NOVA IMS Information Management School

MEGI

Mestrado em Estatística e Gestão de Informação
Master Program in Statistics and Information Management

Credit Risk Scoring: A Stacking Generalization Approach

Bernardo Dias Raimundo

Dissertation presented as a partial requirement for obtaining a Master's degree in Statistics and Information Management

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

2022 Title: Credit Risk Scoring: A Stacking Generalization Approach Bernardo Dias Raimundo MEGI

2022 Title: Credit Risk Scoring: A Stacking Generalization Approach Bernardo Dias Raimundo MGI

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

Credit Risk Scoring: A Stacking Generalization Approach

by

Bernardo Dias Raimundo

Dissertation presented as a partial requirement for obtaining a Master's degree in Statistics and Information Management, with a specialization in Risk Analysis and Management.

Advisor: Prof. Dr. Jorge Miguel Ventura Bravo

October 2022

DEDICATION

This dissertation is devoted to the memory of my grandpa, Mário Dias.

ACKNOWLEDGEMENTS

The realization of this dissertation would not have been possible without the help, support, and guidance of many people. I would like to extend my sincere gratitude to all of them.

Foremost, I would like to express my gratitude to Prof. Doctor Jorge Bravo who made this dissertation possible. His advice, expertise, and continuous support were essential through all stages of this journey.

To my parents, for providing me the opportunity to follow my dreams and supporting my decisions, which would not have been possible without their love and motivation.

To my brother Rodrigo, for not only being the best brother I could have asked for but also for his ability to simplify things, even when he did not understand the subject matter.

To my friend Fátima, for the moments of laughter, joy, and distraction that were essential throughout these years and for always listening to my complaints.

To Nova Information Management School, for receiving me with open arms, embracing my goals, and being my second home.

Thank you all!

ABSTRACT

Credit risk regulation has been receiving tremendous attention, as a result of the effects of the latest global financial crisis. According to the developments made in the Internal Rating Based approach, under the Basel guidelines, banks are allowed to use internal risk measures as key drivers to assess the possibility to grant a loan to an applicant. Credit scoring is a statistical approach used for evaluating potential loan applications in both financial and banking institutions. When applying for a loan, an applicant must fill out an application form detailing its characteristics (e.g., income, marital status, and loan purpose) that will serve as contributions to a credit scoring model which produces a score that is used to determine whether a loan should be granted or not. This enables faster and consistent credit approvals and the reduction of bad debt. Currently, many machine learning and statistical approaches such as logistic regression and tree-based algorithms have been used individually for credit scoring models. Newer contemporary machine learning techniques can outperform classic methods by simply combining models.

This dissertation intends to be an empirical study on a publicly available bank loan dataset to study banking loan default, using ensemble-based techniques to increase model robustness and predictive power. The proposed ensemble method is based on stacking generalization an extension of various preceding studies that used different techniques to further enhance the model predictive capabilities. The results show that combining different models provides a great deal of flexibility to credit scoring models.

KEYWORDS

Credit scoring; Ensemble learning; Probability of default; Stacking generalization; Risk management

INDEX

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND ACRONYMS

AUC	Area Under the Curve
DT	Decision Tree
DTI	Debt to Income Ratio
FN	False Negative
FP	False Positive
KNN	K Nearest Neighbors
LR	Logistic Regression
NN	Neural Networks
PCA	Principal Component Analysis
ROC	Receiving Operating Characteristics Curve
SGD	Stochastic Gradient Descent
SMOTE	Synthetic Minority Oversampling Technique
SVM	Support Vector Machine
TN	True Negative
TP	True Positive

1. INTRODUCTION

Banks play a major role in all economic and financial activities in modern society. In recent years, finance officers and bankers have faced numerous challenges primarily related to the complexity of financial markets, because of the rapid economic, social, and technological changes. In this regard, possessing knowledge of such complicated financial environments forces financiers to examine and find appropriate tools for measuring losses. The success of a bank largely depends on the quality of its loans and advances, and thus proper credit risk management is essential. In the words of Saunders and Cornett (2008), credit risk is the "risk that the promised cash flows from loans and securities held by financial institutions may not be paid in full". Therefore, there is uncertainty over the borrower's performance in the future. The current volatile, political, and economic setting demands greater management expertise in the credit environment to contribute to the financial performance of banks as the viability of a bank, is heavily influenced by the growth of non-performing loans, due to poor risk management practices.

The critical role in the lending market caused by the major uproar of the latest global financial crisis led to an increase in banking regulation and academic research in credit risk modeling. The regulatory changes in the banking framework brought by the revised Basel Accords (subsequently adopted by many countries and regions) introduced stronger risk management obligations for banks. Encouraged by regulators, banks now devote significant resources to develop internal credit risk models to better support decisions when granting loans, quantify expected credit losses, and assign mandatory economic capital (Chamboko & Bravo, 2016, 2020). To do this, four risk components are required: (i) Probability of Default, which measures the likelihood that a loan will not be repaid; (ii) Exposure at Default, the expected value of the loan at the time of default; (iii) Loss Given Default, which represents the amount of loss if the applicant defaults expressed as a percentage of the Exposure at Default; and (iv) Effective Maturity, which signifies the maturity of the exposures.

Thus, the primary problem of any lender is to distinguish "good" from bad" debtors before granting credit. To distinguish between them, a new concept was introduced in the credit risk literature known as credit scoring. Credit scoring is a statistical procedure that involves the usage of historical data and statistical techniques to rank applicants according to a score that reflects their creditworthiness (Mester, 1997). Before the availability of high computational systems and the introduction of machine learning models, credit analysis used a pure judgmental approach to accept or reject an application, known as the five Cs of credit, composed of the following metrics: (i) Character, measures the applicant character and integrity (e.g., reputation, honesty); (ii) Capital, measures the difference between the applicant's assets and liabilities; (iii) Collateral, measures the collateral provided in case payment problems occur; (iv) Capacity, measures the applicant's ability to pay (e.g., job status, income); and Condition, measures the applicant's circumstances (e.g., market conditions, competitive pressure) (Thomas, 2000). Due to its subjectivity, this approach is difficult to standardize, which could result in inconsistent measurements (Saunders & Allen, 2002).

Vojtek and Kocenda (2006) provide an outline of indicators that are typically important for credit scoring models. The authors divide these indicators into four distinct categories: demographic, financial, employment, and behavioral. They also supply an overview of the most relevant credit scoring methods used to predict default while also discussing problems that arise from implementing these procedures.

Developing credit scoring models can be time-consuming and resource-heavy meaning it is common that most institutions use the same credit scoring models for several years. Consumer behavior was altered and their ability to repay the debt was conditioned and many of these traditional systems are not prepared to deal with these changes, becoming severely outdated (Sousa, Gama, & Brandão, 2015).

The standard approach to credit risk modeling is to pursue a "winner-take-all" perspective by which, for each dataset, a single believed to be the "best," or "true" model is selected from a set of candidate approaches using some method or criteria often neglecting model uncertainty (conceptual uncertainty) for statistical inference purposes. The use of different lookback periods, diverse selection procedures, alternative accuracy metrics, misspecification problems, and the presence of structural breaks in the data generating process can lead to different model choices and predictive accuracy (Bravo et al., 2021; Bravo, 2021; Bravo & Ayuso, 2020, 2021; Ayuso et al., 2021).

To tackle this limitation, model combination (also known as an ensemble) has been gaining a lot of traction as a more contemporary approach to predicting default. Ensemble learning is a technique that combines two or more algorithms with different prediction capabilities to make more accurate predictions (Chopra & Bhilare, 2018). This allows for a more robust model since it includes the predictions from all these weak learners to get the best result while avoiding any potential preference.

The widely used ensemble methods are bagging (Breiman, 1996), boosting (Schapire, 1990; Freund, 1995), and stacking (Wolpert, 1992). Bagging and boosting, who consider homogenous weak learners, combine the results of multiple models to get a generalized result. Bagging is an application of the bootstrap procedure (Efron & Tibshirani, 1993), where randomly sampled datasets are produced from the original data taking the average of these predictions to make a final prediction. Boosting is a sequential procedure; it follows a sequential ensemble technique where each subsequent model corrects the error of the earlier model. By identifying these misclassifications in previous iterations, more weight is given to them thus in the next iteration the learner will focus more on these misclassifications. Stacking considers heterogeneous weak learners, combining models of diverse types. The architecture of a stacking ensemble involves two or more base models, often referred to as level-0 models with different weight combinations, and a meta-model that combines the predictions of these level-0 models to make a complete final prediction.

The main contribution of this dissertation lies in comparing the performance of individual models against stacking generalization to predict default. Performance metrics such as accuracy, f1-score, precision, and recall will be used to compare individual classifiers and the ensemble approach. In addition to these metrics and to get a better grasp of our data confusion matrices, the receiver operating characteristic curves (ROC curve) and area under each curve (AUC Score) are also used to further enhance this analysis. Four different algorithms will be considered: (i) K Nearest Neighbors (KNN), (ii) Support Vector Machine (SVM), and (iii) Decision Tree (DT) as level-0 models and use those predictions as inputs for the (iv) Logistic Regression (LR), the meta-model.

As with any research, the comparison between single classifiers and an ensemble approach to predict default is assessed through the first research question. The second question emphasizes the number of models that should be considered to achieve better results:

Research question #1:

> How does an ensemble learning approach compare against more traditional models when predicting the probability of default?

Research question #2:

> Given the ensemble-based approach from question one, how sensitive is the outcome of the ensemble to different model combinations?

Financial stability is vital for any economy; hence banks need to be effectively managed. In an increasingly complex and global environment of growing demands for trusted information, fast-developing, and accessible technologies financial institutions increasingly rely on artificial intelligence to develop credit scoring models that better support the decision-making process, allow for more precise risk assessments and better customer and user experience. Traditionally, the LR and DT algorithms are regarded as industry standards when it comes to predicting default risk of an applicant. Studies on credit scoring applications using stacking ensemble are far more infrequent in the relevant literature. The novelty of this dissertation lies in not only, to contribute to the ongoing experimental findings in the credit risk subject, but also serve as a baseline for implementing an ensemble combination to predict default risk.

Regarding the dissertation outline, the first section introduces the concept of credit risk modeling and highlights some of the most notable statistical techniques. The second section will focus on presenting some of the many contributions to the credit risk literature, concluding with a summarized table of additional more recent studies on this matter. The third section will start by introducing and discussing the chosen models and present an in-depth look at the most popular ensemble techniques and the performance metrics considered for evaluation. The fourth section will feature a deep understating of the data, preprocessing steps applied, and the sampling strategy used. The fifth section will start by presenting the results of the individual classifiers and then a comparison with the results of the stacking approach. The concluding section will present some final remarks and recommendations for future work. Bibliographic references and annexes are included at the end of this dissertation.

2. LITERATURE REVIEW

This section provides a brief overview of the main contributions in the credit risk literature by presenting different credit scoring applications while discussing their results.

2.1. A LOOKBACK AT PARALLEL STUDIES

The literature on credit risk modeling has grown extensively since the initial work of Beaver (1966) and Altman (1968). Using univariate analysis on thirty different financial ratios, under different criteria, Beaver (1966) concluded that the value of certain ratios varied between healthy financial institutions and the ones that presented financial difficulties. Altman (1968) published a model that predicts bankruptcy by combining different observable characteristics that may help to distinguish default from non-default firms. This credit scoring model developed by Altman known as the "Z Score Model," proposes a model using five different financial ratios to predict bankruptcy by assigning the highest Z-score to the obligor. Altman concluded that companies with a Z-score lower than 1.81 fall into the default category whereas a Z-score that is higher than 2.99 corresponds to the non-default category. Values between this interval fall in the "zone of ignorance" where the author considers that the value that best separates both the default and non-default categories is 2.675.

Since these initial approaches, several techniques have been developed to further help decision-makers and financial analysts in predicting default by considering both traditional statistical methods and more sophisticated modeling techniques (Ashofteh & Bravo, 2021). The most well-known traditional models are regression models. Some noteworthy studies in regression models include logit models (see, e.g., Martin, 1977; Ohlson, 1980, and Zavgren, 1985) and probit models (see, e.g., Zmijewski, 1984 and Skogsvik, 1990).

Although these traditional models are the most preferred, especially those that are based on the logit model, for estimating the probability of default some concerns need to be raised. According to Bartual et al. (2012) professionals in the field should pay close attention to robustness problems that can arise when using the logit model by considering three factors: (i) the choice of variables to be used in the model; (ii) the influence of the sample in the model results and (iii) the cutoff point as it may influence the percentage of correct and incorrect predictions.

In the meantime, advancements made in credit risk modeling allowed for the development of newer and more contemporary machine learning models to assess credit risk amid advances in computer technology. Since credit risk analysis is similar to pattern-recognition problems, algorithms can be used to classify the creditworthiness of counterparties (Barboza, Kimura, & Altman 2017).

Lee and Chen (2003) performed credit scoring tasks using the DT algorithm and multivariate adaptive regression splines. The authors concluded that these algorithms were able to outperform discriminant analysis, LR, neural networks (NN), and SVM by reaching a superior classification rate. Zhu et al. (2019) assessed the performance of the random forest algorithm against the LR, DT and SVM highlighting the strong ability of generalization of the random forest.

Despite the usage of single classifiers, it is hard to overstate the importance of model uncertainty for economic modeling. The empirical work in economics and social modeling is subject to a large amount of uncertainty about the model specification. Steel (2020) provides a summary of the types of

uncertainty that need to be considered when modeling: (i) Theory uncertainty, i.e., the lack of a universally accepted theory that has been empirically verified as a near-perfect explanation of reality; (ii) Specification uncertainty, i.e., the different ways in which theories can be implemented in empirical models; and (iii) Heterogeneity, i.e., uncertainty and independence of the observables.

When presented with multiple candidate models or algorithms picking one model that can give optimal performance for future data can result in unstable or useless information. An alternative to this scenario is model averaging, which tries to find an optimal model combination of all individual models using some method or criteria. (Yao et al., 2017).

The concept of combining several candidate models is known as ensemble learning, and many techniques have been suggested under a variety of frameworks. Both staking and bayesian model averaging (Madigan & Raftery, 1994) use the concept of combining candidate models with carefully chosen candidate model weights. In stacking, weights are chosen to minimize the sum of the squares of the distances between the response variable values and a linear combination of predicted values obtained by each candidate model (Lee & Song, 2021). On the other hand, bayesian model averaging takes advantage of the concept of posterior probability to form a weighted average of a class of models, based on weights that depend on the relative likelihood of each model, using approximated bayesian information criterion (Hayden, Stomper & Westerkamp, 2010). Bravo and Mekkaoui (2022) applied a flexible bayesian model averaging approach to consumer price index inflation models to mitigate conceptual uncertainty and increase predictive accuracy. To select the best candidate models, the authors adopted the model confidence set approach (Hansen, Lunde & Nason, 2011) in which a sequence of tests allows the construction of a set of "superior" models, at a specified confidence level.

Tripathi et al. (2018) proposed a cluster-based feature selection with an ensemble classifier model to avoid redundant variables as these may degrade the predictive power of a credit scoring model. The heterogenous ensemble consisted of five different models evaluated on three different datasets. The result showed that the clustered feature selection approach improved the model and therefore outperformed the existing models. Furthermore, Wang et al. (2011) analyzed different credit risk databases using ensemble methods (bagging, boosting, and stacking) coupled with base learners (LR, DT, artificial NN, and SVM) to find that bagging presented the best results. Tian et al. (2020) proposed the use of a gradient boosting DT against more traditional credit scoring models highlighting its improvements in both accuracy and recall rate.

Several studies have dealt with the application of machine learning algorithms in credit risk modeling. Academics and practitioners are exploring more complex techniques since results regarding the superiority of models are still inconclusive. With the advancement made in this subject, data scraping will allow the observation of new information that may result in relevant inputs to machine learning models and lead to different and even more valuable results.

2.2. RECENT APPLICATIONS OF CREDIT RISK MODELING

Table 1 presents some of the more recent studies about the application of machine learning models in credit scoring applications. The table includes the year of publication, authors names, the data used in each publication, and models that served as the baseline for modeling credit risk.

Year	Author	Dataset	Models
2022	Chang, Yang, Tsaih, & Lin	Lending Club	Logistic Regression, Support Vector Machine, Decision Tree, Random Forest, XGBoost, LightGBM and Artificial Neural Network
2021	Turjo, Rahman, Karim, Biswas, Dewan & Hossain	Lending Club	Logistic Regression, K Nearest Neighbor, Gradient Boosting, XGBoost, Artificial Neural Network
2021	Tahmid, Haque, Faruque, Keya, Khushbu, & Marouf	Data Warehouse	Logistic Regression, Gaussian Naïve Bayes, Bernoulli Naïve Bayes, Random Forest, Support Vector Machine, Decision Tree, Artificial Neural Network
2021	Khanh, Duong, Quang-Linh, Ân, Nguyen & Nguyen	Kalapa Credit Score Dataset	LightGBM, CatBoost, and Random Forest
2021	Wu & Pan	Lending Club	Logistic Regression, Random Forest, Support Vector Machine
2020	Trivedi	German Credit Risk Dataset	Naïve Bayes, Support Vector Machine, Random Forest, and Decision Tree
2019	İlter, Kocadağli & Ravishanker	Irish Dummy Banks Dataset	Logistic Regression, Recursive Partitioning, Random Forest, Conditional Inference Trees, Support Vector Machine, and Least Absolute Shrinkage Selection Operator
2019	Kim & Cho	Lending Club	Convolutional Neural Network, Deep Learning

Table 1: Summary of recent applications regarding credit risk modeling
Source: Author preparation

3. METHODOLOGY

This section will provide a discussion of the models used in this dissertation while also providing an overview of different ensemble techniques and the performance metrics used for evaluation.

3.1. PREDICTION MODELS

3.1.1. LOGISTIC REGRESSION

The LR (Cox, 1958) is a parametric method that is very appealing for credit risk assessments among financial institutions. It analyzes the relationship between multiple independent variables and a categorical dependent variable and estimates the probability of occurrence of a certain event by fitting the data in a logistic curve (Awang & Alimin, 2016).

LR is regarded as an industry standard and is widely applied in practice because of its simplicity and balanced error distribution. Anderson (2007) draws attention to its fairly robust estimate of the actual probability, given the available information and the final probability cannot fall outside of the range 0 to 1.

A link function, known as logit, the natural logarithm of the odds, is used to establish a linear function with the input variables. With this link function, a LR manages to model a nonlinear relationship between the input variables and the dependent variable in a linear way.

The LR is given by the following equation (1):

$$ln\left(\frac{p}{1-p}\right) = \beta_0 + \sum_{i=1}^{n} \beta_i x_i, \tag{1}$$

where p denotes the probability prediction, β_0 represents the constant coefficient, and β_i the coefficient corresponding to the feature x_i. In other words, for a certain input, LR outputs the conditional probability of a sample belonging to a specific class.

The probability of default can be expressed as follows:

$$P(y = 1|x_1, \dots x_n) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \dots + \beta_n x_n)}}, \tag{2}$$

where n represents the number of independent variables.

For non-default applicants, the formula is simply given by $1 - P(y = 1|x_1, \dots x_n) = P(y = 0|x_1, \dots x_n)$.

LR models are fitted via maximum likelihood estimation which involves maximizing a likelihood function to find the probability distribution and parameters that best explain the data. In other words, during the fitting process, this method will estimate the coefficients in such a way it will maximize the probability of labeling an applicant as default as well as maximize the probability of labeling an applicant as non-default.

The LR model was trained using stochastic gradient descent (SGD), a type of optimization algorithm that is widely used to solve machine learning algorithm model parameters. Through continuous iteration, it obtains the gradient of the objective function, gradually approaches the optimal solution of the objective function, and finally obtains the minimum loss function and related parameters (Wang, Yan, & Zhang, 2021). It only uses one example of the training set for each iteration making it easier to fit into memory while also being computationally fast as only one sample is processed at a time.

When performing hyperparameter tunning, two parameters were taken into consideration: a regularization technique and the model alpha. In terms of regularization techniques, two types were evaluated: ridge regression and lasso regression.

The ridge regression adds a square magnitude of the coefficient as the penalty to the loss function. According to James et al. (2013), the ridge regression coefficient estimates $\hat{\beta}^R$ are the values that minimize:

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j\, x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}{\beta_j}^2, \tag{3}$$

where $\lambda \geq 0$ is a tunning parameter. When $\lambda = 0$ the ridge regression will produce least squares estimates. If the value of λ is exceptionally large, it will add too much weight, the ridge regression coefficients estimates will move towards zero, leading to underfitting.

The lasso regression is a recent alternative to the Ridge regression that adds a penalty equivalent to the absolute value of the magnitude of the coefficients:

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j\, x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\left|\beta_j\right|. \tag{4}$$

As with ridge regression, the lasso regression shrinks the coefficient estimates towards zero.

However, the penalty of the lasso regression has the effect of making some of the coefficient estimates to be equal to zero when the tunning parameter is sufficiently large. As a result, models generated by the lasso regression are much easier to interpret compared to those developed by the ridge regression (James et al., 2013). The model alpha simply represents a constant that is multiplied by the penalty.

3.1.2. SUPPORT VECTOR MACHINES

The SVM algorithm (Cortes & Vapnik, 1995) is a strong machine learning algorithm model used in both classification and regression problems and is popular due to its good generalization performance and computational efficiency. To understand how SVM works, it is important to be familiar with the concept of margin. The margin represents the distance between the decision line of the classifier and the closest observation (Kirchner & Signorino, 2018).

Given the labeled training set, the SVM constructs and uses a discriminant hyperplane or a set of hyperplanes to identify classes. Intuitively, a good separation is achieved by the hyperplane with

maximum distance to the nearest training data samples (i.e., maximum margin) therefore an optimal hyperplane maximizes the margin. This optimal hyperplane is determined by the observations that fall close within the margin, the support vectors. The mapping of the original data into this new space is conducted with the help of kernel functions. Larger margin sizes allow for better and more accurate classifications (Pradhan, 2012).

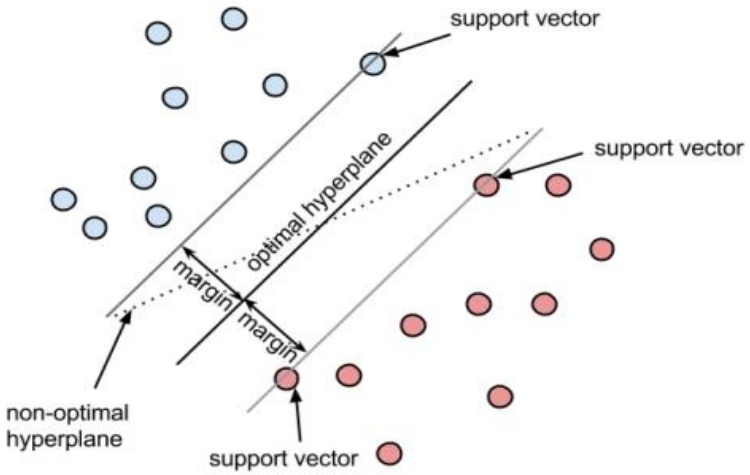

Figure 1: An example of a SVM to determine the optimal hyperplane
Source: Eye Tracking with EEG lifestyle (Haji & Mohammad, 2015)

The SVM algorithm has been successfully applied to a wide range of problems including pattern recognition (Byun & Lee, 2002), bankruptcy prediction (Olson, Delen & Meng, 2012), and record linkage purposes (Christen, 2008), among many applications. The SVM algorithm, like the LR, was trained using SGD due to its computational competence, and the same hyperparameter tunning was applied.

3.1.3. K-NEAREST NEIGHBORS

The KNN algorithm is a type of supervised machine learning algorithm that can be used for classification and regression problems. It is known as a "lazy" learning algorithm since it does not have a specialized training phase. It is also a non-parametric technique, implying it does not make any assumptions about the underlying data distribution (Mukid et al., 2018).

KNN is a distance-based algorithm, taking a majority vote between the "K" closest observations. From a mathematical point of view, distance is defined as a quantitative measurement of how spaced out two data points are. The main idea of the KNN algorithm is that whenever there is a new point to predict, its nearest neighbors are chosen from the training data (Zhang & Jianxue, 2016). For $K > 1$, usually, a majority voting rule is applied. In majority voting, each class takes a vote for each point in the "K" neighbors, who are labeled as that class. Then the unseen data point is classified as the class with the highest number of votes (Zhang et al., 2017).

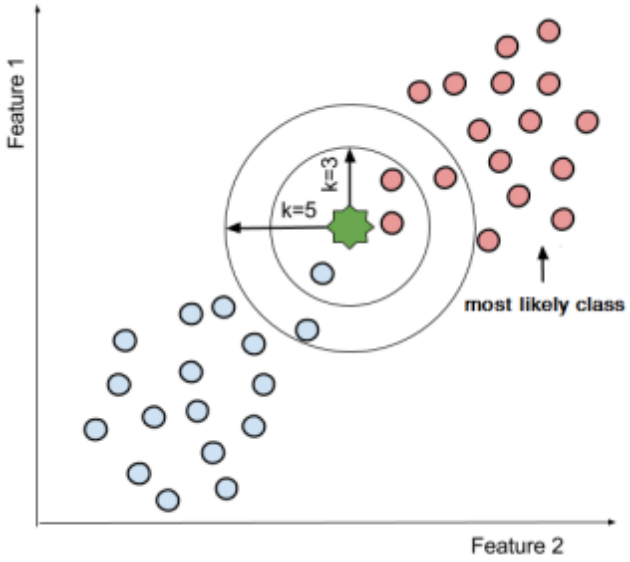

Figure 2: An example of a KNN classification for a two-class problem
Source: Eye Tracking with EEG lifestyle (Haji & Mohammad, 2015)

The output of the KNN algorithm depends on the problem. For classification problems, majority voting is applied among the "K" instances, whereas for regression, an average value of the "K" instances is considered.

Some noteworthy distance metrics include Euclidean, Manhattan, and Minkowski distances. The Euclidean distance simply refers to the distance between two points using the Pythagorean theorem for calculations. It is calculated as the square root of the sum of the squared differences concerning a new point and an existing point across all input variables:

$$D(X,Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{5}$$

The Manhattan distance represents the sum of the distances from all the attributes using the absolute distance:

$$D(X,Y) = \sum_{i=1}^{n}|x_i - y_i| \tag{6}$$

The Minkowski distance is considered a generalization of both the Euclidean and the Manhattan distance. The formula is presented in equation (7):

$$D(X,Y) = \sqrt[p]{\sum_{i=1}^{n}|x_i - y_i|^p} \tag{7}$$

It includes an exponent variable p in its formulation. The Minkowski distance is typically used with p being 1 or 2 which correspond to the Manhattan distance and the Euclidean distance, respectively. For modeling purposes, the Minkowski distance was applied. When performing hyperparameter tunning, a parameter grid was created assigning different values for "*K*," the neighborhood size.

3.1.4. DECISION TREES

A DT is a graphical representation with the primary goal of creating a model that predicts the value of a target variable based on several input variables. In these tree structures, the leaf nodes represent classifications, the inner nodes represent the current predictive attributes, and the branches represent conjunctions of attributions that lead to the final classifications (Zhang et al., 2010).

The algorithm is based on a measure of data impurity that determines the split of each node, using different metrics such as Gini impurity, and entropy, to calculate information gain. These metrics are applied to each node, and the resulting values are combined (i.e., averaged) to provide a measure of the quality of the split (Jijo & Abdulazeez, 2021). Then, the impurity of each test is compared, and the split with the lowest impurity is chosen. This process is then continued for each node of the tree to find the "purest" node in terms of the target variable.

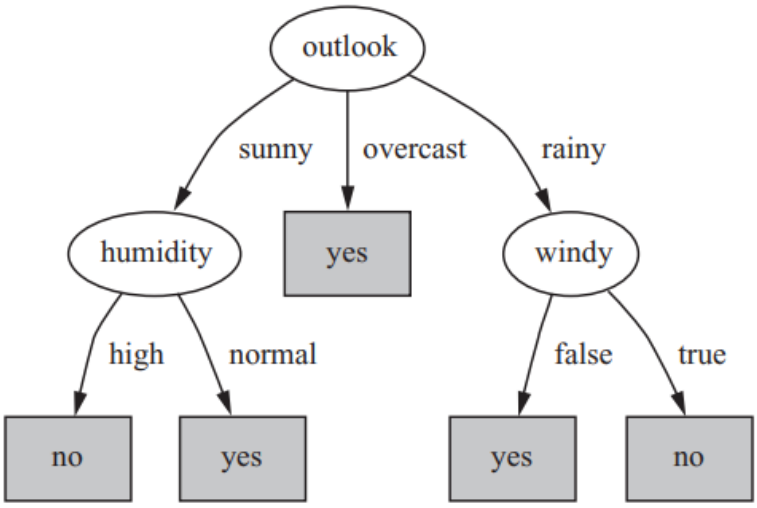

Figure 3: An example of a decision tree based on weather data
Source: Data Mining: Practical Machine Learning Tools and Techniques (Witten & Frank, 2002)

DT algorithms can be applied to both regression and classification problems. Classification trees are designed for dependent variables that take a finite number of unordered values, with prediction error measured in terms of misclassification cost. Regression trees are for dependent variables that take continuous or ordered discrete values, with prediction error typically measured by the squared difference between the observed and predicted values (Loh, 2011). Some notable DT algorithms include ID3 (Quinlan, 1986) and C4.5 (Quinlan, 1993).

As for hyperparameter tunning, both the Gini index and entropy were considered as measures of the impurity of a node. Both formulas are presented in equations (8) and (9), respectively:

$$Gini = 1 - \sum_{i=1}^{c} (p_i)^2 \quad (8)$$

$$Entropy = \sum_{i=1}^{c} -p_i \log_2 p_i \quad (9)$$

Additionally, parameter grids were computed for the minimum number of splits required to split an internal node and for the depth of the DT algorithm.

3.1.5. ENSEMBLE LEARNING METHODS

Ensembles are sets of learning machines that combine in some way data from different modeling approaches to obtain more dependable and more accurate predictions in supervised and unsupervised learning problems (Re & Valentini, 2012).

Dietterich (2000) provides three fundamental reasons why an ensemble may work better than a single classifier. The first reason is statistical, where the author starts by considering situations where the amount of training data is too small causing a single learning algorithm to give the same accurate results on the training data although with different hypotheses. By using an ensemble based approach the author concludes that the algorithm is able to reduce the risk of choosing the wrong classifier and find a good approximation to the true hypothesis. The second reason is computational, where he highlights the problems that arise when the training data is enough, but the learning algorithm is still jammed in local optima making it difficult to find the best hypothesis. An ensemble allows for multiple starting points providing a better approximation to the true hypothesis. The final reason is representational, where in most applications of machine learning the true hypothesis does not belong in the defined space of hypothesis. An ensemble allows for the expansion of the space of representable functions.

An ensemble contains several learners, called based learners, generated by a base learning algorithm. Most ensemble methods produce homogeneous base learners, i.e., learners with the same characteristics, leading to homogeneous ensembles, but some methods utilize multiple learning algorithms to produce heterogeneous learners, i.e., learners with distinctive characteristics leading to heterogeneous ensembles (Zhou, 2012).

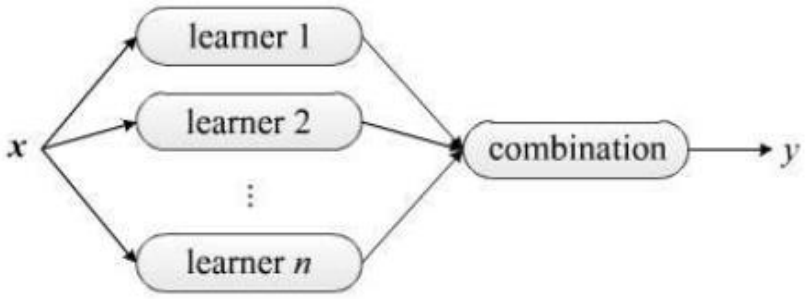

Figure 4: An example of a common ensemble architecture
Source: Ensemble Methods: Foundations and Algorithms (Zhou, 2012)

The simplest approach to ensemble learning is model averaging, where multiple models are trained on the same dataset, and the average of these predictions is taken to make one final prediction. This can be easily computed as shown in equation (10):

$$F_t = \frac{\sum_{i=t-k}^{t-1} Y_i}{k} \tag{10}$$

Model 01	Model 02	Model 03	Model 04	Final Result
70	50	80	40	60

Table 2: An example of model averaging
Source: Author preparation

Due to its simplicity, as models contribute to the final prediction in an equal manner, an extension of model averaging was developed known as weighted model averaging. In this instance, all models are assigned different weights that define the importance of each model for predictions, Equation (11) computes this scenario:

$$F_t = \frac{\sum_{i=t-k}^{t-1} w_i Y_i}{k} \tag{11}$$

Other simple ensemble approaches include majority voting, where multiple models are used to make predictions that are considered a "vote." The final output class label is the one that has more than half of the votes. If none of the class labels receive more than half of the votes, a rejection option is applied, and the combined classifier makes no prediction.

Model 01	Model 02	Model 03	Model 04	Final Result
0	1	1	1	1

Table 3: An example of model averaging by majority voting
Source: Author preparation

Other concepts of voting such as plurality voting and the borda count voting method, are further investigated by (Leung & Parker, 2003).

These simple ensemble techniques were the basis of more advanced techniques. When considering these advanced techniques two paradigms of ensemble learning arise: (i) Parallel ensemble methods, represented by the bagging algorithm, and (ii) Sequential ensemble methods, represented by the boosting algorithm.

Bagging is an ensemble method for improving unstable estimation or classification schemes. Breiman, (1996) motivated bagging as a variance reduction technique for a given base procedure, such as DT or methods that do variable selection and fitting in a linear model. The author also provides three basic steps to implement the bagging algorithm: (i) Bootstrapping, bagging leverages a bootstrapping technique to create diverse samples allowing for the creation of different subsets of the training

dataset; (ii) Parallel Training, these bootstrap samples are then trained independently and in parallel with each other using homogeneous weak learners; (iii) Aggregation, depending on the task an average or a majority vote of the predictions are taken to compute a more accurate estimate.

Boosting aims at improving predictive performance by training homogeneous base learners on a sequence of reweighting datasets. To implement a boosting algorithm: (i) Initial training, train a base learner assigning equal weights to each observation; (ii) Sequential training, in the following learning cycles, the weights are updated according to the performance of the previous learner where the weight of instances with high learning error will increase, while the correctly classified instances will gain lower weights. Thus, the misclassified observations will receive more attention from the following learners. Base learning algorithms are repeatedly trained based on the sampled training set with adjusted weights until the iteration reaches the pre-specified number. Each established base-learner is specialized to the sampled training set in the corresponding learning round (Hu et al., 2021). The final strong learner is then produced.

In contrast to bagging and boosting, stacking implements an ensemble based on heterogeneous base learning algorithms. The typical stacking framework is comprised of two modules, base-learners (level-0) and a meta-model (level-1). The low-level output is taken as the input of the high level for relearning thus, meta-learners generalize the predictions of multiple base-learners (Hu et al., 2021). To conduct stacking: (i) Base Learning, the base level classifiers are trained with the training set and generate their predictions. After training, the base learners create a new dataset for the meta classifier; (ii) Meta Learning, the meta classifier is trained with the new meta dataset. The trained meta classifier is then used to make an overall final prediction by considering each individual prediction made by the base level classifiers (Rokach, 2010).

Bagging:

Input:

Data set $D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_M, y_M)\}$;

Base learning algorithm L;

Number of learning round T.

Process:

For $t = 1, 2, \cdots, T$:

$D_t = bootstrap\ samlple(D)$; #Generate a bootstrap sample from D

$h_t = L(D_t)$; #Train a base learner h_t from the a bootstrap sample

end.

Output:

$H(x) = \arg\max_y \sum_{t=1}^{T} 1\left(y = h_t(x)\right)$

Boosting:

Input:

Data set $D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_M, y_M)\}$;

Base learning algorithm L;

Number of learning round T.

Process:

$D_1(i) = \frac{1}{M}$; #Initialize the weight distribution

For $t = 1, 2, \cdots, T$:

$h_t = L(D, D_t)$; # Train a base learner h_t from D using D_t

$\varepsilon_t = \sum_{i=1}^{M} D_t(i)\left[h_t(x_i) \neq y_i\right]$; # Measure the error of h_t

$\alpha_t = \frac{1}{2} \ln \frac{1-\varepsilon_t}{\varepsilon_t}$; # Determine the weight of h_t

$$Z_t = \sum_{i=1}^{m} D_t(i) \times \begin{cases} e^{-\alpha_t} & if\ h_t(x_i) = y_i \\ e^{\alpha_t} & if\ h_t(x_i) \neq y_i \end{cases};$$

#Z_t is a normalization factor that enables D_{t+1} to be a distribution

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & if\ h_t(x_i) = y_i \\ e^{\alpha_t} & if\ h_t(x_i) \neq y_i \end{cases}; \#$$

Update the distribution

end.

Output:

$H(x) = sign \sum_{t=1}^{T} \alpha_t h_t(x)$

Stacking:

Input:

Data set $D = \{(x_1, y_1), (x_2, y_2), \cdots, (x_M, y_M)\}$;

Base learning algorithm L_t (t = 1,2,⋯, T);

Meta learning algorithm L.

Process:

For $t = 1, 2, \cdots, T$:

$h_t = L(D_t)$; # Train base learners h_t by applying the level-0

end.

$D' = \emptyset$; # Great a new data set

For $m = 1, 2, \cdots, M$:

For $t = 1, 2, \cdots, T$:

$z_{it} = h_t(x_i)$; # Use h_t to classify the training example x_i

end;

$D' = D' \cup \left\{\left((z_{it}, z_{it}, \cdots, z_{it}), y_i\right)\right\}$; # A new data set is finished

end;

$h' = L(D')$; # Train meta-learner h' by applying the level-1

Output:

$H(x) = h'\left(h_1(x), h_2(x), \cdots, h_T(x)\right)$

Figure 5: Pseudo code for bagging, boosting, and stacking approaches
Source: Performance evaluation of ensemble learning techniques for landslide susceptibility mapping at the Jinping county, Southwest China. Natural Hazards (Hu et al., 2021)

3.2. EVALUATION AND PERFORMANCE METRICS

Performance metrics are a part of every machine learning problem. Whether faced with a classification or regression problem, performance metrics are used to monitor and measure the performance of a model during training and testing.

Various performance metrics are used when considering credit scoring applications. Abdou and Pointon (2011) provide an overview of the most frequent performance metrics used in credit scoring. To evaluate the performance of the individual classifiers and compare it to the stacking ensemble approach, the performance metrics considered in this study include the confusion matrix, the ROC curve, and the AUC score.

A confusion matrix is a table layout that allows the visualization of the performance of an algorithm (Powers, 2008). A confusion matrix presents the combinations of the number of predicted and actual observations in a set of data. Table 4 highlights the general form of a traditional confusion matrix.

		Predicted Values	
		Negative = 0	Positive = 1
Actual Values	Negative = 0	True Negative (TN)	False Positive (FP)
	Positive = 1	False Negative (FN)	True Positive (TP)

Table 4: An example of a confusion matrix

Source: Author preparation

Each row of a confusion matrix is an observation in an actual class while the columns are instances in a predicted class. It consists of four basic characteristics that are used as measurement metrics of a classifier. These four characteristics are: (i) True negatives, which represent non-default situations that are correctly classified as non-default; (ii) True positives, which represent default situations that are correctly classified as default; (iii) False positives, which represent applicants that do not default, but the model classified these applicants as default. These are known as type I errors; (iv) False negatives, which represent applicants that default, but the model classified these applicants as non-default. These are known as type II errors.

These characteristics are then used to compute four different metrics:

Accuracy Score: Accuracy simply refers to the number of correct predictions divided by the total number of predictions made by the classifier.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

Precision Score: Precision or Confidence measures how accurately the model can capture default, i.e., out of the total predicted default cases, how many turned out to be default.

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

Recall Score: Recall or Sensitivity measures out of all the actual default cases; how many the model could predict correctly as default.

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

F1-Score: this is a balance between precision and recall. Since the dataset is imbalanced, the F1 score becomes more beneficial since it combines both precision and recall into a single metric.

$$\mathrm{F1 - Score} = 2 * \frac{(Precision * Recall)}{Precision + Recall} \tag{15}$$

To complement these metrics the ROC curve and AUC score will also be considered as additional metrics of performance.

A ROC curve is a graphical plot for visualizing, organizing, and selecting classifiers based on their performance. It shows the relationship between sensitivity and false positive rate (also known as the probability of false alarm) at different threshold points. The closer the ROC curve approaches the top left corner of the graph, the better the quality of the test in terms of its capacity to discriminate between groups. Taking Figure 6 as an example, the lower left coordinate (0, 0) represents the strategy of never issuing a positive classification; such a classifier commits no false positive errors but also gains no true positives. The opposite strategy is represented by the upper right point (1, 1). The point (0, 1) represents a perfect classification.

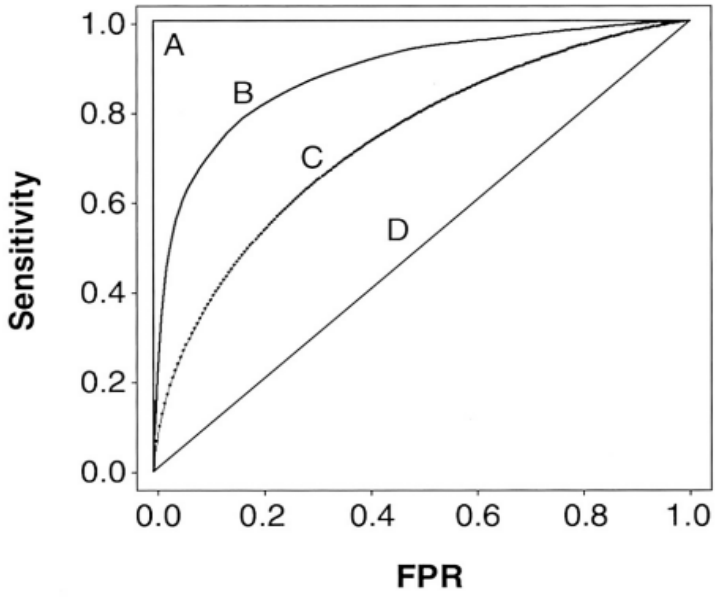

Figure 6: An example of multiple ROC curves
Source: Receiver Operating Characteristic (ROC) Curve: Practical Review for Radiologists (Park, Go & Jo, 2004)

The AUC score represents the area enclosed by the ROC curve. It is a measure of the ability of a classifier to distinguish between both classes with a value range between 0 and 1 where the diagonal line indicates an AUC score of 0.5, equivalent to random guessing.

3.3. SOFTWARE PACKAGES AND VERSIONS

The programming language used in this study is python, version 3.8.10. An overview of all the ground packages, their versions, and functions are displayed in table 5.

Package	Version	Functions
Pandas	1.3.4	Multiple basic functions
NumPy	1.20.1	Multiple basic functions
Matplotlib	3.4.3	Multiple basic functions and graphical representations

Seaborn	0.11.2	Multiple basic functions and graphical representations
Scikit Learn	1.0.2	Train_test-split, StandardScaler, StratifiedKFold, SGDClassifier, KNN, DecisionTreeClassifier, GridSearchCV, Pipeline, Column Transformer, Learning Curve, PCA, OrdinalEncoder, OneHotEncoder
Imblearn	0.9.0	RandomUnderSampler, RandomOverSampler, Pipeline

Table 5: Packages and functions used
Source: Author preparation

3.4. DATASET PROPERTIES

The dataset used in this dissertation belongs to Lending Club, a peer-to-peer lending company that matches lenders and borrowers through an online platform without the need for any financial intermediation. It contains data for all loans issued between 2007 and 2018 divided into two sets of files: accepted loans and reject loans. The dataset focuses on three different aspects: (i) Personal details (e.g., address, employment status, homeownership); (ii) Credit history (e.g., the balance of accounts, revolving and current past due accounts); and (iii) Loan characteristics (e.g., application type, purpose, grade, term). **Annex I** provides an overview of these variables and their description.

It is available at Kaggle[1], an online community dedicated to machine learning practitioners that provides public data sets.

[1] https://www.kaggle.com/datasets/wordsforthewise/lending-club

4. MACHINE LEARNING APPROACH TO CREDIT SCORING

This part of the dissertation seeks to be a walkthrough of the machine learning approach. It will focus on two different steps: (i) Exploratory data analysis and (ii) Data preprocessing. It will also introduce the concept of cross-validation and the sampling techniques considered.

4.1. INITIAL PREPROCESSING AND EXPLORATORY DATA ANALYSIS

The dataset was downloaded from Kaggle, containing 151 unique features and 2.26 million rows. Given that the focus of this dissertation is to predict default only loans with a known final status were contemplated, eliminating any records that do not meet this criterion leaving a total of 1.34 million rows of fully paid and charged off loans. There were 58 columns with more than 25% of missing data, which were considered unusable and consequently dropped. Along with these variables, only features that were available at the time of application were considered since the main purpose is to predict the possibility of an applicant entering a default state before the loan is granted. Of the remaining loans, 1,076,751 correspond to fully paid loans and 268,559 to charged-off loans, totalizing 1,345,310 loans.

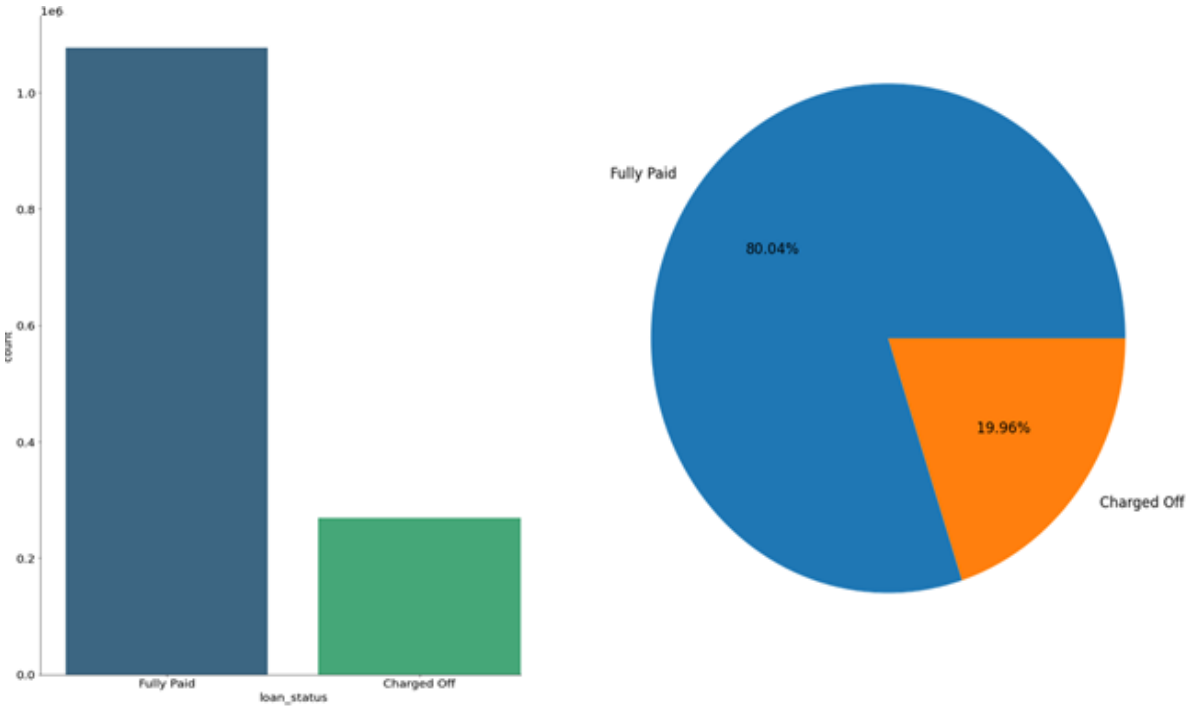

Figure 7: Frequency and percentage of loan status
Source: Author preparation

It is evident by both graphs that class imbalance exists in this dataset, although not extreme. To deal with this issue, different sampling techniques will be presented in section 4.4.

From the selected features, variables such as the id of the loan and zip code were immediately dropped since these do not have any impact on the likelihood of an applicant defaulting.

Borrowers that apply for a loan provide personal information such as the purpose of the requested loan, address state, annual income, and loan amount and the platform uses this information to assess

the applicant's creditworthiness. Figure 8 presents the applications of these loans where most loans fall in debt consolidation, credit card, and home improvement categories.

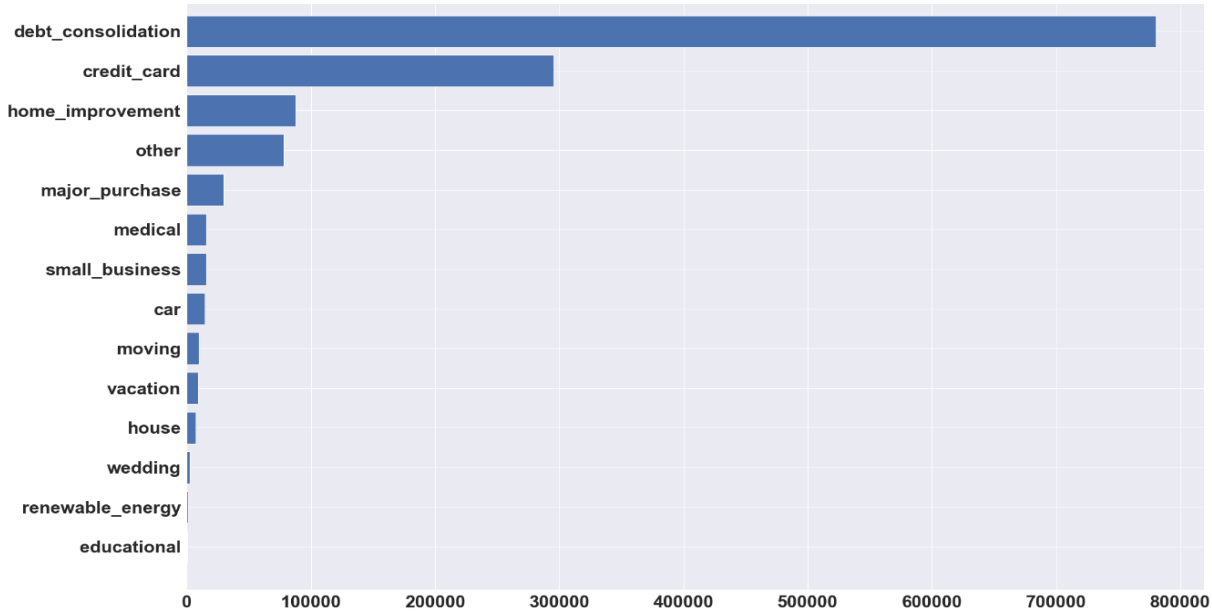

Figure 8: Purpose occcurrence
Source: Author preparation

Despite the higher demand for these categories, they do not represent the highest probability of default. These fall in small business applications with a 30% probability of no repayment. Additionally, the educational category does not present high loan amounts compared to the other categories. This could be associated with the fact that the borrowers applying with education in mind are students who want to pursue a higher level of education.

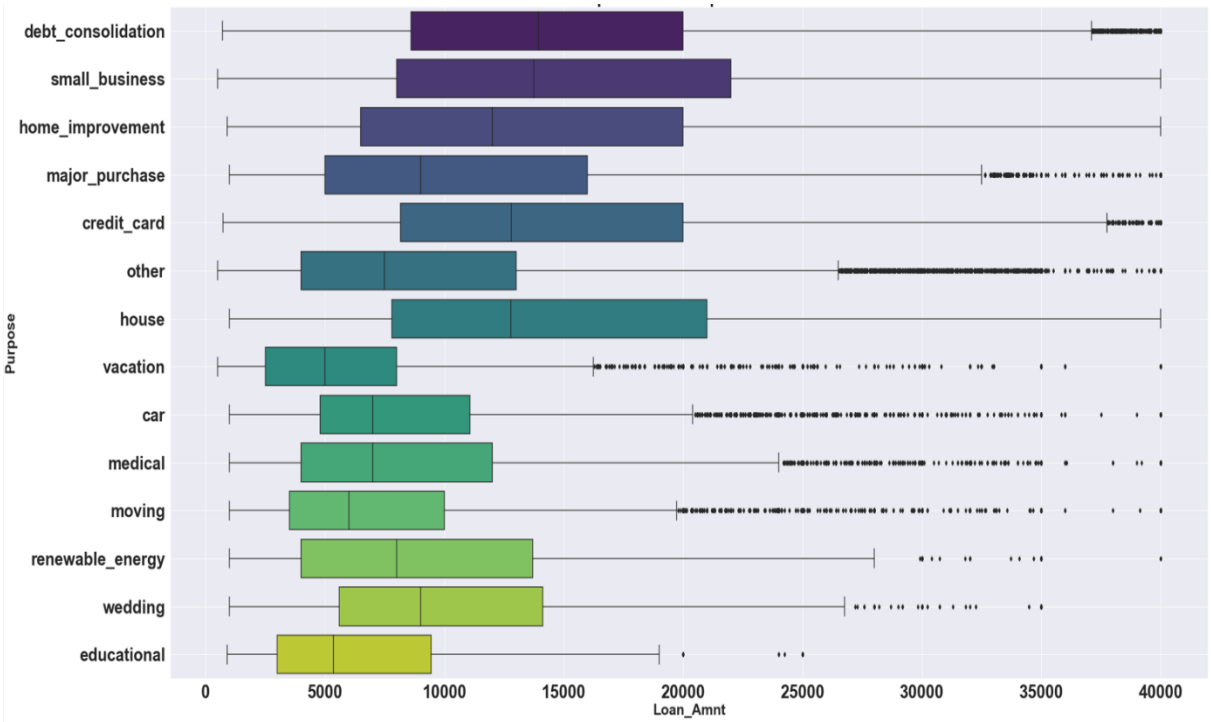

Figure 9: Relationship between Purpose and Loan Amount
Source: Author preparation

The debt-to-income ratio (DTI) has an average of 18,28% meaning a borrower spends 18,28% of his or her monthly income on debt. DTI is a measure of comparing the net income of borrowers and serves as an extra indicator of the potential wealth of a borrower. Furthermore, DTI is on average higher for charged-off loans (20,17%) compared to fully paid loans (17,81%).

Extra credit worthiness metrics provided include the FICO score, a method to quantify and evaluate an individual credit performance that considers payment history, the current level of indebtedness, types of credit used, length of credit history, and new credit account for calculations. The dataset contains both the upper and lower bound of the FICO score, the average of these bounds was calculated for modeling and a new variable "fico_score" was created.

Regarding the duration of the loan, the platform provides loans with a duration of 36 and 60 months where the latter has a higher probability of default (32,45% for 60 months loans and 15,99% for loans of 36 months). An explanation for this event could be associated with the fact that a longer maturity increases the possibility of financial instability of borrowers and thus a longer period of default.

Lending Club also uses a rating score with seven possibilities (grade A loans to grade G loans) attributed based on the credit information provided by the borrower. There is a clear upward trend between the grade and the interest rate assigned to a loan. As the grade tends to aggravate, the interest rate is higher for said loans. This correlation is logical, if an applicant is likely to default extra steps should be taken to ensure the proper protection against the risk of non-remittance. Figure 10 demonstrates this relationship:

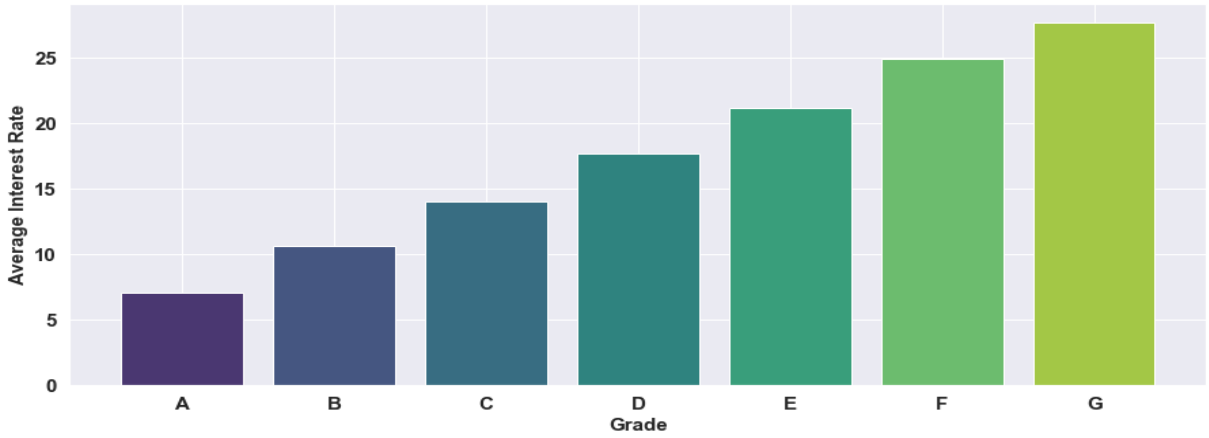

Figure 10: Relationship between grade and interest rate
Source: Author preparation

Likewise, another grade variable, sub-grade, is available that provides the same information as the grade variable but in more detail. Since including both variables would be redundant, the sub-grade variable was dropped, and the grade variable was used for modeling purposes.

The dataset also contains information regarding the type of employment and the length of employment. Both variables were not included in the final model with the following reasoning: (i) The type of employment contains 378,353 unique entries and many of these entries are oddly specific titles. Additionally, many of these titles have duplicate entries that would require proper treatment. When selecting variables for modeling it is important to evaluate not only the potential relationship between the said variable and our target variable but also the quality and transparency of the

information; (ii) The length of employment describes how long a borrower has been employed before asking for a loan. It has 78,511 missing records that could be associated with, for instance, information not provided by the borrower at the time of application or unemployment. Like the type of employment, this variable is also associated with socioeconomic conditions, and by looking at the probability of default it remained almost constant, so it was not considered for the modeling stage.

Home Ownership, the home status provided by the borrower at the time of application, presented six features but for simplicity purposes, features that do not provide a specific application were grouped in a feature denoted as “Other.”

Supplementary to the process of granting a loan is the type of application. Lending Club provides two options: an individual and a joint application. Despite joint applications being lesser compared to individual applications, these have a higher probability of default (24.59% compared to 19,87%, respectively).

There is also information concerning the issue date and the date of the earliest reported credit line. According to the information provided, there was a clear upward trend of issued loans from 2007 until 2015 and it started to decrease in 2018. Both variables were considered redundant and not included in the modeling stage.

Other features provided and included for modeling, are associated with credit characteristics. These include inquiries, revolving balance, mortgage accounts, delinquencies, opened accounts, and public records and serve as insightful information for the lender and affect the probability of default of the borrower.

To conclude this analysis, the correlation between each numeric feature was analyzed. Fig 11 presents these insights:

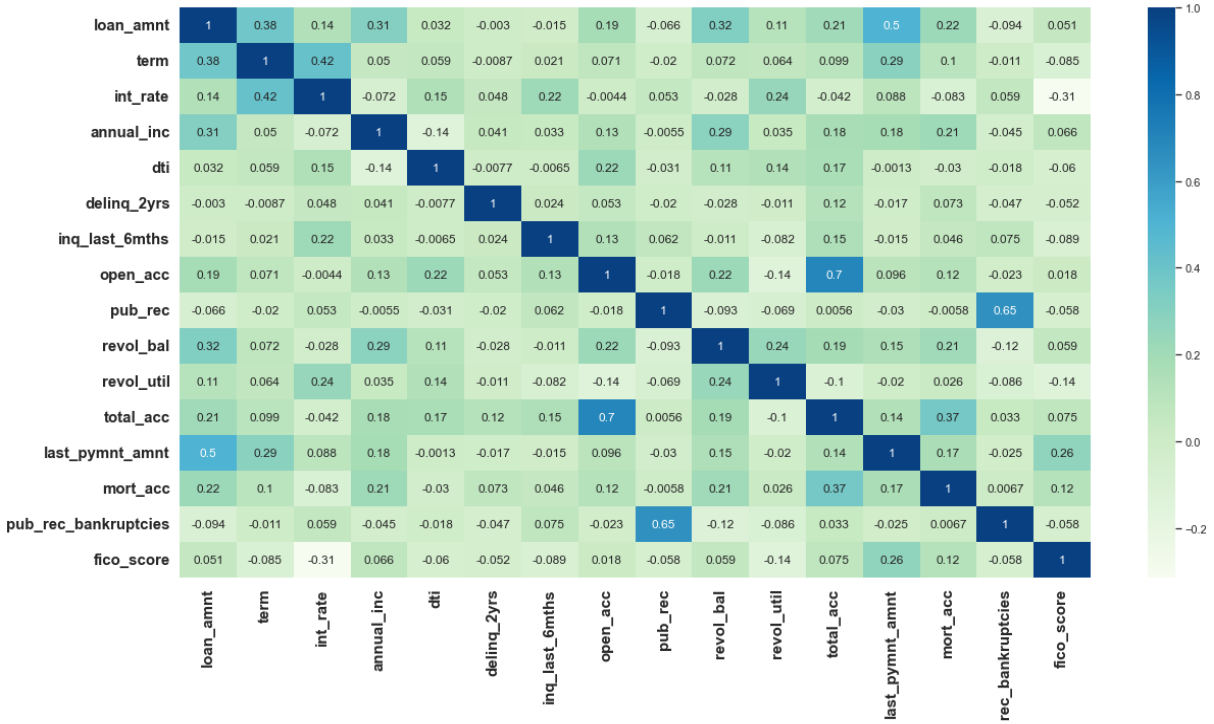

Figure 11: Correlation matrix
Source: Author preparation

From the correlation matrix, both the installment and the loan amount present a correlation of 95%. Including both these variables in the modeling stage would be superfluous, the installment feature was dropped in favor of the loan amount.

Table 6 provides a summary of the twenty-three features that were chosen to be a part of the models, including a description of their type:

Feature	Type	Group
Loan_amnt	Numeric	Loan Characteristics
Term	Numeric	Loan Characteristics
Int_rate	Numeric	Loan Characteristics
Grade	Categorical	Loan Characteristics
Verification_status	Categorical	Borrower Characteristics
Home_ownership	Categorical	Borrower Characteristics
Annual_Inc	Numeric	Borrower Characteristics
Loan_status	Categorical	-
Purpose	Categorical	Loan Characteristics
Addr_state	Categorical	Borrower Characteristics
Dti	Numeric	Credit Characteristics
Delinq_2yrs	Numeric	Credit Characteristics
Inq_last_6mths	Numeric	Credit Characteristics
Open_acc	Numeric	Credit Characteristics
Pub_rec	Numeric	Credit Characteristics
Revol_bal	Numeric	Credit Characteristics
Revol-util	Numeric	Credit Characteristics
Total_acc	Numeric	Credit Characteristics
Last_pymnt_amnt	Numeric	Borrower Characteristics
Application_type	Categorical	Loan Characteristics
Mort_acc	Numeric	Credit Characteristics
Fico_score	Numeric	Credit Characteristics
Pub_rec_bankruptcies	Numeric	Credit Characteristics

Table 6: Features chosen for modeling
Source: Author preparation

4.2. DATA PREPROCESSING

Data preprocessing is the process of transforming data into a comprehensible format. By looking at the descriptive statistics some numeric variables have outliers. Outliers are observations located at an unusual distance from other values in a sample or a population. It is important to assess whether it represents a true value that should be kept or if it results from incorrect or measured data. Table 7 presents the descriptive statistics of all numeric features:

Feature	Mean	St. Dev	Min	Median	Max
Loan_amnt	14,419.97	8717.05	500.00	12,000.00	40,000.00
Term	41.79	10.27	36.00	36.00	60.00
Int_rate	13.24	4.77	5.31	12.74	39.99
Annual_inc	76,247.64	69925.10	0.00	65,000.00	10,999,200.00
Dti	18.28	11.16	-1.00	17.61	999.00
Delinq_2yrs	0.31	0.88	0.00	0.00	39.00
Inq_last_6mths	0.66	0.94	0.00	0.00	8.00
Open_acc	11.59	5.47	0.00	11,00	90.00
Pub_rec	0.21	0.60	0.00	0.00	86.00
Revol_bal	16,248.11	22,328.17	0.00	11,134.00	2,904,836.00
Revol_util	51.81	24.52	0.00	52.20	892.30
Total_acc	24.98	11.00	2.00	23.00	176.00
Last_paymn_amnt	5,423.57	7,117.00	0.00	2042.05	42,192.05
Mort_acc	1.67	2.00	0.00	1.00	51.00
Pub_rec_bankruptcies	0.13	0.38	0.00	0.00	12.00
Fico_score	668.72	102.69	0.00	692.00	847.50

Table 7: Descriptive statistics of numeric variables
Source: Author preparation

Analysing table 7, there are some borrowers with suspiciously high annual incomes. For instance, there was a borrower who reported an annual income of 1,400,000.00 and applied for a loan of 16,000.00. This record was considered erroneous. Furthermore, there were a total of 290 records that reported an annual income higher than 1,000,000.00. These records were dropped in the final dataset to avoid skewness in the final results. Other interesting patterns found were: (i) The max value of DTI is 999.00, it is evident that this value is abnormally large considering the median of this variable. Further analysis denotes that the majority of the loans are concentrated between a DTI of 12% and 24%. Additionally, it presents two records with the value -1. The maximum value was set to 60 and the minimum set to 0; (ii) The revolving balance feature also presents an unusual max value of 2,904,836.00. The same treatment was performed as annual income, fifteen records present a revolving balance higher than 1,000,000.00, and these were immediately dropped; (iii) The fico score presents 209 records with the value 0 which does not make any sense, so these were also dropped.

After dealing with these outliers the data was split into two sets: a train set, the initial data used to evaluate the machine learning algorithms, and a test set, the remainder data used to get an unbiased evaluation of the machine learning models that were fit in the train set. The split ratio is 80% for the train set and the remaining 20% for the test set.

Following the split, additional preprocessing steps are required such as treating missing values, encoding categorical features, and standardizing the data for modeling. To achieve this, a pipeline was built using the libraries provided by `sklearn`. A machine learning pipeline is a procedure to automate a workflow enabling data to be transformed into a model that can then be analyzed to achieve

solutions. A pipeline makes the process of inputting data into the machine learning algorithms fully automated avoiding data leakage, i.e., when information outside the training dataset is used to create the model resulting in biased estimations. Fig 12 introduces the machine learning pipeline used in this dissertation:

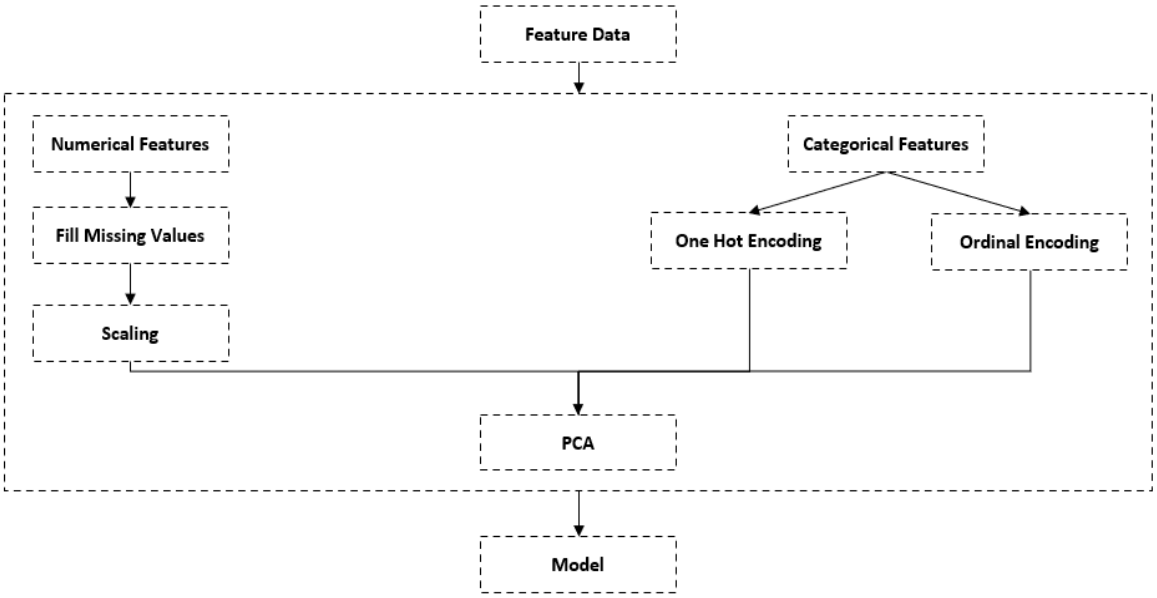

Figure 12: Machine learning pipeline
Source: Author preparation

From the selected features considered for modeling, transformations were performed to both numerical and categorical features: (i) For numerical features, five have missing values but these represent less than 10% of the total data. These features were median imputed. Additionally, some models require numerical features to be appropriately scaled. To achieve this, numerical features were standardized by removing the mean and scaled to unit variance; (ii) For categorical features, three types of encoding were performed. The grade variable received a different treatment since its levels follow a specific ordering. Proper mapping was performed respecting said order. The target variable was properly encoded to 0 (represents fully paid loans) and 1 (represents charged-off loans). The remaining categorical features do not follow a specific order so dummy variables were created using the one hot encoding library for integer representation.

Finally, principal component analysis (PCA) was performed. The main idea of PCA is to reduce dimensionality of a large dataset, by transforming a large set of variables into smaller ones while retaining most of the information from the large dataset (Mishra et al., 2017). The main reason behind the use of PCA is associated with the improvement of algorithms performance, the reduction of overfitting of the models, and reducing the number of dimensions by discarding irrelevant information.

4.3. K FOLD CROSS-VALIDATION AND HYPERPARAMETER TUNNING

Cross-validation is a resampling method to assess the generalization ability of predictive models. The general procedure is to partition the data into K subsets where each subset is used as testing data in one of the K iterations and the remaining K-1 subsets are used for model training.

Yoonsuh (2017) states that a typical choice for K is between five and ten. The author also highlights the possibility of having multiple predictions which are then averaged to produce a final prediction that may be more accurate compared to a single prediction. For this research, given the imbalance nature of the dataset, stratified K fold cross validation was performed to ensure that each partition contains the same proportions of both class labels and K was set to five. Fig 13 presents a representation of the five-fold cross-validation. The dataset was split into five parts where four represent the training data and one the test data. Using the predictions of each fold, the average $Error$ is calculated and used to evaluate the models.

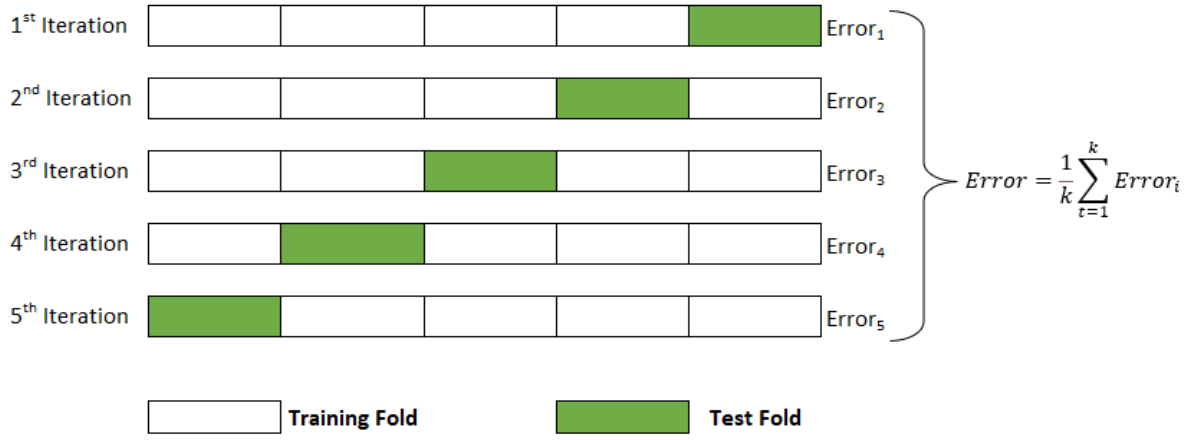

Figure 13: K fold cross validation
Source: Author preparation

In addition to cross-validation, the tunning of hyperparameters was performed using a parameter grid to maximize the underlying score of the estimator. The performance obtained in each combination of hyperparameters was measured using the f1 score. The choices for the hyperparameters for each model were presented in section 3.1. Table 9 provides a summary of the hyperparameters chosen for model optimization and the tuned parameters used for predictions.

Models	Hyperparameters	Values	Chosen Values
Logistic Regression	Penalty	L1, L2	L2
	Model Alpha	10**-3, 10**-2 , 10**-1	10**-2
Support Vector Machine	Penalty	L1, L2	L1
	Model Alpha	10**-3, 10**-2 , 10**-1	10**-3
K Nearest Neighbors	Neighborhood Size	3, 5, 7, 9, 11	9
	Metric	Minkowski	Minkowski
Decision Tree	Criterion	Gini, Entropy	Entropy
	Min_Sample_Splits	20, 30, 50, 100	100
	Max_Depth	3, 5, 10, 15, 20	15

Table 8: Hyperparameter optimization
Source: Author preparation

4.4. RESAMPLING TECHNIQUES

Imbalanced data is a common problem in most classification tasks. It simply refers to a problem where the classes are not represented equally. The main purpose of sampling methods is to create a dataset that has a balanced class distribution so that traditional classifiers can distinguish the minority and majority classes better. For this research, two types of sampling techniques were combined: (i) An undersample method, which takes the majority class to make it reach a size closer to that of the smaller class; and (ii) An oversample method, which takes copies of the minority class to reach a size closer to that of the larger class (Estabrooks & Japkowicz, 2001).

While undersampling and oversampling allow the creation of more balanced class distributions some concerns need to be raised. For instance, in undersampling vast quantities of data are discarded. This can be highly problematic, as the loss of such data can make it harder for the classifier to learn, resulting in a loss in classification performance (Mohammed, Rawashdeh & Abdullah, 2020). Alternatively, in random oversampling, since instances are repeated until balanced data is achieved the possibility of overfitting increases drastically, making the generalization performance of the classifier extremely poor (Chawla et al., 2002). Thus, when using sampling techniques, one needs to determine how much sampling should be applied.

For this reason, this dissertation used a combination of both meaning a modest amount of undersampling is applied to the majority class (a sampling strategy of 50% was applied), and afterwards, the data was oversampled by the minority class, instances were created until the data was balanced. Research suggests that using a combination of both approaches provides a great deal flexibility and versatility (Jiang et al., 2019).

5. RESULTS AND DISCUSSION

The concluding section of this dissertation serves to present the results from the application of the different machine learning experiments and compare them to the stacking approach.

5.1. HYPERPARAMETER OPTIMIZATION AND RESAMPLING RESULTS

As stated in the sections above, the purpose of this dissertation is to optimize the process of evaluating credit risk by improving the efficiency of machine learning models in evaluating the probability of default. Table 9 provides the first experimental results, hyperparameter optimization with no sampling techniques applied:

	Accuracy		Recall		Precision		F1 Score		Auc Score	
	Before Tunning	After Tunning	Before Tunning	After Tunning	Before Tunning	After Tunning	Before Tunning	After Tunning	Before Tunning	After Tunning
Logistic Regression	0.7438	0.7489	0.7676	0.7607	0.4221	0.4276	0.5446	0.5474	0.8325	0.8329
Support Vector Machine	0.7374	0.7454	0.7766	0.7667	0.4156	0.4238	0.5415	0.5459	0.7528	0.7564
K Nearest Neighbors	0.8315	0.8396	0.4639	0.4566	0.6012	0.6372	0.5237	0.5320	0.7845	0.8453
Decision Tree	0.9135	0.9258	0.7757	0.9433	0.7876	0.7492	0.7816	0.8351	0.5768	0.9764
Average	0.8066	0.8149	0.6960	0.7318	0.5566	0.5595	0.5979	0.6151	0.7365	0.8528

Table 9: Hyperparameters optimization results
Source: Author preparation

The results obtained from the first experiment demonstrate how the usage of hyperparameter tuning can increase the overall performance of a classifier. The DT classifier had the best overall performance among the different classifiers. It had the highest f1 score of 0.8351 which is the proportion of charged-off applicants correctly classified and the overall capability of the classifier to predict default. Furthermore, the AUC score of the DT algorithm, which represents the classifier's ability to distinguish classes, outclassed every algorithm which further strengthens its predictive capabilities. The LR and SVM, trained under SGD, present almost equivalent results with the LR coming slightly ahead with a f1 score of 0.5474 compared to 0.5459, respectively.

Regarding the KNN, despite its relatively good performance, during the training phase, it was evident that the algorithm was overfitting. Fig 14 shows the plot of the learning curve that further proves this evidence. Learning cures are widely used diagnostic tool for machine learning problems since it shows the score evolution throughout different subsets of data.

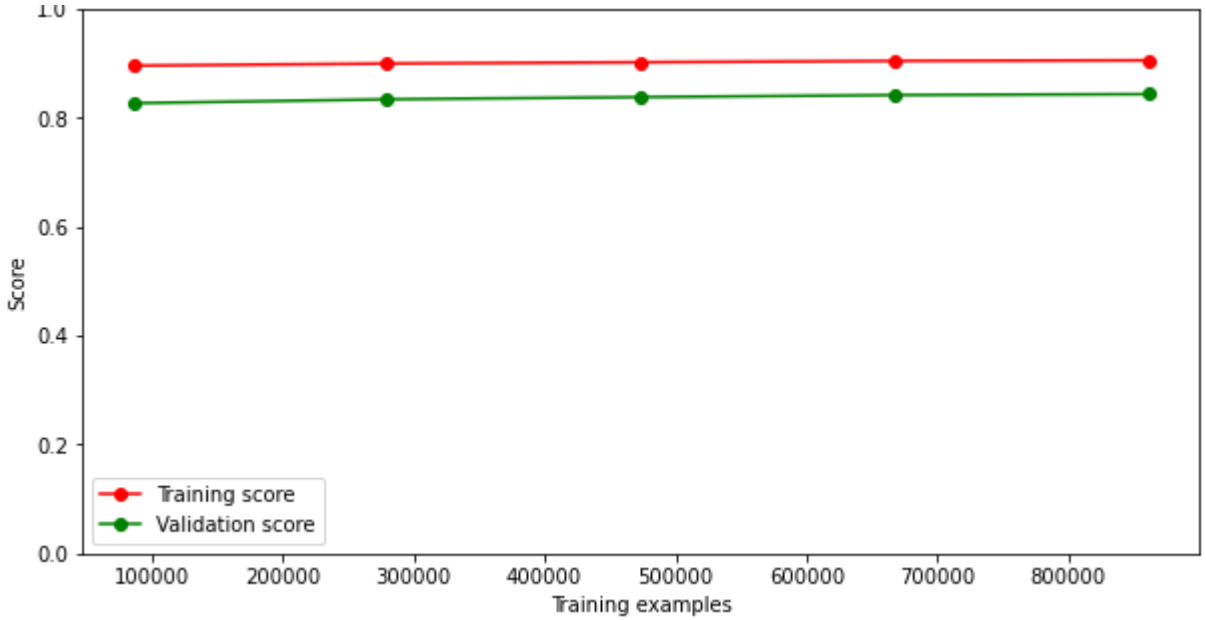

Figure 14: Learning curve of the KNN algorithm
Source: Author preparation

As training examples increase, the model is not able to perform well on unseen data. It is capturing blunders from the data that prevent a good generalization. This is one of the main reasons why resampling techniques will be applied in the next section, as this helps prevent poor generalization in the classifiers.

After this initial analysis, sampling was applied to the dataset. Table 10 presents the results of this second experiment:

Model	Accuracy	Recall	Precision	F1 Score	AUC Score
Logistic Regression	0.8387	0.8514	0.5636	0.6783	0.9185
Support Vector Machine	0.8379	0.8584	0.5618	0.6782	0.8450
K Nearest Neighbors	0.8607	0.8829	0.6003	0.7167	0.9265
Decision Tree	0.9262	0.9422	0.7514	0.8361	0.9732
Average	0.8659	0.8837	0.6193	0.7272	0.9162

Table 10: Results with sampling strategy applied
Source: Author preparation

From the second experiment, it is evident the improvements across all four models. The DT algorithm despite the loss in its recall score still holds up as the best classifier even when applying sampling techniques. It is also interesting to point out that the DT algorithm benefited the least from the sampling approach compared to the remainder classifiers. Fig 15 presents a comparison of the F1 score considering both experiments:

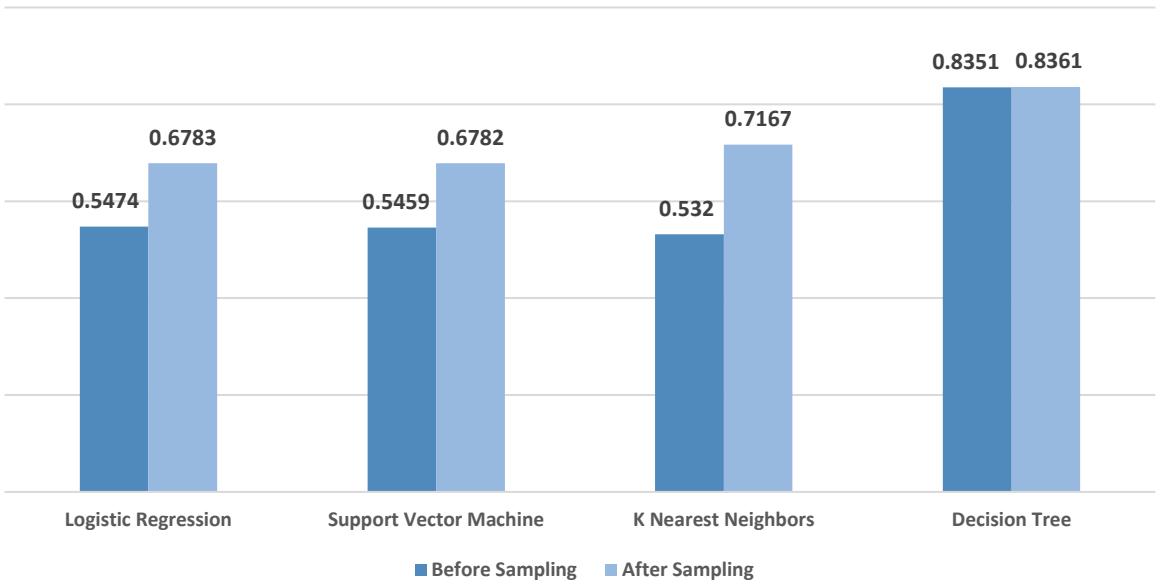

Figure 15: Comparison of the f1 score for both experiments
Source: Author preparation

The results provided by both the SVM and LR algorithms are remarkably close to each other while the KNN algorithm benefited the most from this sampling strategy. During the implementation phase, the problem of overfitting that was present in the KNN algorithm was resolved which demonstrates the strength of applying sampling methods.

5.2. STACKING GENERALIZATION APPROACH

Having a set of diverse base-level learners is essential things for a successful stacking solution. After performing both hyperparameter optimization and sampling, the best base-level learners and meta-learners were taken to perform the ensemble approach experiments.

The setup of the stacking ensemble considering three base learners can be seen in Fig 16. When an input instance is classified, the base learners first make their predictions. These predictions serve as meta-features for the meta learner that makes a final prediction.

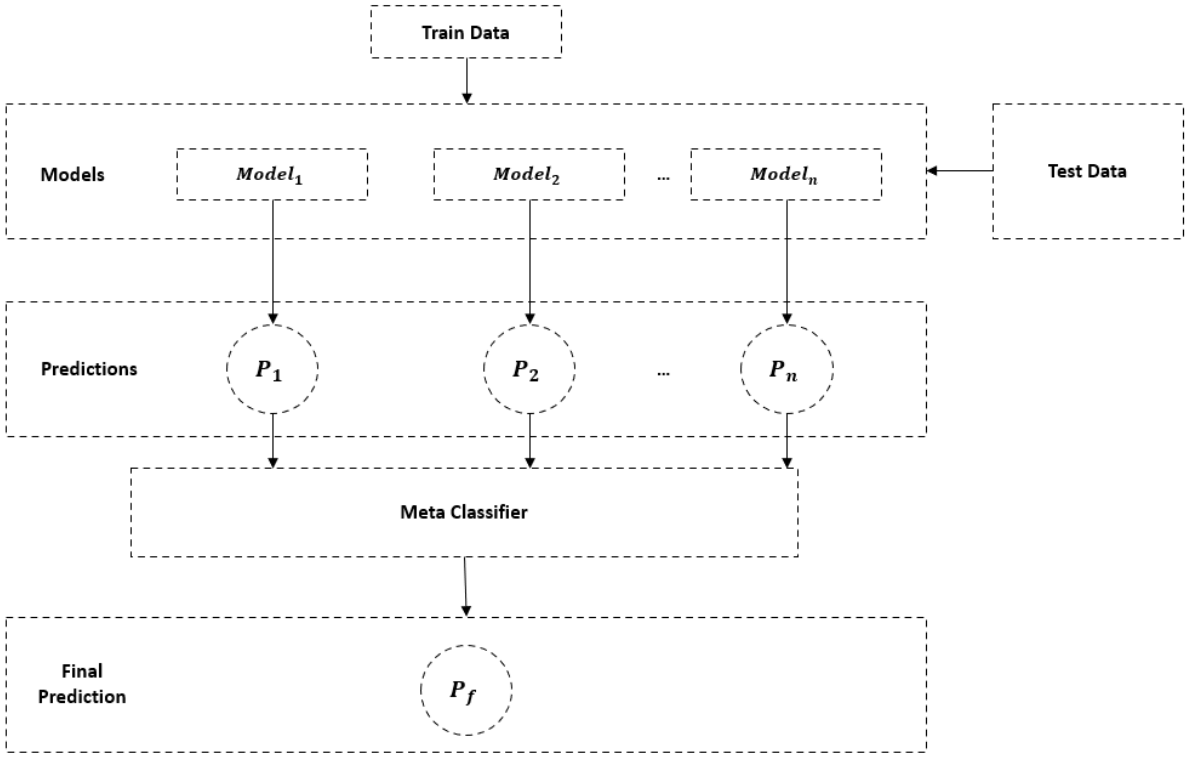

Figure 16: Stacking generalization approach
Source: Author preparation based on (Jiang et al., 2019)

For the stacking approach, four different model combinations have been experimented with. The stacking approaches were built and trained using five-fold cross-validation. As mentioned in section 4.3. the data is split into five different parts, the base learners are trained on the training set and then make their predictions on the test set. The predictions for each iteration are then gathered, along with the corrected labels. These are then used as inputs for the meta classifier which will discover the strengths and weaknesses of each base learner and how to best combine these predictions achieving an overall result. Table 11 provides a synopsis of the ensemble combinations considered and Table 12 the results for each combination:

Combination	Base Learners	Meta-Model
SC1	K Nearest Neighbors and Support Vector Machine	Logistic Regression
SC2	Decision Tree and Support Vector Machine	Logistic Regression
SC3	Decision Tree and K Nearest Neighbors	Logistic Regression
SC4	Decision Tree, K Nearest Neighbors and Support Vector Machine	Logistic Regression

Table 11: Model combinations
Source: Author preparation

Stacking Combination	Accuracy	Recall	Precision	F1 Score	AUC Score
SC1	0.8623	0.8790	0.6072	0.7182	0.9335
SC2	0.9265	0.9415	0.7525	0.8365	0.9726
SC3	0.9284	0.9404	0.7587	0.8399	0.9739
SC4	0.9277	0.9409	0.7564	0.8386	0.9731

Table 12: Results for each combination
Source: Author preparation

From the results of Table 12, the majority of the stacking ensemble approaches performed better than the best base-level learners. The best ensemble is SC3 – stacking with DT and KNN as the base level models and the LR as the meta-model followed by SC4, the combination that includes all models. These results may indicate that the usage of the SVM algorithm as a base learner in the stacking ensemble may be confusing the meta learner, lowering the importance of base learner's predictions, and decreasing the overall performance of the stacking ensemble. Furthermore, SC1 – stacking with SVM and KNN performed much worse than the single best classifier, the DT algorithm further reinforcing its robustness and predictive capabilities. Fig 16 provides a comparison of the f1 score of the different ensemble approaches and individual classifiers:

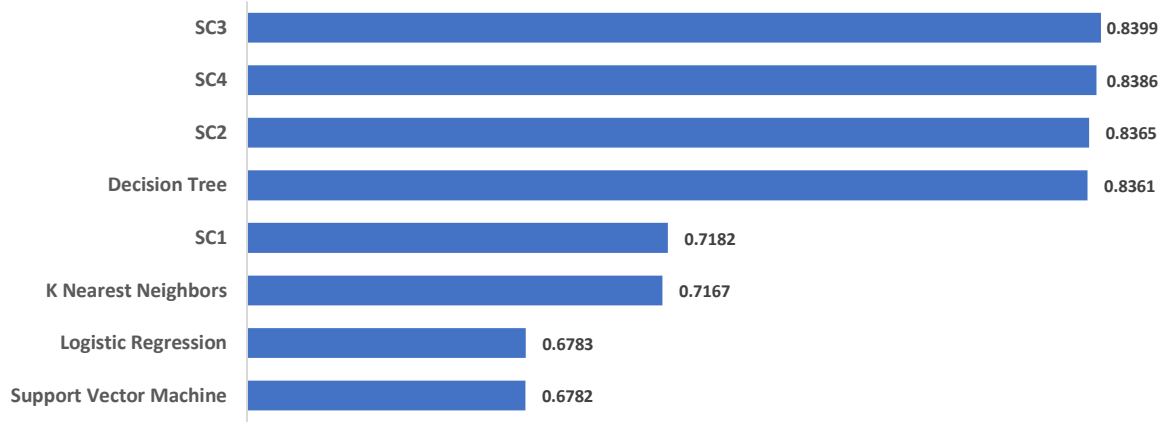

Figure 17: Comparison between stacking ensemble and individual classifiers
Source: Author preparation

It can be concluded that, depending on the classifiers used, the model combination does overall perform better than single classifiers. Although the stacking ensemble approaches improvements over the individual base learners, the increased improvement may not always be worth the increased complexity. Stacking ensemble is a good approach for pushing the performance limits but the improvements of the approach and the costs need to be independently evaluated by the researcher.

6. CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

Credit scoring methods serve as a risk management tool that allows a financial institution to check applicant reliability to pay off the debt in time. With the developments made in machine learning applications, credit scoring replaces the reliance on “gut feeling” with statistical analysis reducing the potential risks associated with personal judgment. Furthermore, the loan application can be processed much faster, which increases customer satisfaction and reduces costs.

The main objective of this dissertation was to minimize credit risk by utilizing different machine learning models to evaluate the probability of default and identify possible loan defaulters. Due to the peculiar nature of the dataset, it was extremely important to understand how credit scoring is conducted, perform rigorous data exploration to gain and extract useful insights about the data and how can preprocessing be applied, taking into consideration each base classifier's characteristics. During this phase, the use of machine learning pipelines facilitated the implementation of different transformations such as dealing with missing values, encoding categorical variables, and proper scaling. In addition to this, cross-validation and hyperparameter tunning were performed to ensure robustness in the classifiers. Four different classification algorithms were employed to build a supervised machine learning model. The results obtained showed that the DT algorithm performed best with the highest f1 score compared to the remainder classifiers in the first experiment. As with most credit datasets, the Lending Club dataset is highly imbalanced, therefore a sampling technique was employed to overcome the imbalance of class distribution in the target variable. The second experiment provided a significant improvement in the LR, SVM, and KNN algorithms while the DT algorithm did not see many improvements but still outperformed the other classifiers.

After model combination through stacking ensemble was performed. The results show that the proposed stacked generalization model is comparatively superior in performance to the single-based model classifiers. It is shown that the combination of the DT and KNN algorithms provided better results illustrating the ensemble method's usefulness in allowing effective decision-making for financial institutions. However, when considering an ensemble-based approach some considerations need to be made: (i) Reduction of model interpretability, when using an ensemble-based approach due to the increase in model complexity it is exceedingly difficult to draw any crucial business insights; (ii) Computation time, an ensemble is a very computationally expensive approach which might not be suitable for real-life applications.

During the realization of this dissertation, several lessons have been learned, which must be mentioned. They are summarized as follows: (i) Dealing with real data is not an easy task, besides the difficulties regarding the nature of the variables analyzed, data must be properly preprocessed; (ii) Depending on the task at hand the simplest solution may be the best one when deploying a machine learning model, it is not only important to gain key insights on the data but also its application in realistic scenarios. Solving complex problems is an arduous task and even if the optimal solution for the task is achieved, it will not be of much use if it lacks interpretability; (iii) There is always room for improvement, new promising methods, and techniques that may improve results achieved by the proposed solutions are constantly arising permitting continuous learning.

In general, machine learning is a powerful toolbox for financial analysts to make predictions and discover patterns in data with precision and rigor. Future research should focus on experimenting with

other state-of-the-art machine learning models combinations and different sampling techniques such as, Synthetic Minority Oversampling Technique (SMOTE) (Chawla et al., 2002). Unlike random oversampling which only duplicates random instances of the minority class, SMOTE creates synthetic data points that are slightly different from the original data points. These new instances are generated based on the distance of each data point and the minority class nearest neighbors. This procedure is executed until balancing is achieved. By using SMOTE the possibility of overfitting is minimized and no information is lost during sampling. However, by introducing these new synthetic points, the possibility of adding noise to the data increases.

Additionally, given the challenges that come with unbalanced data it would be interesting to evaluate a cost sensitive learning approach since it allows for different misclassifications costs for both type I and type II errors. Rather than artificially balancing the data distribution via sampling techniques, cost sensitive learning solves the imbalanced problem by utilizing cost matrices that outline the costs associated with the misclassifications of the various classes (Mienye & Sun, 2021). Research has shown that cost sensitive learning yields enhanced performance in applications where the dataset has a skewed class distribution (Yu et al., 2018).

Setting up a credit scoring model is not a one-time task. It is a sophisticated discipline that requires extensive knowledge and specialized expertise for continuous and precise model improvement. As technological advancements are made in the field of finance, it is crucial to further study and develop techniques that reflect modifications made in consumer behavior and market conditions.

7. REFERENCES

Abdou, H., & Pointon, J. (2011). Credit Scoring, Statistical Techniques and Evaluation Criteria: A Review of the Literature. Int. Syst. in Accounting, Finance and Management. 18. 59-88. https://doi.org/10.1002/isaf.325

Altman, E. (1968). Financial Ratios, Discriminant Analysis, and the Prediction of Corporate Bankruptcy. The Journal of Finance, 23(4): 589–609. https://doi.org/10.1111/j.1540-6261.1968.tb00843.x

Anderson, R. (2007). The Credit Scoring Toolkit - Theory and Practice for Retail Credit Risk Management and Decision Automation. Oxford University Press pp 171. Retrieved from: https://www.researchgate.net/publication/227467693_The_Credit_Scoring_Toolkit_Theory_and_Practice_for_Retail_Credit_Risk_Management_and_Decision_Automation

Ashofteh, A., & Bravo, J. M. (2021). A conservative approach for online credit scoring. Expert Systems with Applications, 176, 1-16. [114835]. https://doi.org/10.1016/j.eswa.2021.114835

Awang, S., & Alimin, N. (2016). The significant factors for the people with epilepsy high employability based on multiple intelligence scores. Malaysian Journal of Fundamental and Applied Sciences. 12. https://doi.org/10.11113/mjfas.v12n1.345

Ayuso, M., Bravo, J. M., Holzmann, R., & Palmer, E. (2021). Automatic Indexation of the Pension Age to Life Expectancy: When Policy Design Matters. Risks, 9(5), 96. MDPI AG. http://doi.org/10.3390/risks9050096

Barboza, F., Kimura, H., & Altman, E. (2017). Machine learning models and bankruptcy prediction. Expert Systems With Applications, 405-417 https://doi.org/10.1016/j.eswa.2017.04.006

Bartual, C., García, F., Guijarro, F, & Romero Civera, A. (2012). Probability of default using the logit model: The impact of explanatory variable and data base selection. International Scientific Conference: Whither our Economics. 118-124. Retrieved from: http://hdl.handle.net/10251/61415

Beaver, W. H. (1966). Financial Ratios As Predictors of Failure. Journal of Accounting Research, 4, 71–111. https://doi.org/10.2307/2490171

Bravo, J. M. (2021). Pricing participating longevity-linked life annuities: A Bayesian Model Ensemble approach. European Actuarial Journal. https://doi.org/10.1007/s13385-021-00279-w

Bravo, J. M., & Ayuso, M. (2020). Mortality and life expectancy forecasts using bayesian model combinations: An application to the portuguese population. RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação, E40, 128–144. https://doi.org/10.17013/risti.40.128–145.

Bravo, J. M., & Ayuso, M. (2021). Forecasting the retirement age: A Bayesian Model Ensemble Approach. Advances in Intelligent Systems and Computing, Volume 1365 AIST, 123 – 135 [2021 World Conference on Information Systems and Technologies, WorldCIST 2021] Springer, Cham. https://doi.org/10.1007/978-3-030-72657-7_12.

Bravo, J. M., Ayuso, M., Holzmann, R., & Palmer, E. (2021a). Addressing the Life Expectancy Gap in Pension Policy. Insurance: Mathematics and Economics, 99, 200-221. https://doi.org/10.1016/j.insmatheco.2021.03.025.

Bravo, J. M., & Mekkaoui, N. (2022). Short-Term CPI Inflation Forecasting: Probing with Model Combinations. In: Rocha, A., Adeli, H., Dzemyda, G., Moreira, F. (eds) Information Systems and Technologies. WorldCIST 2022. Lecture Notes in Networks and Systems, vol 468. Springer, Cham. https://doi.org/10.1007/978-3-031-04826-5_56

Breiman, L. (1996). Bagging Predictors. Machine Learning. 24. 123-140. https://doi.org/10.1007/BF00058655

Byun, H., & Lee, S. (2002). Applications of Support Vector Machines for Pattern Recognition: A Survey.. LNCS. 2388. 213-236. https://doi.org/10.1007/3-540-45665-1_17

Chamboko, R., & Bravo, J. M. (2016) On the modelling of prognosis from delinquency to normal performance on retail consumer loans. Risk Manag 18, 264–287. https://doi.org/10.1057/s41283-016-0006-4

Chamboko, R., & Bravo, J. M. (2020). A Multi-State Approach to Modelling Intermediate Events and Multiple Mortgage Loan Outcomes. Risks, 8(2), 64. MDPI AG. http://doi.org/10.3390/risks8020064

Chang, A., Yang, L., Tsaih, R., & Lin, S. (2022). Machine learning and artificial neural networks to construct P2P lending credit-scoring model: A case using Lending Club data. Quantitative Finance and Economics. 6. 303-325. https://doi.org/10.3934/QFE.2022013.

Chawla, N., Bowyer, K., Hall, L., & Kegelmeyer, W. (2002). SMOTE: Synthetic Minority Over-sampling Technique. J. Artif. Intell. Res. (JAIR). 16. 321-357. https://doi.org/10.1613/jair.953

Christen, P. (2008). Automatic record linkage using seeded nearest neighbor and support vector machine classification. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 151-159. https://doi.org/10.1145/1401890.1401913

Chopra, A., & Bhilare, P. (2018). Application of Ensemble Models in Credit Scoring Models. Business Perspectives and Research. 6. https://doi.org/10.1177/2278533718765531

Cortes, C., & Vapnik, V. (1995). Support Vector Network. Machine Learning. 20. 273-297. https://doi.org/10.1007/BF00994018

Cox, D. R. (1958). The Regression Analysis of Binary Sequences. Journal of the Royal Statistical Society. Series B (Methodological). 20, 2 215-232 https://doi.org/10.1111/j.2517-6161.1958.tb00292.x

Dietterich, T.G. (2000). Ensemble methods in machine learning. Multiple Classifier Systems: First International Workshop, MCS 2000, Lecture Notes in Computer Science. 1-15. https://doi.org/10.1007/3-540-45014-9_1

Estabrooks, A., & Japkowicz, N. (2001). A Mixture-of-Experts Framework for Learning from Imbalanced Data Sets. 2189. 34-43. https://doi.org/10.1007/3-540-44816-0_4

Efron, B., & Tibshirani, R. J. (1993). An Introduction to the Bootstrap. New York: Chapman & Hall. http://doi.org/10.1007/978-1-4899-4541-9

Freund, Y. (1995) Boosting a weak learning algorithms by majority. Information and Computation 121, 256–285. https://doi.org/10.1006/inco.1995.1136

Haji, S., & Mohammad, R. (2015). Eye Tracking with EEG life-style. Available at: https://www.researchgate.net/publication/322358139_Eye_Tracking_with_EEG_life-style

Hansen, P. R., Lunde, A., & Nason, J. M. (2011). The Model Confidence Set. Econometrica, 79(2), 453–497. http://doi.org/10.2139/ssrn.522382

Hayden, E., Stomper, A., & Westerkamp, A. (2010). Selection vs. Averaging of Logistic Credit Risk Models. Journal of Risk. 16. 10.2139. http://doi.org/10.2139/ssrn.1724124

Hu, X., Mei, H., Zhang, H., Li, Y., & Li, M. (2021). Performance evaluation of ensemble learning techniques for landslide susceptibility mapping at the Jinping county, Southwest China. Natural Hazards 105, 1663–1689. https://doi.org/10.1007/s11069-020-04371-4

İlter, D., Kocadağlı, O., & Ravishanker, N. (2019). Feature Selection Approaches for Machine Learning Classifiers on Yearly Credit Scoring Data. Recent Advances in Data Science and Business Analytics, 200–204. Retrieved from: https://www.researchgate.net/publication/341381322_Feature_Selection_Approaches_for_Machine_Learning_Classifiers_on_Yearly_Credit_Scoring_Data

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning with Applications in R (8th ed.). Springer New York Heidelberg Dordrecht London. pp. 215-219. https://doi.org/10.1007/978-1-4614-7138-7

Jiang, W., Chen, Z., Xiang, Y., Shao, D., Ma, L., & Zhang, J. (2019). Ssem: A novel self-adaptive stacking ensemble model for classification. IEEE Access, 7, 120337–120349. https://doi.org/10.1109/ACCESS.2019.2933262

Jijo, B. T., & Abdulazeez, A. M. (2021). Classification based on decision tree algorithm for machine learning. Journal of Applied Science and Technology Trends, 2(01), 20-28. https://doi.org/10.38094/jastt20165

Khanh, T., Duong, B., Quang-Linh, T., Ân, L., Nguyen, A., & Nguyen, K. (2021). Machine Learning-Based Empirical Investigation for Credit Scoring in Vietnam's Banking. https://doi.org/10.1007/978-3-030-79463-7_48.

Kim J-Y, & Cho S-B. (2019) Towards Repayment Prediction in Peer-to-Peer Social Lending Using Deep Learning. Mathematics. 7(11):1041. https://doi.org/10.3390/math7111041

Kirchner, A., & Signorino, C. S. (2018). Using Support Vector Machines for Survey Research. Survey Practice, 11(1), 1–14. https://doi.org/10.29115/sp-2018-0001

Lee, T., & Chen, I. (2003). Mining the Customer Credit Using Classification and Regression Tree and Multivariate Adaptive Regression Splines.. Proceedings of the International Conference on

Information and Knowledge Engineering. 2. 533-538. https://doi.org/10.1016/j.csda.2004.11.006

Lee, Y., & Song, J. (2021). Robustness of model averaging methods for the violation of standard linear regression assumptions. Communications for Statistical Applications and Methods. 28. 189-204. https://doi.org/10.29220/CSAM.2021.28.2.189

Loh, Wei-Yin. (2011). Classification and Regression Trees. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 1. 14 - 23. https://doi.org/10.1002/widm.8

Leung, K., & Parker, D.S. (2003). Empirical comparisons of various voting methods in bagging. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 595-600. https://doi.org/10.1145/956750.956825

Madigan, D., & Raftery, A. E. (1994). Model Selection and Accounting for Model Uncertainty in Graphical Models Using Occam's Window. Journal of the American Statistical Association, 89(428), 1535–1546. https://doi.org/10.2307/2291017

Martin, D. (1977). Early warning of bank failure: A logit regression approach. Journal of Banking & Finance, 1(3), 249-276. https://doi.org/10.1016/0378-4266(77)90022-X

Mester, L. (1997). What's the point of credit scoring?. Business Review, Federal Reserve Bank of Philadelphia, issue Sep, 3-16. Retrieved from: https://www.researchgate.net/publication/5051659_What_Is_the_Point_of_Credit_Scoring

Mienye, D., & Sun, Y. (2021). Performance analysis of cost-sensitive learning methods with application to imbalanced medical data. Informatics in Medicine Unlocked, 25, 1-10. http://doi.org/10.1016/j.imu.2021.100690.

Mishra, S., Sarkar, U., Taraphder, S., Datta, S., Swain, D., & Saikhom, R. et al. (2017). Multivariate Statistical Data Analysis- Principal Component Analysis (PCA). International Journal of Livestock Research, 7(5), 60-78. http://doi.org/10.5455/ijlr.20170415115235

Mohammed, R., Rawashdeh, J., & Abdullah, M. (2020). Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results. 2020 11th International Conference on Information and Communication Systems (ICICS), 243-248. http://doi.org/10.1109/ICICS49469.2020.239556

Mukid, M., Widiharih, T., Rusgiyono, A., & Prahutama, A. (2018). Credit scoring analysis using weighted k nearest neighbor. Journal of Physics: Conference Series. 1025. 012114. https://doi.org/10.1088/1742-6596/1025/1/012114

Ohlson, J. A. (1980). Financial Ratios and the Probabilistic Prediction of Bankruptcy. Journal of Accounting Research, 18(1), 109–131. https://doi.org/10.2307/2490395

Olson, David & Delen, Dursun & Meng, Yanyan. (2012). Comparative analysis of data mining methods for bankruptcy prediction. Decision Support Systems. 52. 464-473. https://doi.org/10.1016/j.dss.2011.10.007.

Quinlan, J. R. (1986). Induction of decision trees. Machine Learning, 1(1), 81–106 https://doi.org/10.1007/BF00116251

Quinlan, J. R. (1993). C4.5: Programs for machine learning. Morgan Kaufmann Publishers. https://doi.org/10.1007/BF00993309

Park, S., Goo, J., & Jo, C. (2004). Receiver Operating Characteristic (ROC) Curve: Practical Review for Radiologists. Korean journal of radiology : official journal of the Korean Radiological Society. 5. 11-8. https://doi.org/10.3348/kjr.2004.5.1.11

Pradhan, A. (2012). Support vector machine - A survey. IJETAE. 2. Retrieved from: https://www.researchgate.net/publication/275331297_Support_vector_machine-A_survey

Powers, D. (2008). Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation. Mach. Learn. Technol.. 2. https://doi.org/10.48550/arXiv.2010.16061

Re, M., & Valentini, G. (2012). Ensemble methods: A review. Advances in Machine Learning and Data Mining for Astronomy. Chapman & Hall 563-594 https://doi.org/10.1201/B11822-34

Rokach, L. (2010). Ensemble-based classifiers. Artif. Intell. Rev.. 33. 1-39. https://doi.org/10.1007/s10462-009-9124-7

Saunders, A., & Allen, L. (2002). Credit Risk Measurement-New Approaches to Value at Risk and Other Paradigms. Published by John Wiley & Sons, Inc., New York 10-11. Retrieved from: https://www.researchgate.net/publication/239487772_Credit_Risk_MeasurementNew_Approaches_to_Value_at_Risk_and_Other_Paradigms

Saunders, A., & Cornett, M. M. (2008). Financial Institutions Management, 6th Edition, McGraw-Hill/Irwin, pp 173. Retrieved from: https://www.academia.edu/27866946/Financial_Institutions_Management

Schapire, R. (1990). The Strength of Weak Learnability. Machine Learning - ML. 5. 28-33. https://doi.org/10.1007/BF00116037

Skogsvik, K. (1990). Current cost accounting ratios as predictors of business failure: The Swedish case. Journal of Business Finance & Accounting. 17. 137 - 160. https://doi.org/10.1111/j.1468-5957.1990.tb00554.x

Sousa, M., Gama, J., & Brandão, E. (2015). A new dynamic modeling framework for credit risk assessment. Expert Systems with Applications. 45. https://doi.org/10.1016/j.eswa.2015.09.055

Steel, M. F. J. (2020). Model Averaging and Its Use in Economics. Journal of Economic Literature, 58, 644–719. https://doi.org/10.1257/jel.20191385

Tahmid, N., Haque, N., Faruque, U., Keya, M., Khushbu, S., & Marouf, A. (2021). A Concern of Predicting Credit Recovery on Supervised Machine Learning Approaches. 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), 1-5, https://doi.org/10.1109/ICCCNT51525.2021.9579706

Thomas, L. (2000). A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers. International Journal of Forecasting 16: 149-172. https://doi.org/10.1016/S0169-2070(00)00034-0

Tian, Z., Xiao, J., Feng, H., & Wei, Y. (2020). Credit Risk Assessment based on Gradient Boosting Decision Tree. Procedia Computer Science. 174. 150-160. https://doi.org/10.1016/j.procs.2020.06.070

Tripathi, D., Edla, D., Kuppili, V., Bablani, A., & Ramesh, D. (2018). Credit Scoring Model based on Weighted Voting and Cluster based Feature Selection. Procedia Computer Science. 132. 22-31. https://doi.org/10.1016/j.procs.2018.05.055

Trivedi, S. (2020). A study on credit scoring modeling with different feature selection and machine learning approaches. Technology in Society. 63. https://doi.org/10.1016/j.techsoc.2020.101413

Turjo, A., Rahman, Y., Karim, M., Biswas, T., & Dewan, I., & Hossain, M. (2021). CRAM: A Credit Risk Assessment Model by Analyzing Different Machine Learning Algorithms. 4th International Conference on Information and Communications Technology (ICOIACT), 125-130, https://doi.org/10.1109/ICOIACT53268.2021.9563995

Vojtek, M., & Kocenda, E. (2006). Credit scoring methods. Finance a Uver - Czech Journal of Economics and Finance. 56. 152-167. Retrieved from: https://www.researchgate.net/publication/285873211_Credit_scoring_methods

Wang, G. , Hao, J. , Ma, J. , & Jiang, H. (2011). A comparative assessment of ensemble learning for credit scoring. Expert Systems with Applications, 38 (1), 223–230 . https://doi.org/10.1016/j.eswa.2010.06.048

Wang, X., Yan, L., & Zhang, Q (2021). Research on the Application of Gradient Descent Algorithm in Machine Learning, International Conference on Computer Network, Electronic and Automation (ICCNEA), 11-15. https://doi.org/10.1109/ICCNEA53019.2021.00014

Witten, I., & Frank, I.H. (2002). Data Mining - Practical Machine Learning Tools and Techniques with JAVA Implementations. Morgan Kaufmann. 31. pp 103 https://doi.org/10.1016/C2009-0-19715-5

Wolpert, D. (1992). Stacked Generalization. Neural Networks. 5. 241-259. https://doi.org/10.1016/S0893-6080(05)80023-1.

Wu, Y., & Pan, Y. (2021). Application Analysis of Credit Scoring of Financial Institutions Based on Machine Learning Model. Complexity. 2021. https://doi.org/10.1155/2021/9222617

Yao, Y., Vehtari, A., Simpson, D., & Gelman, A. (2017). Using Stacking to Average Bayesian Predictive Distributions. Bayesian Analysis. 13. https://doi.org/10.1214/17-BA1091.

Yoonsuh J. (2018). Multiple predicting K-fold cross-validation for model selection, Journal of Nonparametric Statistics, 30:1, 197-215. http://doi.org/10.1080/10485252.2017.1404598

Yu, H., Sun, C., Zheng, S., Wang Q., & Xi, X. (2018). LW-ELM: A Fast and Flexible Cost-Sensitive Learning Framework for Classifying Imbalanced Data. 6 28488 - 28500. https://doi.org/10.1109/ACCESS.2018.2839340

Zavgren, C.V. (1985). Assessing the vulnerability to failure of American industrial firms: A logistic analysis. Journal of Business Finance & Accounting, 12, 19-45. https://doi.org/10.1111/j.1468-5957.1985.tb00077.x

Zhang, D., Zhou, X., Leung, S., & Zheng, J. (2010). Vertical bagging decision trees model for credit scoring. Expert Syst. Appl. 37. 7838-7843. https://doi.org/10.1016/j.eswa.2010.04.054

Zhang, S., Li, X., Zong, M., Zhu, X., & Cheng, D. (2017). Learning k for kNN Classification. ACM Transactions on Intelligent Systems and Technology. 8. 1-19. https://doi.org/10.1145/2990508

Zhang, Y., & Jianxue, W. (2016). K-nearest neighbors and a kernel density estimator for GEFCom2014 probabilistic wind power forecasting. International Journal of Forecasting. 32. https://doi.org/10.1016/j.ijforecast.2015.11.006

Zhou, Z. (2012). Ensemble Methods: Foundations and Algorithms. Chapman and Hall. 15-16 https://doi.org/10.1201/b12207

Zhu, L., Qiu, D., Ergu, D., Ying, C., & Liu, K. (2019). A study on predicting loan default based on the random forest algorithm. Procedia Computer Science. 162. 503-513. https://doi.org/10.1016/j.procs.2019.12.017

Zmijewski, M. (1984). Methodological Issues Related to Estimation of Financial Distress Prediction Models. Journal of Accounting Research. 22. https://doi.org/10.2307/2490859

8. ANNEX

Variable	Description
acc_now_delinq	The number of accounts on which the borrower is now delinquent.
acc_open_past_24mths	Number of trades opened in past 24 months.
addr_state	The state provided by the borrower in the loan application
all_util	Balance to credit limit on all trades
annual_inc	The self-reported annual income provided by the borrower during registration.
annual_inc_joint	The combined self-reported annual income provided by the co-borrowers during registration
application_type	Indicates whether the loan is an individual application or a joint application with two co-borrowers
avg_cur_bal	Average current balance of all accounts
bc_open_to_buy	Total open to buy on revolving bankcards.
bc_util	Ratio of total current balance to high credit/credit limit for all bankcard accounts.
chargeoff_within_12_mths	Number of charge-offs within 12 months
collection_recovery_fee	post charge off collection fee
collections_12_mths_ex_med	Number of collections in 12 months excluding medical collections
debt_settlement_flag	Flags whether or not the borrower, who has charged-off, is working with a debt-settlement company.
deferral_term	Amount of months that the borrower is expected to pay less than the contractual monthly payment amount due to a hardship plan
delinq_2yrs	The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years
delinq_amnt	The past-due amount owed for the accounts on which the borrower is now delinquent.
dti	A ratio calculated using the borrower’s total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower’s self-reported monthly income.
dti_joint	A ratio calculated using the co-borrowers' total monthly payments on the total debt obligations, excluding mortgages and the requested LC loan, divided by the co-borrowers' combined self-reported monthly income
earliest_cr_line	The month the borrower's earliest reported credit line was opened
emp_length	Employment length in years. Possible values are between 0 and 10 where 0 means less than one year and 10 means ten or more years.
emp_title	The job title supplied by the Borrower when applying for the loan.

Variable	Description
funded_amnt	The total amount committed to that loan at that point in time.
funded_amnt_inv	The total amount committed by investors for that loan at that point in time.
grade	LC assigned loan grade
hardship_amount	The interest payment that the borrower has committed to make each month while they are on a hardship plan
hardship_dpd	Account days past due as of the hardship plan start date
hardship_end_date	The end date of the hardship plan period
hardship_flag	Flags whether or not the borrower is on a hardship plan
hardship_last_payment_amount	The last payment amount as of the hardship plan start date
hardship_length	The number of months the borrower will make smaller payments than normally obligated due to a hardship plan
hardship_loan_status	Loan Status as of the hardship plan start date
hardship_payoff_balance_amount	The payoff balance amount as of the hardship plan start date
hardship_reason	Describes the reason the hardship plan was offered
hardship_start_date	The start date of the hardship plan period
hardship_status	Describes if the hardship plan is active, pending, canceled, completed, or broken
hardship_type	Describes the hardship plan offering
home_ownership	The home ownership status provided by the borrower during registration or obtained from the credit report. Our values are: RENT, OWN, MORTGAGE, OTHER
id	A unique LC assigned ID for the loan listing.
il_util	Ratio of total current balance to high credit/credit limit on all install acct
initial_list_status	The initial listing status of the loan. Possible values are – W, F
inq_fi	Number of personal finance inquiries
inq_last_12m	Number of credit inquiries in past 12 months
inq_last_6mths	The number of inquiries in past 6 months (excluding auto and mortgage inquiries)
installment	The monthly payment owed by the borrower if the loan originates.
int_rate	Interest Rate on the loan
issue_d	The month which the loan was funded
last_credit_pull_d	The most recent month LC pulled credit for this loan
last_fico_range_high	The upper boundary range the borrower’s last FICO pulled belongs to.

Variable	Description
last_fico_range_low	The lower boundary range the borrower’s last FICO pulled belongs to.
last_pymnt_amnt	Last total payment amount received
last_pymnt_d	Last month payment was received
loan_amnt	The listed amount of the loan applied for by the borrower. If at some point in time, the credit department reduces the loan amount, then it will be reflected in this value.
loan_status	Current status of the loan
max_bal_bc	Maximum current balance owed on all revolving accounts
mo_sin_old_il_acct	Months since oldest bank installment account opened
mo_sin_old_rev_tl_op	Months since oldest revolving account opened
mo_sin_rcnt_rev_tl_op	Months since most recent revolving account opened
mo_sin_rcnt_tl	Months since most recent account opened
mort_acc	Number of mortgage accounts.
mths_since_last_delinq	The number of months since the borrower's last delinquency.
mths_since_last_major_derog	Months since most recent 90-day or worse rating
mths_since_last_record	The number of months since the last public record.
mths_since_rcnt_il	Months since most recent installment accounts opened
mths_since_recent_bc	Months since most recent bankcard account opened.
mths_since_recent_bc_dlq	Months since most recent bankcard delinquency
mths_since_recent_inq	Months since most recent inquiry.
mths_since_recent_revol_delinq	Months since most recent revolving delinquency.
next_pymnt_d	Next scheduled payment date
num_accts_ever_120_pd	Number of accounts ever 120 or more days past due
num_actv_bc_tl	Number of currently active bankcard accounts
num_actv_rev_tl	Number of currently active revolving trades
num_bc_sats	Number of satisfactory bankcard accounts
num_bc_tl	Number of bankcard accounts
num_il_tl	Number of installment accounts
num_op_rev_tl	Number of open revolving accounts

Variable	Description
num_rev_accts	Number of revolving accounts
num_rev_tl_bal_gt_0	Number of revolving trades with balance >0
num_sats	Number of satisfactory accounts
num_tl_120dpd_2m	Number of accounts currently 120 days past due (updated in past 2 months)
num_tl_30dpd	Number of accounts currently 30 days past due (updated in past 2 months)
num_tl_90g_dpd_24m	Number of accounts 90 or more days past due in last 24 months
num_tl_op_past_12m	Number of accounts opened in past 12 months
open_acc	The number of open credit lines in the borrower's credit file.
open_acc_6m	Number of open trades in last 6 months
open_act_il	Number of currently active installment trades
open_il_12m	Number of installment accounts opened in past 12 months
open_il_24m	Number of installment accounts opened in past 24 months
open_rv_12m	Number of revolving trades opened in past 12 months
open_rv_24m	Number of revolving trades opened in past 24 months
orig_projected_additional_accrued_iterest	The original projected additional interest amount that will accrue for the given hardship payment plan as of the Hardship Start Date. This field will be null if the borrower has broken their hardship payment plan.
out_prncp	Remaining outstanding principal for total amount funded
out_prncp_inv	Remaining outstanding principal for portion of total amount funded by investors
payment_plan_start_date	The day the first hardship plan payment is due. For example, if a borrower has a hardship plan period of 3 months, the start date is the start of the three-month period in which the borrower is allowed to make interest-only payments.
pct_tl_nvr_dlq	Percent of trades never delinquent
percent_bc_gt_75	Percentage of all bankcard accounts > 75% of limit.
policy_code	publicly available policy_code=1 new products not publicly available policy_code=2
pub_rec	Number of derogatory public records
pub_rec_bankruptcies	Number of public record bankruptcies
purpose	A category provided by the borrower for the loan request.
pymnt_plan	Indicates if a payment plan has been put in place for the loan

Variable	Description
recoveries	post charge off gross recovery
revol_bal	Total credit revolving balance
revol_bal_joint	Sum of revolving credit balance of the co-borrowers, net of duplicate balances
revol_util	Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit.
sec_app_chargeoff_within_12_mths	Number of charge-offs within last 12 months at time of application for the secondary applicant
sec_app_collections_12_mths_ex_md	Number of collections within last 12 months excluding medical collections at time of application for the secondary applicant
sec_app_earliest_cr_line	Earliest credit line at time of application for the secondary applicant
sec_app_fico_range_high	FICO range (low) for the secondary applicant
sec_app_fico_range_low	FICO range (high) for the secondary applicant
sec_app_inq_last_6mths	Credit inquiries in the last 6 months at time of application for the secondary applicant
sec_app_mort_acc	Number of mortgage accounts at time of application for the secondary applicant
sec_app_num_rev_accts	Number of revolving accounts at time of application for the secondary applicant
sec_app_open_acc	Number of open trades at time of application for the secondary applicant
sec_app_open_act_il	Number of currently active installment trades at time of application for the secondary applicant
sec_app_revol_util	Ratio of total current balance to high credit/credit limit for all revolving accounts
sub_grade	LC assigned loan subgrade
tax_liens	Number of tax liens
term	The number of payments on the loan. Values are in months and can be either 36 or 60.
title	The loan title provided by the borrower
tot_coll_amt	Total collection amounts ever owed
tot_cur_bal	Total current balance of all accounts
tot_hi_cred_lim	Total high credit/credit limit
total_acc	The total number of credit lines currently in the borrower's credit file
total_bal_ex_mort	Total credit balance excluding mortgage
total_bal_il	Total current balance of all installment accounts
total_bc_limit	Total bankcard high credit/credit limit
total_cu_tl	Number of finance trades
total_il_high_credit_limit	Total installment high credit/credit limit

Variable	Description
total_pymnt	Payments received to date for total amount funded
total_pymnt_inv	Payments received to date for portion of total amount funded by investors
total_rec_int	Interest received to date
total_rec_late_fee	Late fees received to date
total_rec_prncp	Principal received to date
total_rev_hi_lim	Total revolving high credit/credit limit
url	URL for the LC page with listing data.
verification_status	Indicates if income was verified by LC, not verified, or if the income source was verified
verification_status_joint	Indicates if the co-borrowers' joint income was verified by LC, not verified, or if the income source was verified
zip code	The first 3 numbers of the zip code provided by the borrower in the loan application