleaning process are as follows:

- Removing irrelevant variables (columns)
- Removing Null Values from every row and column
- Conversion of categorical values to numerical values

For removing irrelevant variables, The correlation between all variables and the target Variable ‘Staus’ was done. Afterwards, the correlation between the target variable and rest of the variables was studied that resulted in dropping unnecessary variables.

As a result, in our dataset, out of 112 columns, 99 columns were removed. The removed columns either had null values or were not relevant to our algorithms and their purpose. For instance, columns such as ‘LoanID’, ‘Country’ and ‘First date of payment’ are not relevant and were removed. Meanwhile, in order to be able to implement the algorithms to the dataset, all null values have to be removed. Otherwise, the algorithms woud face failure in implementation. Therefore, all rows and columns with null values were removed.

Eventually, the cleaned data contained 13 columns and 191254 rows. In other words, the algorithms are implemented for the dataset that corresponds to 191254 applicants. It worth to mention that one of the variables is the probability of default column that will be used to compare the Machine learning accuracy results with the estimated probability of default by the P2P lending company by using the classical credit scoring models.

The clean dataset contains the information for each applicant who submitted a loan application. When a new applicant applies, the proposed models attempt to predict the nature of loan distribution based on the applicant's data. The classification problem for a standard supervised classification task is to predict whether a loan will be approved or not. This paper contains a number of data collections, which are referred to collectively as the data collection. Table 2 lists some of the most important characteristics of the test dataset.

Table 2. Selected Variables (Features)

Attribute Name	Description
Age	The age of the borrower, numeric value
Amount	Amount of the assigned loan, numeric value
Interest	Interest gained from the loan, numeric value
Loan Duration	Duration of the loan in months, numeric value
Monthly Payment	Estimated amount the applicant has to pay each month, numeric value
Interest and Penalty Balance	Amount of unpaid interest and penalties, numeric value
Gender	Male/Female/undefined, numeric value scaled 0-2
Applied amount	Amount of applied loan, numeric value
Liabilities Total	Total monthly liabilities of the applicant, numeric value
Income Total	Borrower's total income, numeric value
Principal Balance	Unpaid principal amount, numeric value
Status	The current status of the loan application, non-numeric value

Source: Compiled by author based on data from Bondora

A process known as Feature Scaling is carried out in order to normalize the data by standardizing the available features in the dataset. This process is used to bring all features to the same level of measurement so that one significant feature's large magnitude data values do not affect the model. This process is one of the most crucial steps performed during data pre-processing and is the most important factor in determining whether a machine learning model is weak or strong. This technique can significantly reduce the time required to locate the support vectors in the Support Vector Machine model, thereby improving the convergence rate of the algorithm.

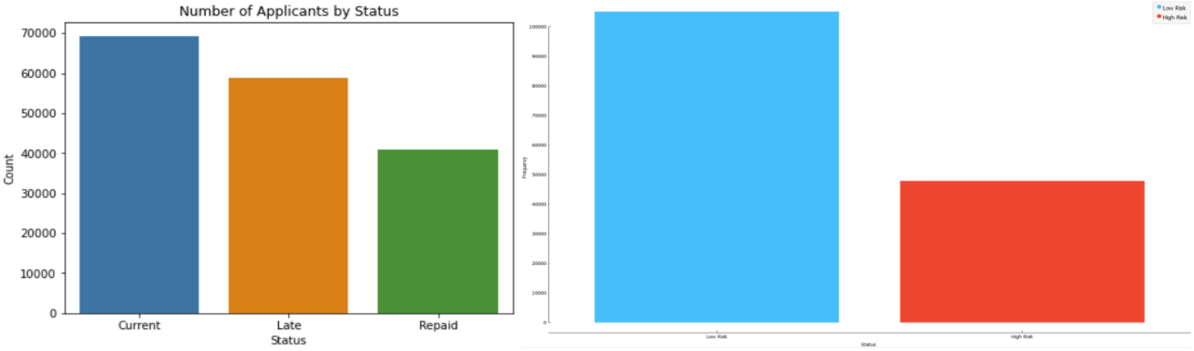


Figure 4. Target Variable ‘Status’

Source: Compiled by author through Orange software

As evidenced in figure 4, the 'Status' column indicates the current status of the loan. In the raw dataset, before cleaning the dataset, the status variable had values of ‘current’, ‘repaid’, or ‘late’. In this paper, the 'Status' column is divided into two categories, 'low risk' and 'high risk,' in order to make the target data more comprehensible. The status of customers with a low risk profile is either ‘current’ or ‘repaid’. On the other hand, customers with a status of ‘late’ is considered high risk. In accordance with the objective of this study, the outcomes of the models will be based on whether customers are high-risk or low-risk. This binary classification is the core principal of implementing Machine Learning classifiers.

3.3 Descriptive Statistics

In order to have a better overview of the data, some statistical features of important metrics have been studied.

To do so, for each metric, the statistical measures are extracted and sumoned in the table below. The report stands for 191254 applicants.

Table 3. Statistical report of variables

	MEAN	MEDIAN	MIN	MAX
AGE	40.28	39	1	77
AMOUNT	2613.81	2125	6.39	10632
APPLIED AMOUNT	2745.27	2125	10	10632
GENDER	0.37	0	0	1

INCOME TOTAL	1855.56	1256	0	101202
INTEREST	30.67	27.07	2	264.31
INTEREST AND PENALTY BALANCE	870.74	0	-2.66	78982.1
LIABILITIES TOTAL	413.58	264.56	4.06	250151
LOAN DURATION	48.11	60	1	120
MONTHLY PAYMENT	108.552	90.8	0.92	2368.54
PRINCIPAL BALANCE	1549.26	625.11	-34.2	10632
PD	0.21	0.18	0	0.99

Source: Author’s calculation using Orange software

One important thing to note in the statistical report of variables is the mean value of the Probability of Default (PD) that is calculated by the P2P lending company. 21% is relatively low probability of default. Accordingly, it can be intrepreted that due to the low percenatge of probability of default, it is risky for the company to grant loans to its applicants. Figure 5 takes a closer look at Probability of Default of loan applications and their amounts.

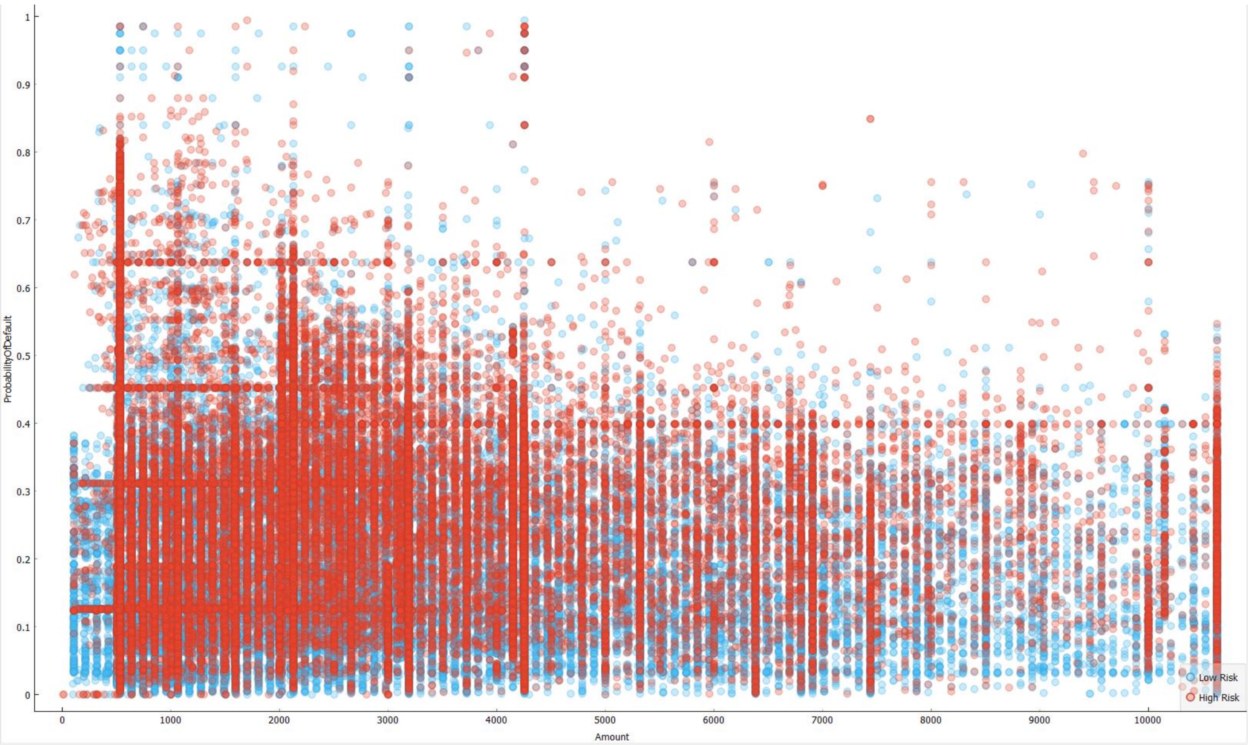


Figure 5. Probability of Default and amount of applications

Source: Compiled by author using Orange software

As visualized in figure 5, the density of observations are higher in lower left corner of the figure, amount values less than 5000 and probability of default values of below 0.5. Besides, this figure demonstrates that the Probability of Default values are not correlated with amounts of the loan applications. To find out which features are more correlated wih probability of default values, the correlation analysis of all other features of the cleaned dataset is done. According to the analysis, The correlation between Interest and Probability of Default of the loan applications is the highest.

3.4 Data Splitting

The first step in executing a machine learning model is separating data into a train set and a test set. Using the train test spilit package in the Python programming language, the dataset was divided into train set and test set. The data set is initially trained using a sub-data set. In this study, the sub-dataset corresponds to 80% of the dataset. This trained data is used to provide the algorithm with experience for use in the model. Training the data is a crucial component of any machine learning-based prediction model. Observations derived from trained data include multiple inputs and corresponding outputs. The trained data is then evaluated on additional sub-data sets and can be used for prediction. In other words, the trained model is implemented in the remaining 20% of the dataset which is called test dataset.

4. Empirical Results

In this section, the accuracy of each model is presented. After implementing each algorithm mentioned in this paper and visualising the accuracy of each model, the crediblity of each model was checked using set of parameters such as confusion matrix, recall & precision and F1-score. Hence, the Naïve Bayes model is not able to build a good model for the datasets and Gradient Boosting model is the most promising algorithm for building good predictive classification with higher accuracy. However, all of the applied models have shown a better performance than classical models. The results are breifly shown in table 4.

Table 4. Machine Learning Algorithms Accuracy Rate

RANK	NAME OF MODELS	ACCURACY
1	Gradient Boosting	0.950
2	Random Forest	0.949
3	Artificial Neural Networks	0.948
4	Adaptive Boosting	0.943
5	Decision Tree	0.927
6	K-nearest neighbours	0.923
7	Support Vector Machine	0.898
8	Logistic Regression	0.877
9	Naïve Bayes	0.764

Source: Compiled by author

As described in the table 4, The models ‘Gradient Boosting’, ‘Randome Forest’ and ‘Artificial Neural Networks’ are the most accurate models for the task of loan risk prediction. Meanwhile, the models ‘Logistic Regression’ and ‘Naïve Bayes’ are the least accurate models among the presented models.

One thing to take into account is the overall high accuracy of ML models in loan risk prediction tasks. In general, an accuracy rate of 0.8 to 0.9 and above is considered as an ideal and realistic level of accuracy for a Machine Learning model. Therefore, it is strongly suggested to P2P lending companies to utilize Machine Learning models in their risk management tasks.

The Figure 6 summorizes the results of the study in a decending order.

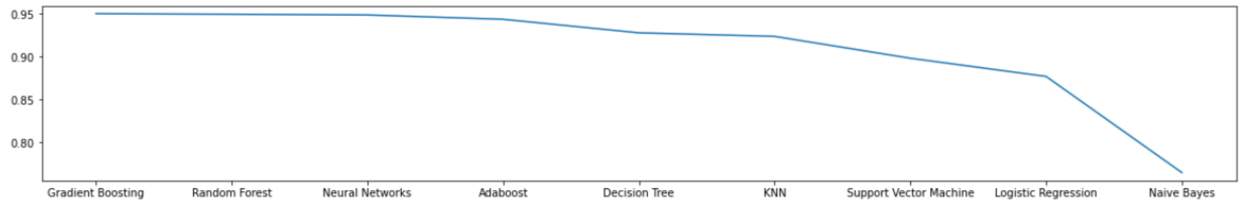


Figure 6. Machine Learning Models Accuracy Comparison

Source: Compiled by author using Python programming language

As mentioned before, to assess the performance of the applied models, this study has measured the metrics Precision, Recall, F-1 score and ROC curve analysis. Table 5 describes the calculated metrics.

Table 5. Model evaluations

	AUC	F1 SCORE	PRECISION	RECALL
GRADIENT BOOSTING	0.973	0.948	0.949	0.949
RANDOM FOREST	0.979	0.949	0.95	0.949
NEURAL NETWORK	0.974	0.944	0.946	0.945
ADAPTIVE BOOSTING	0.914	0.9923	0.924	0.923
DECISION TREE	0.913	0.941	0.941	0.941
KNN	0.927	0.909	0.914	0.911
SVM	0.515	0.584	0.585	0.584
LOGISTIC REGRESSION	0.835	0.781	0.791	0.795
NAÏVE BAYES	0.931	0.865	0.875	0.861

Source: Author's calculations using Orange software

It can be interpreted that among the implemented models, the non-linear ones have had a better performance. Both 'Gradient Boosting' and' Random Forest' models are ensemble models. In other words, they are built by combining two or more existing models. In general terms, ensembled models are expected to have better performance and higher accuracy rates. The result of this thesis acknowledges the same claim.

CONCLUSION

In previous years, P2P lending companies have seen an increase in the number of loan applications to the point where errors in the risk prediction of the loan have resulted in massive revenue losses. Therefore, the process of assigning a loan to a candidate has become more crucial than ever. Existing models for risk prediction, i.e. predicting whether or not a loan applicant will be able to repay the defaults according to the fixed schedule, have performed admirably.

Algorithms for machine learning are fully automated models with a greater predictive capacity. This paper seeks to determine whether Machine Learning algorithms are capable of predicting the level of riskiness associated with assigning a loan to a specific applicant by categorizing the borrowers as high risk and low risk. If so, which of the available Machine Learning models will have the best performance? The research was limited to Bondora, a P2P lending company actively expanding in Estonia, Finland, and Spain, among other EU nations. Due to the large number of records available in their public report, the statistical validity of this study's findings can be assured.

Previous research on the implementation of machine learning algorithms in the financial sector has demonstrated that these models can be surprisingly useful in a variety of fields, including banking and peer-to-peer lending. In addition, for the purpose of loan risk prediction, previous studies have presented Machine Learning algorithms that are widely employed today. In addition to implementing existing models, an encapsulated model was proposed. This model was created by combining two existing classification models, resulting in enhanced accuracy and performance. Despite the fact that extensive research had been conducted on the task of loan risk prediction, a wider variety of models had not been tested for comparison purposes. In addition, there is no assurance that the same successful model for one dataset will be successful for all datasets. A single algorithm may be the most efficient for one dataset and the least efficient for another.

Compared to traditional models, Machine Learning algorithms have a number of benefits. They can easily manage big and complicated datasets, find subtle patterns and links in the data, and modify their predictions in response to new data. However, compared to traditional models, they can be more difficult to execute and need for greater computing power. Additionally, it can be more complicated for banks and other financial institutions like peer to peer lending companies to comply with regulatory standards due to the complexity of machine learning models.

This study implemented nine Machine Learning models and compared their accuracy level of each to another. As a result, it was seen that not only Machine Learning models can be implemented for risk prediction purposes, but also these models do a better job than classical models. Also, after putting the results in order, it became clear that Random Forest and Gradient Boosting models had the highest accuracy level among all other implemented models.

Based on the findings of his study, it is highly recommened to P2P lending companies to make use of Machine Learning algorithms for the risk prediction tasks. The main reason supporting this suggestion is higher performance or in other words, less errors of these models. It is predictable that by utilizing a Random Forest model, a P2P company can acquire higher returns and consequently, a higher growth. However the complex process of implementation and preparation of these models increases the possibility of faulty results in calculations. In a nutshell, higher accuracy may be gained by these models, but complexity of the models could hinder companies from welcoming them.

As concluded by this study, Ensemble Machine Learning algorithms are potential models that may result in higher accuracy. Future studies may focus more on different types of ensemble Machine Learning models and investigate among their accuracy. Moreover, more models from other fields of artificial intelligence, such as Reinforcement Learning algorithms could be implied to automate the process of implementing and monitoring the performance of the disgussed Machine Learning models in companies.

LIST OF REFERENCES

Aggarwal, C. C. (2018). Neural networks and deep learning. Springer, 10(978), 3.

Bandyopadhyay, A. (2016). Managing portfolio credit risk in banks. Cambridge University Press.

Bayraktar, M., Aktaş, M. S., Kalıpsız, O., Susuz, O., & Bayracı, S. (2018, May). Credit risk analysis with classification Restricted Boltzmann Machine. In 2018 26th Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE.

Bonaccorso, G. (2017). Machine learning algorithms. Packt Publishing Ltd.

Charbuty, B., & Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning. Journal of Applied Science and Technology Trends, 2(01), 20-28.

Dreiseitl, S., & Ohno-Machado, L. (2002). Logistic regression and artificial neural network classification models: a methodology review. Journal of biomedical informatics, 35(5-6), 352-359.

Duchamp, T. (2016). Big Data is the Cornerstone of Regulatory Compliance Systems. The FinTech Book: The Financial Technology Handbook for Investors, Entrepreneurs and Visionaries, 100-105.

Fawcett, T. (2006). An introduction to ROC analysis. Pattern recognition letters, 27(8), 861-874.

Friedman, J. H. (2002). Stochastic gradient boosting. Computational statistics & data analysis, 38(4), 367-378.

Gahlaut, A., & Singh, P. K. (2017, July). Prediction analysis of risky credit using Data mining classification models. In 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-7). IEEE.

Han, J., Kamber, M., & Pei, J. (2001). Data mining concepts and techniques, Morgan Kaufmann Publishers. San Francisco, CA, 335-391.

Hassan, A. K. I., & Abraham, A. (2013, August). Modeling consumer loan default prediction using ensemble neural networks. In 2013 International Conference on Computing, Electrical and Electronic Engineering (ICCEEE) (pp. 719-724). IEEE.

Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). The elements of statistical learning: data mining, inference, and prediction (Vol. 2, pp. 1-758). New York: springer.

Hernán, M. A., Hsu, J., & Healy, B. (2019). A second chance to get causal inference right: a classification of data science tasks. Chance, 32(1), 42-49.

Hsueh, S. C., & Kuo, C. H. (2017, August). Effective matching for P2P lending by mining strong association rules. In Proceedings of the 3rd International Conference on Industrial and Business Engineering (pp. 30-33).

Isik, F. M., Tastan, B., & Yolum, P. (2007, April). Automatic adaptation of BPEL processes using semantic rules: design and development of a loan approval system. In 2007 IEEE 23rd International Conference on Data Engineering Workshop (pp. 944-951). IEEE.

Jiang, T., Gradus, J. L., & Rosellini, A. J. (2020). Supervised machine learning: a brief primer. Behavior Therapy, 51(5), 675-687.

Jin, Y., & Zhu, Y. (2015, April). A data-driven approach to predict default risk of loan for online peer-to-peer (P2P) lending. In 2015 Fifth international conference on communication systems and network technologies (pp. 609-613). IEEE.

Kleinbaum, D. G., Dietz, K., Gail, M., Klein, M., & Klein, M. (2002). Logistic regression (p. 536). New York: Springer-Verlag.

Kothari, A. A., & Patel, W. D. (2015). A novel approach towards context based recommendations using support vector machine methodology. Procedia Computer Science, 57, 1171-1178.

Lee, I., & Shin, Y. J. (2018). Fintech: Ecosystem, business models, investment decisions, and challenges. Business horizons, 61(1), 35-46.

Leong, C., Tan, B., Xiao, X., Tan, F. T. C., & Sun, Y. (2017). Nurturing a FinTech ecosystem: The case of a youth microloan startup in China. International Journal of Information Management, 37(2), 92-97.

Lin, W., Wu, Z., Lin, L., Wen, A., & Li, J. (2017). An ensemble random forest algorithm for insurance big data analysis. Ieee access, 5, 16568-16575.

Li, S. T., Shiue, W., & Huang, M. H. (2006). The evaluation of consumer loans using support vector machines. Expert Systems with Applications, 30(4), 772-782.

Maalouf, M. (2011). Logistic regression in data analysis: an overview. International Journal of Data Analysis Techniques and Strategies, 3(3), 281-299.

Malhotra, R., & Malhotra, D. K. (2003). Evaluating consumer loans using neural networks. Omega, 31(2), 83-96.

Margineantu, D. D., & Dietterich, T. G. (1997, July). Pruning adaptive boosting. In ICML (Vol. 97, pp. 211-218).

Mathur, P. (2018). Machine learning applications using python: Cases studies from healthcare, retail, and finance. Apress.

Mitchell, T. M. (1997). Artificial neural networks. Machine learning, 45(81), 127.

Mueller, J. P., & Massaron, L. (2021). Machine learning for dummies. John Wiley & Sons.

Müller, A. C., & Guido, S. (2016). Introduction to machine learning with Python: a guide for data scientists. " O'Reilly Media, Inc.".

Oussous, A., Benjelloun, F. Z., Lahcen, A. A., & Belfkih, S. (2018). Big Data technologies: A survey. Journal of King Saud University-Computer and Information Sciences, 30(4), 431-448.

Pandey, U. K., & Pal, S. (2011). Data Mining: A prediction of performer or underperformer using classification. arXiv preprint arXiv:1104.4163.

Pandimurugan, V., Usha, D., Guptha, M. N., & Hema, M. S. (2022). Random forest tree classification algorithm for predicating loan. Materials Today: Proceedings, 57, 2216-2222.

Rahm, E., & Do, H. H. (2000). Data cleaning: Problems and current approaches. IEEE Data Eng. Bull., 23(4), 3-13.

Ramakrishnan, S., Mirzaei, M., & Bekri, M. (2015). Adaboost ensemble classifiers for corporate default prediction. Research Journal of Applied Sciences, Engineering and Technology, 9(3), 224-230.

Raschka, S., & Mirjalili, V. (2017). Python machine learning: Machine learning and deep learning with python. Scikit-Learn, and TensorFlow. Second edition ed, 3.

Rish, I. (2001, August). An empirical study of the naive Bayes classifier. In IJCAI 2001 workshop on empirical methods in artificial intelligence (Vol. 3, No. 22, pp. 41-46).

Russell, R. (2020). Machine Learning: Step-by-Step Guide To Implement Machine Learning Algorithms with Python. (Knxb).

Schapire, R. E. (1990). The strength of weak learnability. Machine learning, 5, 197-227.

Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge university press.

Shobha, G., & Rangaswamy, S. (2018). Chapter 8-Machine Learning Handbook of Statistics. Elsevier.

Sun, J., Fujita, H., Chen, P., & Li, H. (2017). Dynamic financial distress prediction with concept drift based on time weighting combined with Adaboost support vector machine ensemble. Knowledge-Based Systems, 120, 4-14.

Suryono, R. R., Purwandari, B., & Budi, I. (2019). Peer to peer (P2P) lending problems and potential solutions: A systematic literature review. Procedia Computer Science, 161, 204-214.

Zhang, H. (2004). The optimality of naive Bayes. Aa, 1(2), 3.

APPENDICES

Appendix 1. Codes and Outputs

The link below contains the script in python programming language that was compiled by the author for this thesis.
https://drive.google.com/file/d/1UB3uUsm2iYFKTgw4U5bVFG1ul8Qbfp3m/view?usp=share_link

The codes were compiled in Jupyther notebook platform. Therefore, the file format is. ipynb.

```
import numpy as np
import pandas as pd
from pathlib import Path
from collections import Counter
iris = pd.read_csv('/Users/Sinay/Desktop/LoanData.csv')
columns=['ReportAsOfEOD','LoanId','LoanNumber','ListedOnUTC','BiddingStartedOn','PartyId','LoanApplicationStartedDate','LoanDate','ContractEndDate','FirstPaymentDate','MaturityDate_Original','MaturityDate_Last','ApplicationSignedHour','ApplicationSignedWeekday','LanguageCode','DateOfBirth','AppliedAmount','LoanDuration','City','County','NrOfDependants','EmploymentPosition','IncomeOther','IncomeFromSocialWelfare','IncomeFromPrincipalEmployer','IncomeFromPension','IncomeFromLeavePay','IncomeFromFamilyAllowance','IncomeFromChildSupport','ExistingLiabilities','MonthlyPaymentDay','LastPaymentOn','CurrentDebtDaysPrimary','DebtOccuredOn','CurrentDebtDaysSecondary','DebtOccuredOnForSecondary','EAD1','EAD2','ModelVersion','Rating','EL_V0','Rating_V0','Rating_V1','Rating_V2','EL_V1','ActiveLateCategory','WorseLateCategory','AmountOfPreviousLoansBeforeLoan','PreviousRepaymentsBeforeLoan','PreviousEarlyRepaymentsBefoleLoan','GracePeriodEnd','GracePeriodStart','NextPaymentDate','NextPaymentNr','NrOfScheduledPayments','ReScheduledOn','PrincipalDebtServicingCost','ActiveLateLastPaymentCategory','InterestAndPenaltyBalance','CreditScoreFiAsiakasTietoRiskGrade','Credit
ScoreEsEquifaxRisk','CreditScoreEeMini','StageActiveSince','RecoveryStage','PlannedPrincipalPostDefault','PlannedInterestPostDefault','PrincipalRecovery','InterestRecovery','DefaultDate','PlannedInterestTillDate','PlannedPrincipalTillDate','WorkExperience','InterestAndPenaltyDebtServicingCost','InterestAndPenaltyWriteOffs','PrincipalWriteOffs','BidsPortfolioManager','BidsApi','BidsManual','Country','Interest','ActiveScheduleFirstPaymentReached','ExpectedLoss','LossGivenDefault','ExpectedReturn','ProbabilityOfDefault','PrincipalOverdueBySchedule','Restructured','CreditScoreEsMicroL','InterestAndPenaltyWriteOffs']
iris.info()
from sklearn.metrics import balanced_accuracy_score
from sklearn.metrics import confusion_matrix
from imblearn.metrics import classification_report_imbalanced
target = ['Status']
x = {'Late': 'high_risk'}
iris = iris.replace(x)
x = dict.fromkeys(['Current','Repaid'], 'low_risk')
iris = iris.replace(x)
iris.reset_index(inplace=True, drop=True)
iris.info()
iris_encoded=pd.get_dummies(iris,columns=['NewCreditCustomer','Age','UseOfLoan','EmploymentDurationCurrentEmployer','RefinanceLiabilities'])
X = iris_encoded.drop(['Status'], axis = 1)
y = iris_encoded["Status"]
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state = 1)
labelList = []
resultList = []
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression(random_state = 0, max_iter=50000)
logreg.fit(X_train, y_train)
labelList.append("Logistic Regression")
```

```
resultList.append(logreg.score(X_test,y_test))
from imblearn.ensemble import BalancedRandomForestClassifier
rf_model = BalancedRandomForestClassifier(n_estimators = 100, random_state = 1)
rf_model.fit(X_train, y_train)
labelList.append("Random Forest")
resultList.append(rf_model.score(X_test,y_test))
from sklearn.tree import DecisionTreeClassifier
tree = DecisionTreeClassifier(max_depth=5, random_state=0)
tree.fit(X_train, y_train)
labelList.append("Decision Tree")
resultList.append(tree.score(X_test,y_test))
from sklearn.neural_network import MLPClassifier
mlp = MLPClassifier(random_state=1)
mlp.fit(X_train, y_train)
labelList.append("Neural Networks")
resultList.append(mlp.score(X_test,y_test))
from sklearn.naive_bayes import GaussianNB
nb = GaussianNB()
nb.fit(X_train,y_train)
labelList.append("Naive Bayes")
resultList.append(nb.score(X_test,y_test))
from sklearn.svm import SVC
svm = SVC(random_state=3)
svm.fit(X_train,y_train)
labelList.append("Support Vector Machine")
resultList.append(svm.score(X_test,y_test))
from imblearn.ensemble import EasyEnsembleClassifier
ecc_model = EasyEnsembleClassifier(n_estimators = 100, random_state = 1)
ecc_model.fit(X_train, y_train)
labelList.append("Adaboost")
resultList.append(ecc_model.score(X_test,y_test))
from sklearn.neighbors import KNeighborsClassifier
import matplotlib.pylab as plt
score_list = []
```

```
for each in range(1,15):
    knn2 = KNeighborsClassifier(n_neighbors = each)
    knn2.fit(X_train,y_train)
    score_list.append(knn2.score(X_test,y_test))
plt.plot(range(1,15),score_list)
plt.xlabel("k values")
plt.ylabel("accuracy")
plt.show()
a = max(score_list)
b = score_list.index(a)+1
print("k = ",b," and maximum value is ", a)
labelList.append("KNN")
resultList.append(a)
plt.plot(labelList,resultList)
plt.show()
zipped = zip(labelList, resultList)
zipped = list(zipped)
df = pd.DataFrame(zipped, columns=['label','result'])
new_index = (df['result'].sort_values(ascending=False)).index.values
sorted_data = df.reindex(new_index)
plt.plot(sorted_data.loc[:,"label"],sorted_data.loc[:,"result"])
plt.show()
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = (20,3)
new_index = (df['result'].sort_values(ascending=False)).index.values
sorted_data = df.reindex(new_index)
plt.plot(sorted_data.loc[:,"label"],sorted_data.loc[:,"result"])
plt.show()
sorted_data
```

Appendix 2. Orange File

For better user experience and more visualization options, Orange platform is used, taking the same approach in python scripts. In this file, step by step progress of the algorithms and the overall picture of the study can be demonstrated.

The link below contains the Orange file in .ows format.

https://drive.google.com/file/d/1fHzHQrtxqMzpQX42_MrkTciTE5HJtI0M/view?usp=share_link

Appendix 3. Raw Data

The below link contains the Raw data that was used for this study. The raw data was collected from Bondora website.

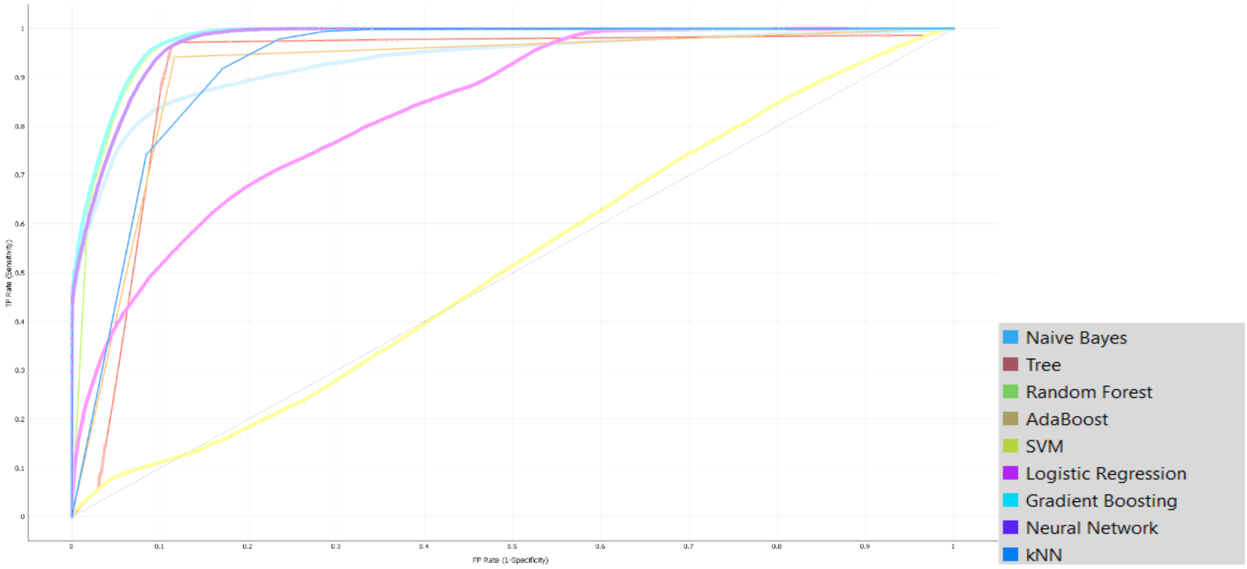
https://www.bondora.com/marketing/media/LoanData.zip

Moreover, a brief description of the raw dataset and its structure can be found in the below link.

https://www.bondora.com/en/public-reports

Appendix 4. Model Output and performance

AUC analysis for applied models are as follows.



Source: Compiled by author using Orange software

The table below demonstrates the confusin matrix of models performed in prediction of loans to be either low risk or high risk.

	TP	FP	FN	TN
GRADIENT BOOSTING	94.7%	5.3%	4.7%	95.3%
RANDOM FOREST	94.7%	5.3%	4.4%	95.6%
NEURAL NETWORK	93.8%	6.2%	3.6%	96.4%
ADAPTIVE BOOSTING	94.7%	5.3%	12.7%	87.3%
DECISION TREE	94.7%	5.3%	7.2%	92.8%
KNN	90.2%	9.8%	5.9%	94.1%
SVM	69.8%	30.2%	66.4%	33.6%
LOGISTIC REGRESSION	80.3%	19.7%	23.4%	76.6%
NAÏVE BAYES	94.1%	5.9%	26.9%	73.1%

Source: Compiled by author using Orange software

Appendix 5. Non-exclusive licence

A non-exclusive licence for reproduction and publication of a graduation thesis[11]

I, Sina Ansari Fard

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis
RISK PREDICTION FOR LOAN APPLICATIONS BY MACHINE LEARNING ALGORITHMS

supervised by Pavlo Illiashenko, ME,

1.1 to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;

1.2 to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.

2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.[i]

3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

08.05.2023

[1] *The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.*