VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY

UNIVERSITY OF ECONOMICS AND LAW

GRADUATION THESIS

Topic:

P2P LENDING DEFAULT PREDICTION USING MACHINE LEARNING

Advisor: **MSc. Pham Chi Khoa**

Student: **Bùi Nguyễn Thùy Như**

Student ID: **K194141737**

Class: **K19414C**

HO CHI MINH CITY, APRIL 2023

UNIVERSITY OF ECONOMICS AND LAW

FACULTY OF FINANCE AND BANKING

GRADUATION THESIS

P2P LENDING DEFAULT PREDICTION USING MACHINE LEARNING

Advisor: MSc. Pham Chi Khoa

Student: Bui Nguyen Thuy Nhu

Student ID: K194141737

Class: K19414C

HO CHI MINH CITY, APRIL 2023

ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude to all those who have supported and contributed to the successful completion of my thesis.

First and foremost, I extend my sincerest appreciation to Msc. Pham Chi Khoa, my thesis advisor, for his invaluable assistance, guidance, and unwavering support throughout the entire research process. His insightful feedback, constructive criticism, and vast knowledge in the field have significantly improved the quality of my work.

I would also like to thank the faculty members of my department for providing me with a stimulating academic environment and the necessary resources to conduct my research. Their expertise and willingness to share their knowledge have greatly contributed to the success of this study.

Furthermore, I extend my gratitude to my friends and family for their constant encouragement and support during this challenging journey. Their unwavering faith in me has been a great source of motivation and inspiration.

In conclusion, I am truly grateful for the opportunities, guidance, and support that I have received, which have enabled me to complete this study successfully.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS

No.	Abbreviations	Description
1	AUC	Area under the curve
2	FN	False Negative
3	FP	False Positive
4	FICO	Credit score created by the Fair Isaac Corporation
5	KS	Kolmogorov-Smirnov Statistic
6	NPL	Nonperforming loan management services
7	P2P	Peer-to-peer lending
8	TN	True Negative
9	TP	True Positive
10	XGBoost	Extreme Gradient Boosting

ABSTRACT

Peer-to-peer lenders have revolutionized the credit market by providing an alternative to traditional financial services and utilizing advanced analytics techniques. Accurately assessing a borrower's creditworthiness and credit scoring are critical to managing credit risk and adapting to changing market conditions. Logistic Regression has been the preferred model for credit scoring, but this thesis aims to evaluate and compare its ability to predict loan defaults with other parametric and non-parametric methods in a peer-to-peer lending context. The study uses data from LendingClub, a P2P lending platform with four algorithms were compared, including Decision Trees, Logistic Regression, Random Forest, and XGBoost. This thesis includes a literature review, pre-processing explanation, and model description. The results show that XGBoost and Decision Tree outperform the benchmark's predictive ability, while the Random Forest and Logistic Regression models have weaker performance compared to the benchmark. Therefore, it is still reasonable to use the benchmark model, but modern techniques should also be taken into consideration.

KEY WORDS: Machine learning, Peer-to-peer lending, Credit scoring, Default prediction, LendingClub .

1. INTRODUCTION

1.1 The Research Context

The emergence of web 2.0 technology has resulted in the rapid development of online markets and virtual communities where people can interact virtually to meet their needs. In the online peer-to-peer (P2P) lending market, borrowers and lenders meet virtually through an online platform to process a lending transaction without the use of a formal conventional financial intermediary, similar to the virtual market. In the P2P lending market, lenders and borrowers can split the savings from traditional intermediation costs. However, in the event of a loan default, lenders bear the default risk of the borrowers. To deal with the potential default risk of the borrowers, lenders face an asymmetric information problem in which they lack information about the borrowers, preventing them from making prudent lending decisions, which leads to an adverse selection problem on the part of the lenders. Banking theory suggests that traditional financial intermediaries such as banks and credit unions can mitigate some of the adverse selection issues by hiring expert executives, obtaining guarantees and collateral, and ensuring post-disbursement monitoring (Akerlof, A., 1970). Unlike in the traditional financial market, it is difficult to reduce the effects of information asymmetry in the online P2P market environment due to the high transaction cost and the limitations of the virtual environment.

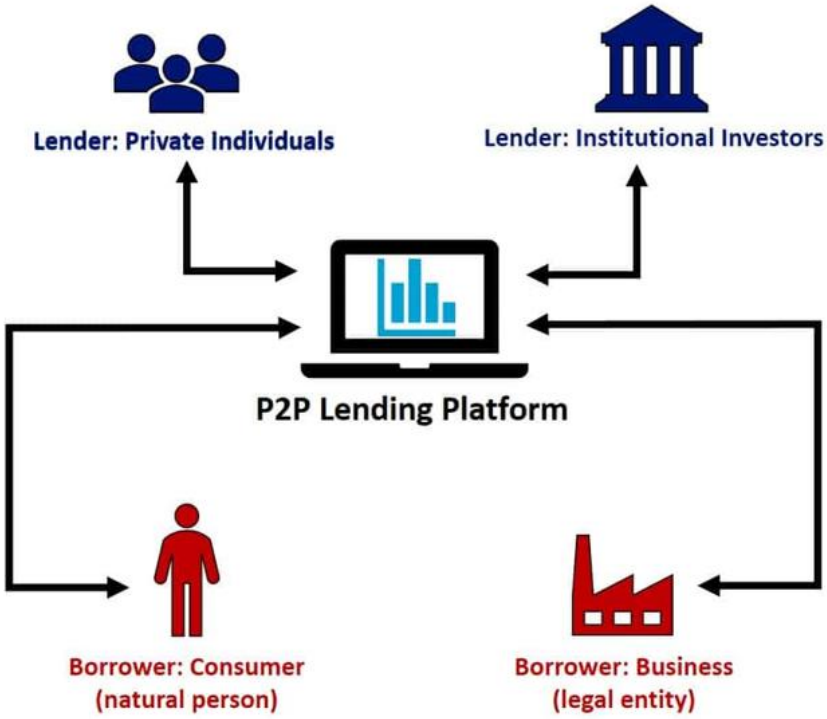

Figure 1.1: P2P Lending Platform Process

Source: P2Pmarketdata.com

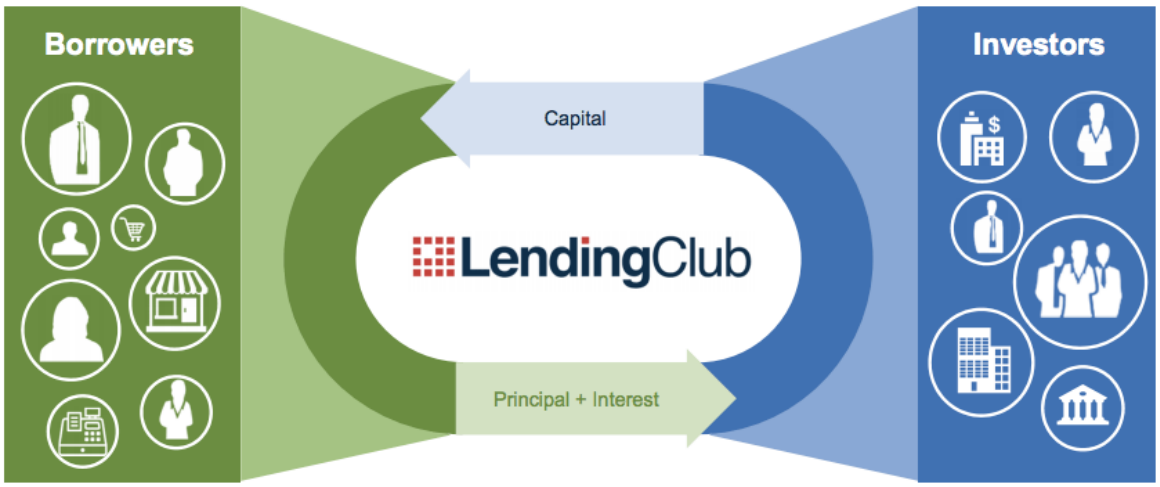

Figure 1.2: P2P Model of Lending Club

Source: LendingClub.com

To reduce the lending risk associated with information asymmetry, lending platforms in the P2P lending market take initiatives to identify trustworthy borrowers. To begin, platforms employ their own screening system to eliminate some potential borrowers based on predefined criteria. For example, in this study, the author use the Lending Club uses a FICO score floor, and customers with FICO scores lower than that cannot be listed on the platform. Second, platforms set a lending limit to reduce individual borrowers' risk exposures. Currently, the lending club sets a maximum limit of USD 35,000 for individual borrowing, allowing investors to spread risk among multiple borrowers. Finally, platforms provide portfolio recommendation services to investors. Finally, platforms provide portfolio recommendation services to investors. Platforms with expertise and scalability can better understand borrowers' risk levels and generate workable recommendation mechanisms. The lending club, like many other lending platforms, assigns credit grades and subgrades to its potential borrowers, which investors use as a recommendation based on the borrowers' risk level. The ultimate investors can mitigate the negative impact of information asymmetry by making lending decisions based on the platform's assigned credit grades. In addition to the foregoing, the lending platform offers nonperforming loan management services (NPLs). They assist investors in engaging in collection agencies to recover NPLs and in obtaining legal services for NPL litigation. Platforms, as market makers employ various tools and methods to reduce the problem of information asymmetry for better risk management of the P2P loans nevertheless the credit risks associated with the P2P loans, have not been eliminated and there are rooms for further improvement of the decision-making capacity of the investors as well as the market makers.

1.2 Urgency of The Topic

With the topic "P2P Lending Default Prediction Using Machine Learning", the author expect the research paper to have scientific value is as follows:

Firstly, the research is the discovery, learning and synthesis of P2P lending forecasting models suitable for the current P2P market, especially on LendingClub.com website, while in other research papers, there is no combination of methods. This contributes to fuller reflection, as well as a more sophisticated algorithmic setup.

Secondly, this study once again confirms the conclusion that loan dataset in the past have had a strong impact on the movements of the P2P lending market in the future, and from that, it is recommended that investors (lenders) should pay attention influence of past prices on the present.

The study has the following practical implications in addition to its academic value:

Firstly, it understands the necessity to invest idle cash flows as well as the challenges that investors encounter. Research is believed to be a useful support tool for novice investors who are still inexperienced, assisting investors in making accurate decisions at all times. Since then, the author has hoped that the technology will help investors generate more income from their idle cash flow and lower the danger of losing money.

Secondly, if a website is fully designed and built, the tool will help to improve the University of Economics and Law's image and name in the pioneering movement of research orientation with high applied works.

1.3 Research Target

This study aims to establish a synthetic algorithm based on machine learning and One of the most important products offered by financial institutions is the loan. All of the institutes are attempting to devise effective business strategies in order to persuade more customers to apply for their loans. However, some customers are unable to repay their loans after their applications are approved. As a result, when approving a loan, many financial institutions consider a number of factors. It is difficult to predict whether a given borrower will fully pay off the loan or cause it to be charged off (not fully pay off the loan). If the lender is too strict, fewer loans are approved, resulting in less interest to collect. Ho wever, if they are too lenient, they end up approving loans that default. Several machine learning models are used to analyze loan behavior in this study. The specific objectives are as follows:

- The objective is to make predictions about loan default and whether investors should lend to a customer or not.

- It creates a variety of distinct models for the default prediction of these loans by identifying the factors that can have a significant impact on the default probability prediction.

- Solve synthetic problems based on machine learning algorithms to forecast P2P default lending.

- Analyze and test forecast models on training dataset and compare them with actual data. From there, draw conclusions as well as create topics for future studies of the author's group.

1.4 Research Objectives and Research Scope

1.4.1 Research Objectives

The objectives of the research paper include the following:

- Machine learning algorithms using input variables as indicators from characteristics and information of the borrower

- The efficiency of the prediction algorithms and model compared to the actual data.

1.4.2 Research Scope

Spatial scope: The study uses secondary data which is historical price including: the information of loan and the characteristics and information of the borrower is taken from LendingClub.com's database.

Time range: The data set is taken from LendingClub.com started for the first time on website to 4th quarter, 2018. This is not the best time period, because most of the loans from that period have already been repaid or defaulted on, with limited empty data and enough background data for the study.

1.5 Research Method

- Documentary research method: includes collecting documents related to machine learning and technical analysis, selecting and analyzing articles related to the topic and presenting a summary of the content. use reference studies.

- Methods of analysis and synthesis: After distilling and studying the documents, the collected data will be analyzed, filtered and synthesized to orient the work, as a basis for carrying out the project.

- Supervised learning method: Using Logistic Regression, Decision Tree, XGBoost, Random Forest algorithms in supervised learning method to build a model from training data divided from a given data set.

1.6 Research Structure

The research paper consists of 5 parts:

1 - Introduction: An overview of the context of P2P lending market, research objectives, object, scope, method of the research and calculation The urgency of the topic brings to the audiences.

2 – Literature review: Reviewing the theoretical foundations of the topic such as machine learning, deep learning, algorithms and a summary of reference studies.

3 – Data and methodology: Describes the steps to carry out the research, how to get and exploit data and describes the variables included in the model.

4 - Research results: Analyze forecast results by accuracy index and by confusion matrix.

5 – Conclusions and recommendations: Overall assessment of the results of the model, concluding that the predictive model is quite stable, thereby making recommendations for additional data files and giving other research directions in the future.

2. LITERATURE REVIEW

2.1 P2P Lending

Peer-to-Peer lending, commonly known as P2P lending, is a relatively new approach to borrowing and lending money. The concept was first introduced in 2005 and has since rapidly gained popularity all over the world. P2P lending is a platform that connects borrowers and lenders directly, bypassing the involvement of financial institutions such as banks in the decision-making process. This approach offers borrowers the possibility of obtaining credit on more favorable terms than those available in the traditional banking system (Bachmann, 2011).

One of the primary advantages of P2P lending is that it provides an online platform that eliminates the brick-and-mortar operating costs associated with traditional banking institutions. This allows P2P lenders to offer lower interest rates to borrowers compared to those charged by banks. As a result, P2P lending has emerged as an alternative way for small businesses and individuals with no credit history to access financing.

However, P2P lending presents certain challenges, particularly with regards to information asymmetry. Lenders are limited to making lending decisions based solely on the information provided by the borrower. This lack of complete information can lead to a higher risk of default, which is a fundamental problem in the P2P lending system.

Despite these challenges, P2P lending continues to grow in popularity worldwide, as it provides a flexible and cost-effective means for borrowers to access credit and for lenders to earn interest on their investments. This paper will further explore the development of P2P lending and its potential benefits and drawbacks.

2.2 P2P Lending Prediction

P2P lending is a form of borrowing that bypasses traditional financial institutions such as banks and credit unions. With good credit (typically a FICO score greater than 720), P2P loans can offer surprisingly low interest rates. Even with less-than-perfect credit, an applicant has a good chance of being approved for a low-interest loan from an online lender like Lending Club.

Unlike traditional loans, P2P loans are made by individuals and investors instead of banks. People who have extra money offer to lend it to others, such as individuals or businesses, who are in need of money. A P2P service, usually in the form of a website, connects lenders and borrowers to simplify the process for everyone involved.

Loan default prediction is a common issue for P2P lenders. This is similar to the challenges faced by banks and credit card companies when customers request a loan. The focus of this study is on the Lending Club dataset, which is freely available on their website. The aim of the study is to forecast loan default and determine whether or not investors should lend to a particular customer.

By developing a predictive model for loan default, this study can help mitigate the risks associated with P2P lending for both borrowers and lenders. The results of this study can provide valuable insights into the development of new models to enhance the efficiency and accuracy of P2P lending systems.

2.3 Machine Learning

Machine learning is a field of study in statistics, artificial intelligence, and computer science that leverages the computing power of computers for predictive analysis. The application of machine learning methods in recent years has become commonplace in everyday life. From automated recommendations for movies to see, food to order or products to buy, to personalized online radio and friend recognition in your photos, many modern websites and devices have machine learning algorithms at the core. Besides applications in real life, this field is also widely applied in the field of data processing.

Machine learning is gradually becoming an important tool in statistics and analysis due to the dramatic increase in data volume, exponential growth in computing power, and advances in algorithm design, driven by the growing demand in the web development field. In general, the goal of machine learning is to determine the general characteristics and rules of the input data set so that it is possible to determine the characteristics of other data sets. It is important to find a appreciate adjusting ways to approach the characteristics of the data, whether it is a collection of images, time series signals, or general descriptive data. In general, machine learning algorithms and methods are only one part of a process to solve a particular problem, so what many analysts recommend is that the processor

should have an overview of the situation so that when it comes to solving a particular problem, the operator can choose the right method suitable for his data.

Single classifier machine learning approaches aim to learn a single hypothesis from training data, while ensemble methods seek to construct a set of hypotheses and combine them (Zhou, 2013). This fundamental difference sets the two approaches apart, with the latter being able to incorporate a variety of base learners, including neural networks and other learning algorithms, to improve prediction accuracy and robustness.

Ensemble approaches are widely considered superior to single classifier machine learning approaches because the training data may not provide sufficient information to select a single best learner and to minimize the search process's deficiencies (Zhou, 2013). By combining multiple base learners, ensemble methods can improve accuracy, reduce variance, and achieve better generalization performance than a single model (Kuncheva, 2014). However, it is important to note that the use of an ensemble approach can result in a lack of comprehensibility of the knowledge acquired, making it difficult to interpret the results and understand the decision-making process (Rokach, 2010). Therefore, a trade-off between performance and interpretability should be considered when choosing between single and ensemble methods.

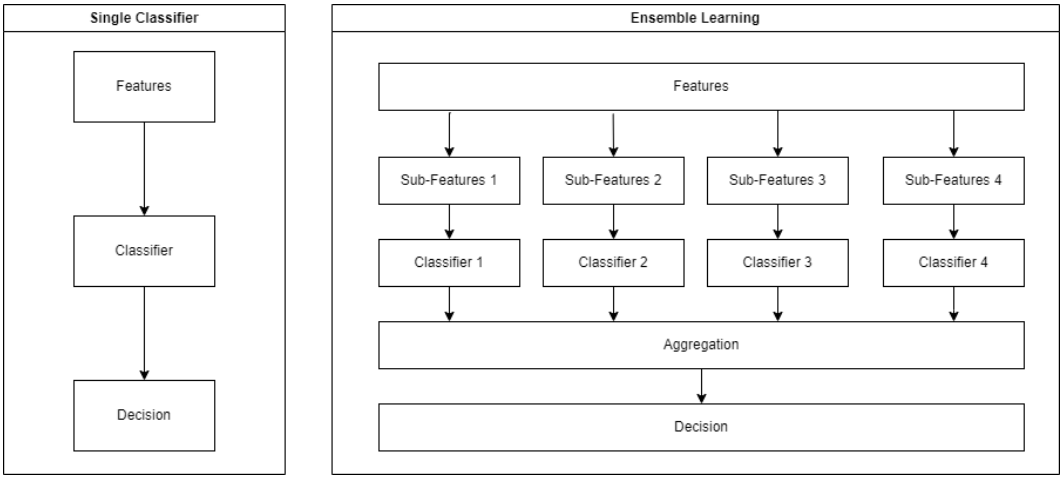

Figure 2.1: Single Classifier vs Ensamble Learning

Source: Author

2.4 Single Classifiers

2.4.1 Logistic Regression

Logistic Regression (Cox, 1958) is a widely-used parametric method in credit scoring among financial institutions due to its interpretability and direct prediction of probabilities. Linear Regression, on the other hand, aims to estimate the parameters so that the sum of the squared errors of a function of the sum of the squared error can be divided into components of model variance and bias (Kuhn & Johnson, 2013). However, its expressiveness is limited, and interactions must be added manually. In addition, its interpretation can be more complex than other models such as Decision Trees because the weights are multiplicative and not additive. Logistic Regression, in contrast, is an iterative and calculation-intensive methodology that typically needs about six training epochs to reach convergence, using one of several convergence criteria (Garson, 2012).

Logistic Regression is the preferred method for developing credit-scoring models (Lessmann et al., 2015), mainly because it ensures that the final probability falls between 0 and 1 and provides a relatively robust estimate of the actual likelihood, given available information (Anderson, 2007). Logistic Regression models the relationship between one or more independent variables (predictors) and categorical variables (output) by using a logistic function to assess the probabilities. Logistic Regression can be binary (with two possible outcomes), multinomial (with three or more categories without ordering), or ordinal (with three or more classes with ordering) (Akindaini, 2017).

In this section, I will focus on fitting logistic regression into the loan data. I will determine the optimal parameter for logistic regression, which is crucial for developing a robust predictive model. By doing so, I aim to identify the key variables that influence the probability of loan default, and hence, facilitate decision-making for lenders.

Kuhn and Johnson (2013) suggest that how it will be applied to the loan data. The Logistic Regression formula is a parametric model used to model the relationship between one or more independent variables and a binary response variable. The formula can be written as:

$$P = \frac{e^{a+bX}}{1+e^{a+bX}} \backslash$$

where Y is the binary response variable (1 for default, 0 for fully paid), X is a vector of independent variables (explanatory variables), and z is the linear predictor defined as:

$$z = \beta 0 + \beta 1X1 + \beta 2X2 + \ldots + \beta kXk$$

where $\beta 0$ is the intercept and Xj (j = 1, … , k) are the explanatory variables with βj (j = 1, … , k) as the corresponding coefficients to be estimated from regressing the model on data. The logistic function transforms the linear predictor z into a probability value between 0 and 1, representing the probability of observing the event (default) given the values of the independent variables. The formula is widely used in credit scoring models to predict loan default and estimate the probability of default based on borrower's financial and personal information.

2.4.2 Decision Tree

The Decision Tree algorithm is a type of supervised learning algorithm that can be used for both regression and classification problems. The primary objective of using a Decision Tree is to develop a training model that can predict the target variable's class or value by utilizing simple decision rules learned from prior data (training data). The Decision Tree method is highly suitable for credit scoring modeling and has been widely employed (Lee & Chen, 2005).

In Decision Trees, the prediction of a class label for a record begins at the root of the tree. The values of the root attribute are compared with the record's attribute. Based on the comparison, the algorithm follows the branch corresponding to that value and moves to the next node.

In Decision Trees, for predicting a class label for a record, the root node contains a set of good and bad credit applications. The algorithm then attempts all possible binary splits to find the attribute x and corresponding cut-off value that best discriminates individuals based on the class they belong to. This procedure is repeated for the new nodes until a stopping criterion is met. To comprehend the splitting criteria, understanding the

concepts of entropy and information gain is critical. The entropy is generally used to assess the quality of the split when constructing a classification tree (James et al., 2013).

The entropy in Decision Trees is a measure of the impurity or disorder in a set of examples, and it is defined as the average amount of information required to classify an example. James et al. (2013) define entropy mathematically as:

$$D = -\sum_{k=1}^{K} \hat{p}_{mk} \, log \hat{p}_{mk}$$

where m is the number of classes, and p_i is the proportion of examples belonging to class i. The entropy is 0 when all examples in the set belong to the same class (i.e., the set is pure), and it is maximum when the classes are equally distributed (i.e., the set is completely impure). The entropy is used in Decision Trees to measure the reduction in impurity that results from splitting the data using a particular attribute. The information gain is then calculated as the difference between the entropy of the parent node and the weighted average of the child nodes' entropies, where the weights are proportional to the number of examples in each child node.

2.5 Ensemble Learning

2.5.1 Random Forest

Random Forest is a type of ensemble classifier that was introduced by Breiman in the early 2000s. This classifier is based on Decision Trees as its base learners. During the training phase, the model is trained on a dataset that is randomly sampled with replacement (bootstrap samples), and each tree generated is trained on a randomly selected subset of features (Brown & Mues, 2012).

Random Forest is a widely used supervised learning algorithm that aims to overcome the limitations of the Decision Tree algorithm, such as overfitting and underfitting. The algorithm builds multiple Decision Trees independently and randomly selects a subset of features to build each tree. This helps to disperse the errors and reduce the overfitting coefficient while maintaining accurate predictions on the dataset. The Random Forest algorithm provides important predictive features by summing the features on decision

trees, which are evaluated as more reliable than those provided by an individual Decision Tree model.

In a Random Forest, each tree generates a class prediction, and the class with the majority of votes becomes the final prediction of the model. This approach can be seen as the wisdom of crowds. To ensure that the trees in the forest are diverse and their predictions are uncorrelated (Yiu, 2019), it is important to ensure that the trees are trained on different subsets of the data and that different subsets of features are used in each tree. This ensures that the trees have different decision boundaries and that features with predictive power are included in the model.

2.5.1 Extreme Gradient Boosting (XGBoost)

Chen and Guestrin (2016) proposed an ensemble algorithm based on decision trees that uses a gradient boosting algorithm, which has gained popularity and is considered an efficient open-source implementation of this algorithm. Extreme Gradient Boosting (XGBoost) has been responsible for winning numerous Kaggle competitions and achieving state-of-the-art results in various real-world applications. It is highly scalable in all scenarios since it can handle various data types, relationships, distributions and fine-tune a variety of hyperparameters, and has multiple applications in regression, classification and ranking problems (Chen & Guestrin, 2016).

In contrast to Random Forest, Extreme Gradient Boosting uses boosting, which sequentially combines weak learners, usually Decision Trees with only one split - decision stumps, so that each new tree corrects the errors of the previous tree (Dhingra, 2020). The algorithm starts with one Decision Tree, and using a loss function, such as Cross entropy or Logarithmic loss, the performance of the tree is evaluated. The loss function penalizes false classifications by considering the probability of classifications. Cross entropy is a similar metric, and the loss associated with it increases as the predicted probability diverges from the actual label (Saha, 2018). After completing the first tree and evaluating the loss function, the next tree is added to lower the loss more than the first tree alone. The core problem of XGBoost is to determine the optimal tree structure, employing a greedy search algorithm to achieve this (Xia et al., 2017).

2.6 Performance Metrics

In credit scoring applications, various performance evaluation criteria are used to assess the efficacy of predictive models. ElMasry (2019) categorizes these measures into three types: discriminatory ability, accuracy of probability predictions, and correctness of predictions. The area under the curve (AUC) is one of the most commonly used measures for prediction models with binary outcomes.

To assess the correctness of predictions, two measures are often used: the Kolmogorov-Smirnov Statistic (KS) and the Percent Correctly Classified (PCC). The introduction of the confusion matrix is necessary to explain these measures. The confusion matrix is a table that summarizes the predicted and actual classifications. The PCC is the proportion of observations that are classified correctly, and it is calculated using the true positive (TP) and true negative (TN) rates. If the predicted classification does not correspond to the actual classification, the observation is labeled as a false positive (FP) or a false negative (FN), resulting in a Type I or Type II error, respectively.

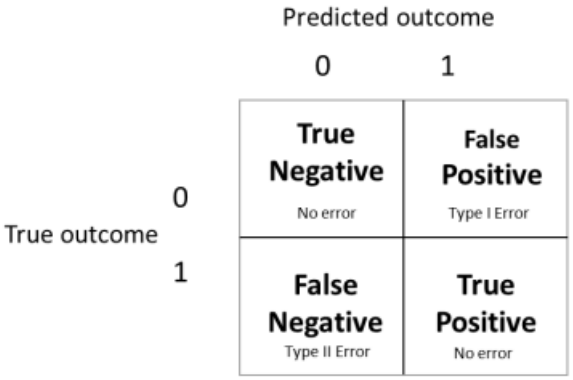

Figure 2.2: Confusion Matrix

Source: Author

- True Positive (TP): A classification outcome where the true value is 1 and the predicted value is also 1. This means that the borrower defaulted and the model correctly predicted this outcome.
- True Negative (TN): A classification outcome where the true value is 0 and the predicted value is also 0. This means that the borrower fully paid their loan and the model correctly predicted this outcome.

- False Positive (FP): A classification outcome where the true value is 0 and the predicted value is 1. This means that the borrower fully paid their loan, but the model predicted that they would default.
- False Negative (FN): A classification outcome where the true value is 1 and the predicted value is 0. This means that the borrower defaulted, but the model predicted that they would fully pay their loan.

In this study, the author used confusion matrix, precision, recall and F1-score.

For classification model, both Charged Off are assigned label 1 and Fully Paid is assigned label 0.

For metrics to evaluate classification performance, the author use confusion matrix whose columns represent predicted values and rows represent true values. Also measure precision, recall, f1-score (the harmonic mean of precision and recall) and the authorighted average as defined below:

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

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

F1-score =$\frac{2TP}{2TP+FP+FN}$

Support = the number of true instances for each label

Weighted-avg metric = metric of weighted by support

2.7 Reference Research

2.7.1 Risk assessment in social lending via random forests

Due to the rise of electronic commerce and social media, social lending (also known as peer-to-peer lending) has become a feasible platform where borrowers and lenders can conduct transactions without the assistance of institutional intermediaries such as banks. Social lending has gained significant momentum in recent times, with some platforms having reached multi-billion dollar loan circulation in a short period. However, the long-term sustainability and widespread adoption of these platforms rely heavily on trustworthy risk assessment of individual borrowers. To address this issue, Malekipirbazari and

Aksakalli (2015) suggest a random forest (RF) based classification approach for forecasting the status of borrowers. Our analysis on data obtained from the well-known social lending platform Lending Club (LC) reveals that the RF-based method outperforms both FICO credit scores and LC grades in the identification of reliable borrowers.

2.7.2 Machine learning applications in mortgage default prediction

The task of determining default risk has posed a significant challenge in credit-risk analysis. Financial institutions are interested in assessing a customer's ability to repay a loan. In this study, Akindaini (2017) investigates the application of various machine learning models for predicting mortgage defaults. Their focus is on exploring how machine learning techniques can be utilized to classify mortgages into categories such as paying, default, and prepay. The machine learning models examined in this research include Logistic Regression (both simple and multi-class), Naive Bayes, Random Forest, and K-Nearest Neighbors. Additionally, the authors incorporate Survival analysis and Cox proportional hazard rate to evaluate the likelihood of loan survival over a specific period and to determine the impact of each variable in predicting the probability of survival.

2.7.3 Machine learning approach for credit score analysis

In order to enhance credit score analysis, financial institutions have implemented techniques and models aimed at improving the assessment of creditworthiness during the credit evaluation process. The primary objective is to classify clients - borrowers - as either non-defaulters, who are more likely to meet their financial obligations, or defaulters, who have a higher probability of failing to pay their debts. In this study, ElMasry (2019) utilize machine learning models to predict mortgage defaults. They employ various single classification machine learning methods, such as Logistic Regression, Classification and Regression Trees, Random Forest, K-Nearest Neighbors, and Support Vector Machine. To further enhance the predictive power, the authors introduce a meta-algorithm ensemble approach, known as stacking, to combine the outputs (probabilities) of the above-mentioned methods. The sample used in this study is based solely on the publicly available dataset provided by Freddie Mac. Through this approach, the authors achieve an improvement in the model's predictive performance. They compare the performance of each model, and the meta-learner, by plotting the ROC Curve and computing the AUC

rate. This study is an extension of previous research that utilized different techniques to further enhance the model's predictability. Finally, they compare our results with the work of other authors.

It should be noted that the evaluation measures used in different studies vary, but since accuracy is the most commonly used, they were selected for inclusion in this table.

Table 2.1: Summary of the results in previous related studies

Author	**Model**	**Accuracy**	**Dataset**
Malekipirbazari and Aksakalli (2015)	1. Random Forest 2. K-Nearest Neighbor 3. Support Vector Machine 4. Logistic Regression	78% 70.1% 63.3% 54.50%	LendingClub (Jan 2012 – Sep 2014)
Akindaini (2017)	1. Logistic Regression 2. K-Neares Neighbor (K=5) 3. Multinomial Logistic Regression 4. Naïve Bayes	95.15% 74.08% 70.74% 83.14%	Mortgage loan data fro Fannie Mae
ElMasry (2019)	1. Random Forest 2. Support Vector Machine 3. K-Nearest Neighbor 4. Decision Tree	89.04% 89.04% 88.84% 88.04%	Freddie Mac (Jan 1999 – Mar 2017)

3. DATA AND METHODOLOGY

The study uses 4 algorithms including Logistic Regression, Decision Tree, Random Forest, XGBoost to predict and compare with each other..

3.1 Dataset

LendingClub is a peer-to-peer lending platform based in San Francisco, California, where investors and borrowers meet virtually. It was the first peer-to-peer lender to register its offerings as securities with the Securities and Exchange Commission (SEC) and to offer secondary loan trading. LendingClub is the largest peer-to-peer lending platform in the world.

The Lending Club processes the application with their own data science methods. However, on the side of the investor, there is nothing to ensure the creditworthiness of the borrower and the level of risk involved in any given case. Applying machine learning to loan default predictions, showcases a useful application of this branch of artificial intelligence to solve real-world and business problems. The publicly available data from Lending Club was used in this study. The data is available at LendingClub.com. The data pertains to the 2260700 loans that were funded by the platform between 2007 and the fourth quarter of 2018.

I excluded loans with statuses that are not yet final, such as "Current" and "Late (less than 30 days)". I consider " Charged Off " to be a positive label, and “Fully Paid" to be negative.

The success of classification learning is heavily dependent on the quality of the data providedfor training. In this chapter, the author will have an overview of the loan repayment data set andperform a data exploratory analysis in order to determine to preprocess the data and improve the prediction result. The data will also be split into training set (80%), and test set (20%). Training set will be used to fit the model, and test set will be to evaluate the best model toget an estimation of generalization error.

3.2 Feature Analysis

A total of 151 variables are present in the original Lending club dataset, with 112 variables being numerical and 39 being categorical. The data focuses on three different

aspects: personal details such as address, employment, and homeownership; credit history including bankruptcy filings, account balances, inquiries, and past due accounts; and loan characteristics such as issue date, FICO score, LC grade, subgrade, status, type of candidacy, policy, payment dates, purpose, term, late fees, and principal amount.

Table 2.1: Important feature and dataset description

Column	Description
credit_policy	1 if the customer meets the credit underwriting criteria of Lending-Club.com, and 0 otherwise
purpose	The purpose of the loan such as: credit card, debt consolidation, etc
int_rate	The interest rate of the loan (proportion)
installment	The monthly installments ($) o the authord by the borrower if the loan is funded
log_annual_inc	The natural log of the annual income of the borrower
dti	The debt-to-income ratio of the borrower
fico	The FICO credit score of the borrower
days_with_cr_line	The number of days the borrower has had a credit line
revol_bal	The borrower’s revolving balance
revol_util	The borrower’s revolving line utilization rate
inqlast6mths	The borrower’s number of inquiries by creditors in the last 6 months

delinq_2yrs	The number of times the borrower had been 30+ days past due on apayment in the past 2 years
pub_rec	The borrower's number of derogatory public records
status	Indicates whether the loan was not paid back in full (the borrower either defaulted or the borrower was deemed unlikely to pay it back)

By checking the data type and definition of each variable, the variables could be clas-sified into two groups: numerical and categorical.

3.3 Experiment Design

The experiment design is presented in the form of a flowchart, serving as an introduction to the methodology discussed in this chapter. The design begins with the raw dataset, as previously introduced, and proceeds to pre-processing. Next, the data is divided into two samples, followed by several steps to prepare for model training and evaluation. These steps are elaborated upon in subsequent sections.

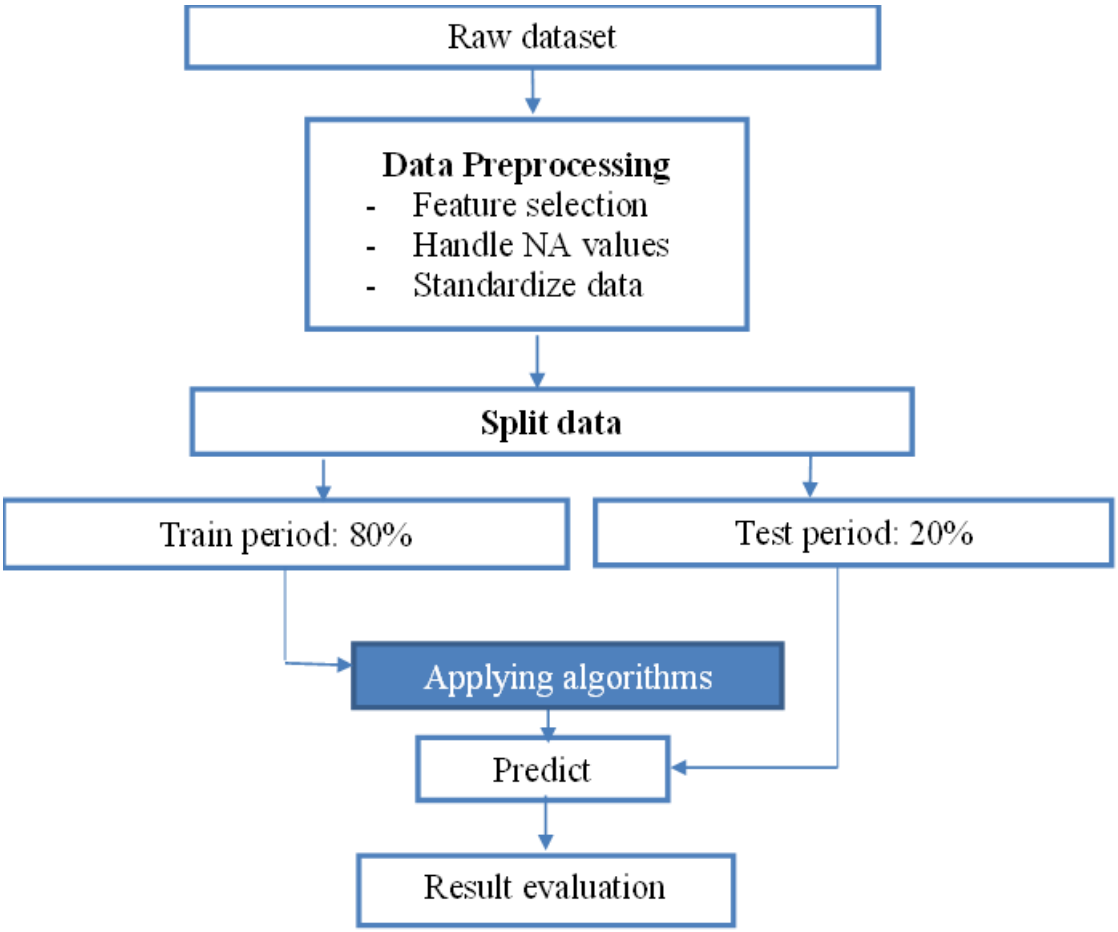

Figure 3.1: Ressearch model process

Source: Author

3.4 Exploratory Data Analysis

In order to enhance the visualization and comprehension of the dataset, a series of graphs were created and presented in figures 3.2 to figure 3.6.

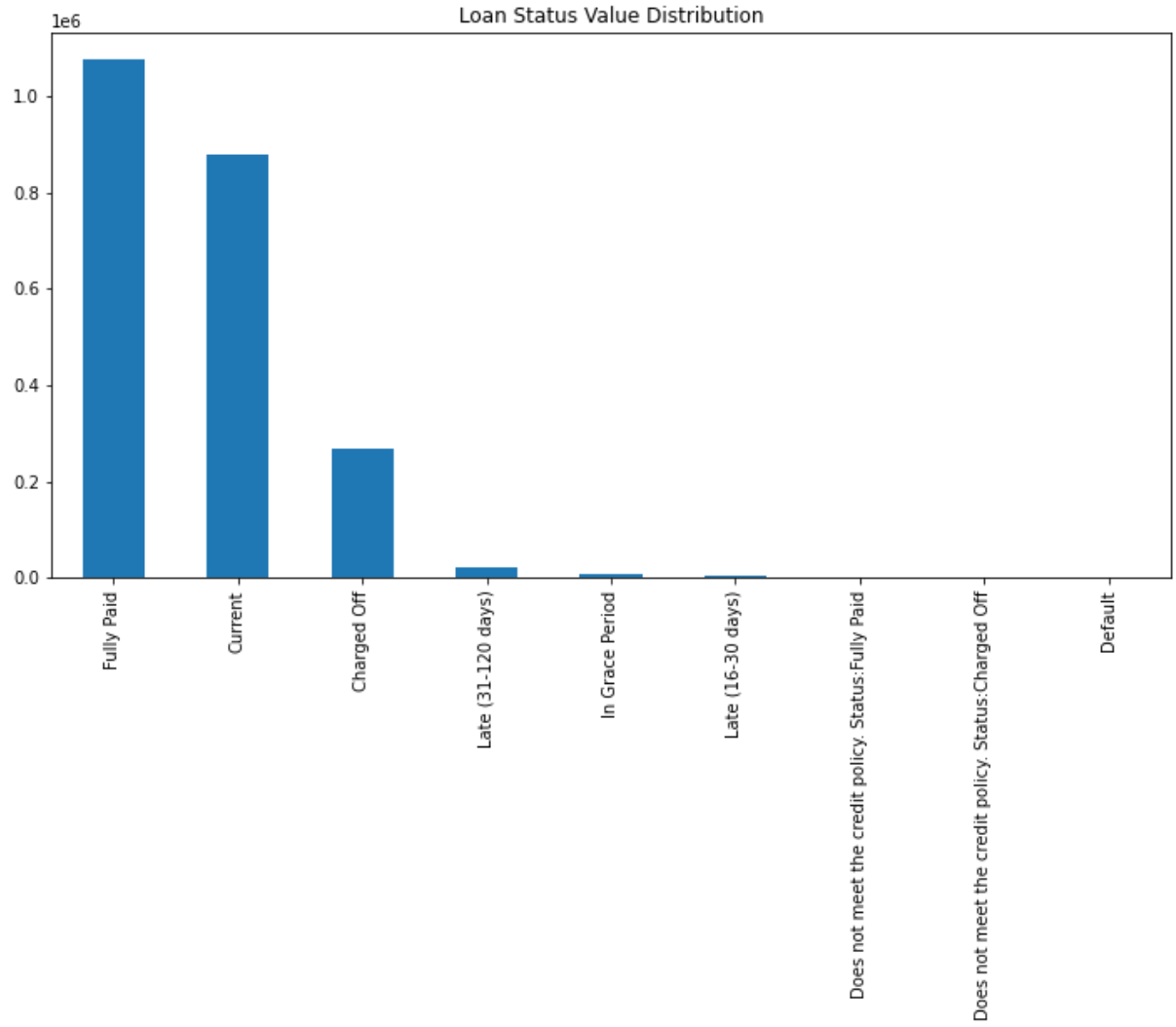

Figure 3.2: Loan status value distribution

Source: Author

Figure 3.2 illustrates the current loan status. However, the author focuses on the dependent variable, which includes the categories of "Fully Paid" and "Charged Off."

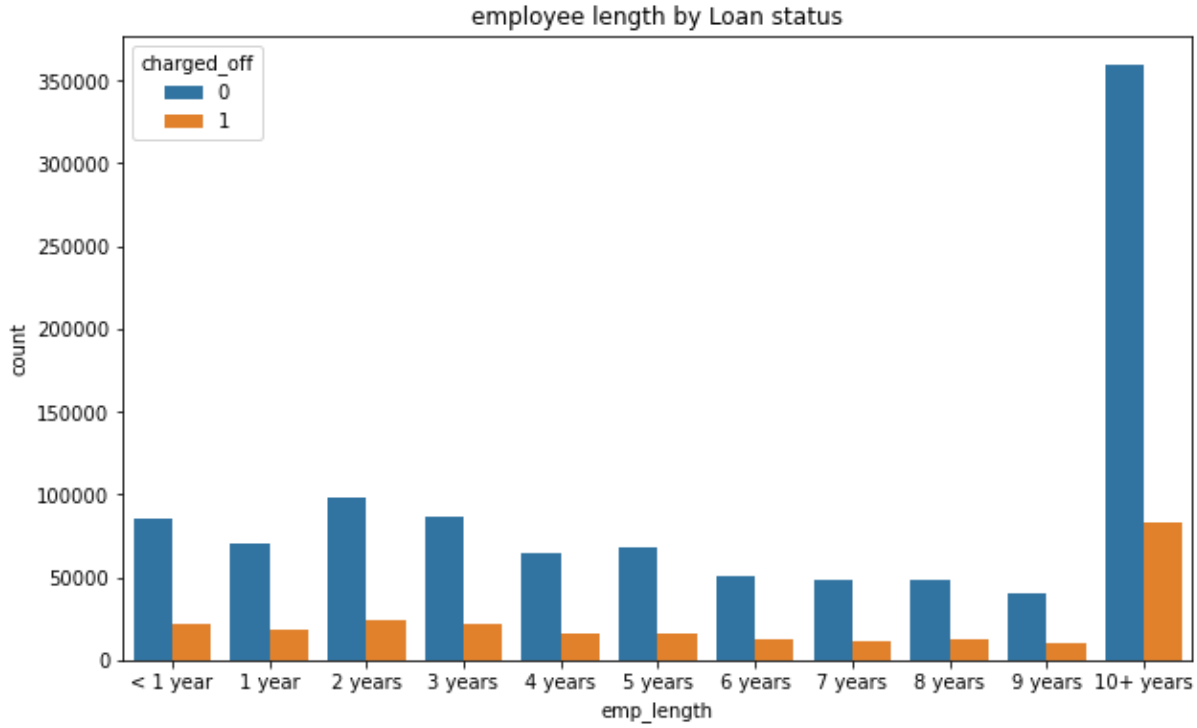

Figure 3.3: Employee length by loan status

Source: Author

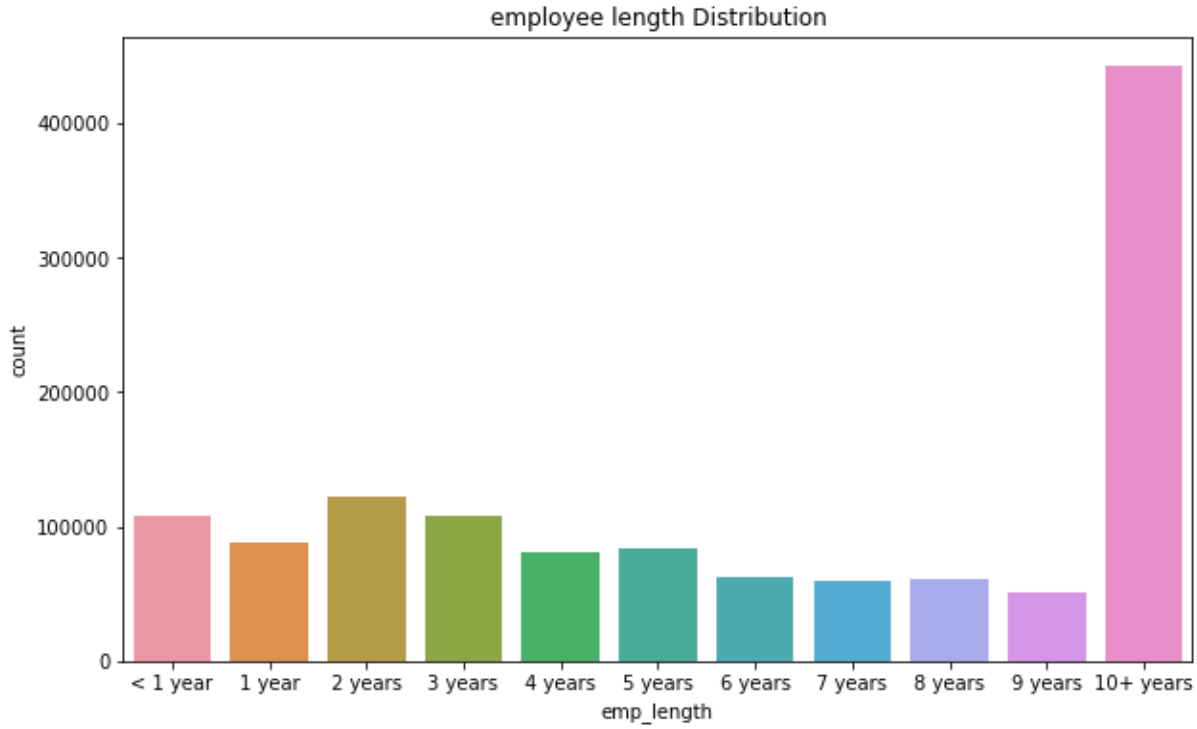

Figure 3.4: Employee length distribution

Source: Author

Figure 3.3 and figure 3.4 show that the dataset contains a significant number of loans with a term duration of over 10 years, totaling more than 400,000 cases.

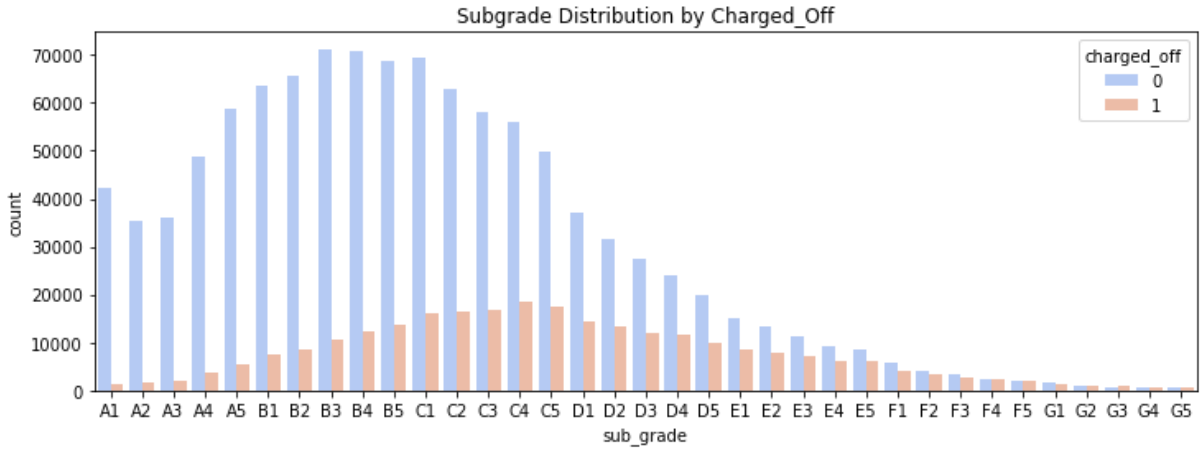

Figure 3.5: Subgrade distribution by charged off

Source: Author

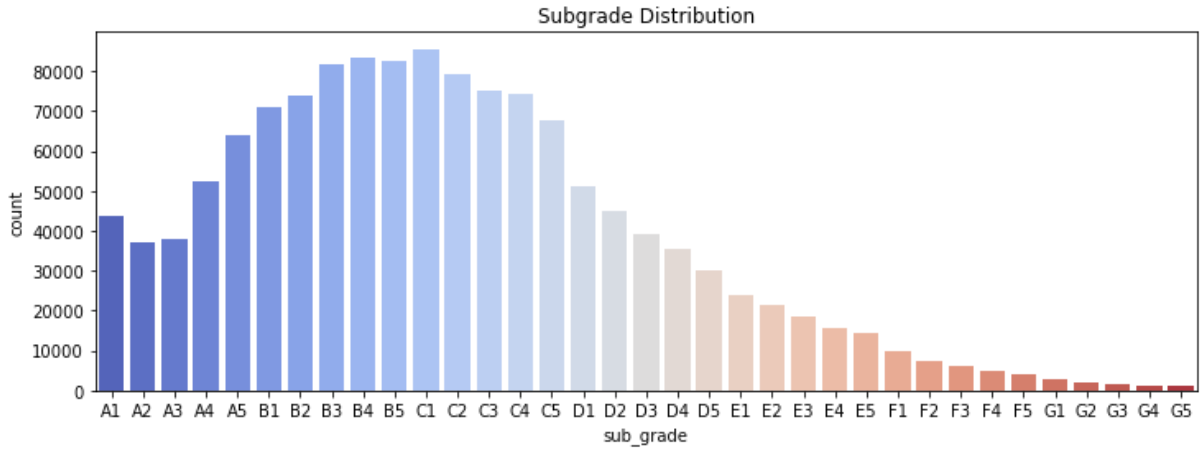

Figure 3.6: Subgrade distibution

Source: Author

One noteworthy observation from Figure 11 is the declining trend in the number of loans towards the bottom of the figure, indicating that there are comparatively fewer loans categorized as grades G to E, as opposed to grades A, B, C, and D.

3.5 Data Pre-Processing

In this research, various models are considered, and each model requires different types of preparation. For example, in Logistic Regression, it is crucial to remove correlated

inputs, as the estimation process can be more susceptible to failure if this is not considered. Further details about the pre-processing steps can be found in the next section.

3.5.1 Missing data

In the pre-processing stage, variables with more than seventy percent of missing values were removed. For variables with missing values below this threshold, various methods were employed to handle them. Lastly, the missing values were treated as a separate category if the percentage of missing values was significant, and imputation was not possible.

Use only “Fully Paid” and “Charged Off” in Loan Status Counts variables to avoid confusion and easily classify running models.

3.5.2 Feature Selection

The objective of feature selection is to prevent overfitting, decrease noise and redundancy, reduce computational effort, and facilitate interpretation of results. The various stages of this process included evaluating variable importance, correlation analysis, and identifying the optimal number of features through performance assessment on the test set. The criteria utilized for feature selection included:

- Excluding features that would not have been available at the time of the loan;
- Converting strings to numerical values;
- Eliminating redundant attributes;
- Removing predictors with zero (or almost zero) variance;
- Removing variables with high numbers of missing values, i.e., over 80%.

To evaluate each variable's significance in constructing the model, the Scikit-learn package was employed to identify redundant variables by creating a correlation matrix. This package can establish a matrix and rank the importance of each variable when constructing a model.

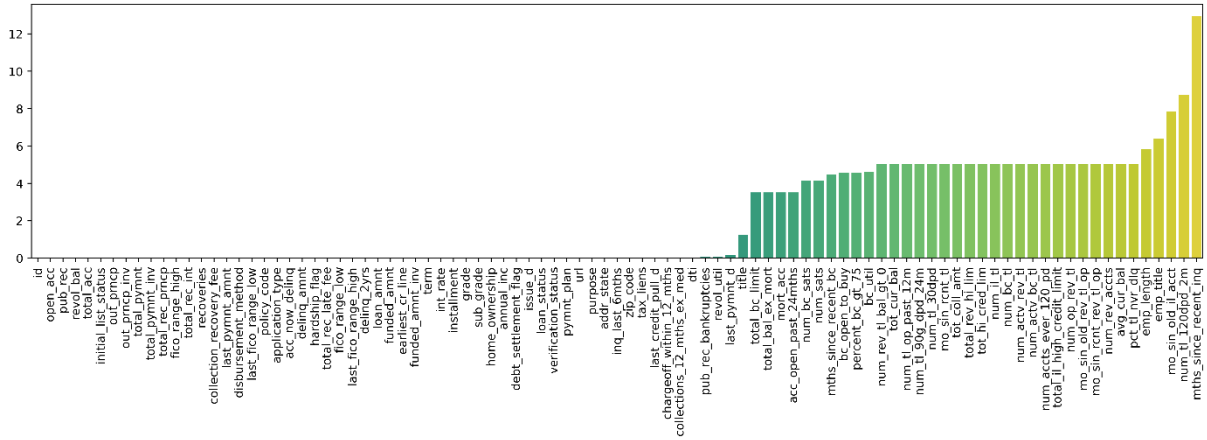

Figure 3.7: Variable importance

Source: Author

The selected variables for analysis are:

'loan_amnt','term','int_rate','installment','grade','sub_grade','emp_title','emp_length','home_ownership','annual_inc','verification_status','issue_d','purpose','title','dti','earliest_cr_line','inq_last_6mths','open_acc','pub_rec','revol_bal','revol_util',‘pub_rec_bankruptcies','addr_state','fico_range_low','fico_range_high','loan_status'

3.5.3 Correlation Matrix

To understand the relationship between multiple variables and attributes in the dataset. The first step is to check the correlation relationship between each variable, as Correlation is used as a basic quantity for many modelling techniques, as it can also help in predicting one attribute from another.

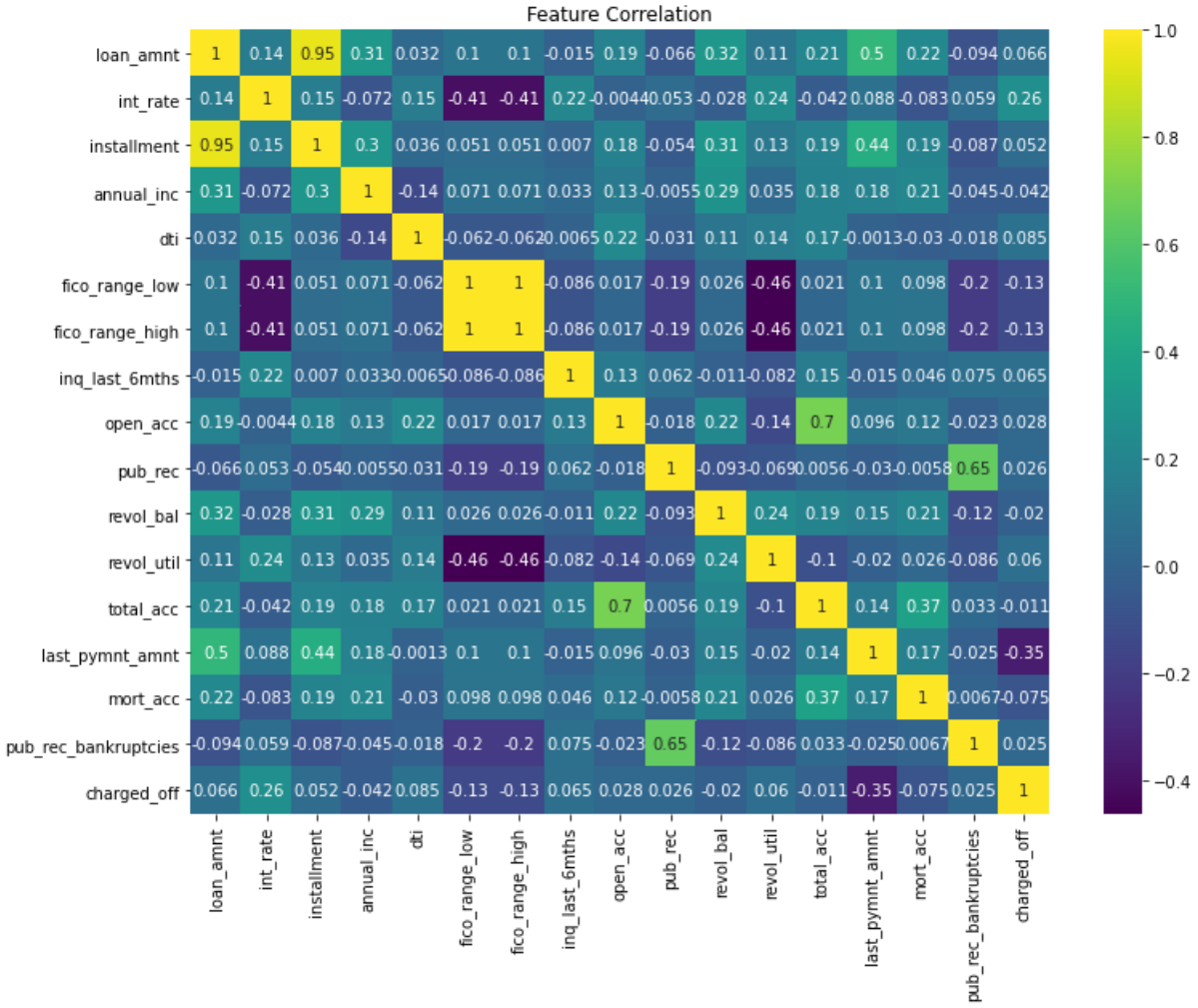

Figure 3.8: Correlation Matrix

Source: Author

From the correlation matrix to consider multicollinearity, the basic assumption of Logistic Regression can be used to use this model. Basic assumptions that must be met for Logistic Regression include independence of errors, linearity in the logit for continuous variables, absence of multicollinearity, and lack of strongly influential outliers.

Correlation between variables is not high. It can be said that the independent variables of the data do not have multicollinearity. Subtracting the correlation between "installment" the "loan_amnt" feature has a rather high correlation 0.95 compared to other variables. And “total_acc” has high correlation with “ open_acc” is 0.7. Also, the quite high correlation between “pub_rec” and “pub_rec_bankrupcies” is 0.65.

3.5.4 Categorical Feature Transformation

Feature transformation (feature engineering) encompasses the process of cleaning and formatting data, as well as evaluating the relationship of each feature to the target variable and applying necessary transformations. The pre-processing approach employed varied depending on the type and cardinality of the variable.

For numerical variables, such as "policy_code" and "pymnt_plan", which predominantly contained a single level of data across all rows, it was determined that removing them would be more beneficial.

Regarding categorical variables with a low number of categories, the date columns were initially in a categorical format and were subsequently converted into a date format to facilitate further analysis.

Categorical variables with a high number of categories, such as "id", "member_id", and "URL", were removed to mitigate overfitting, as these variables contained unique values for each loan solely for identification purposes. Additionally, the "emp_title" variable was also removed due to its nearly unique value for each loan, which could potentially impede learning time.

The variables "title" and "purpose" were found to have similar descriptions, and to reduce redundancy, it was decided to remove one of these features.

In terms of the default status variable, the initial target variable was "Loan status", which could take on values such as "Charged off", "current", "default", and "does not meet the credit policy". To focus on the primary objective of accurately distinguishing between good and bad loans, only loans that were fully paid or charged off, assuming they both met the credit policy, were included in the analysis. A fully paid loan corresponds to loans that have been completely repaid, either through prepayment or at the maturity of a three or five-year term. On the other hand, the charged-off category pertains to loans for which there is no longer a reasonable expectation of further payments. Subsequently, this variable was filtered to include only fully paid and charged-off loans, which represented 80% of the total observations, and transformed into a binary variable for modeling purposes.

It should be noted that the dataset is unbalanced due to the discrepancy between the number of default and non-default loans. Previous studies have highlighted the advantages of undersampling and oversampling techniques in addressing this issue. Further information on these techniques can be found in the "Class Reweight" section.

Lastly, the remaining factor variables, including "grade", "home_ownership", "verification_status", "purpose", "addr_state", and "application_type", were transformed into dummy variables, as Logistic Regression requires all data to be in numerical form (Nisbet et al., 2018).

3.6 Modeling Techniques

3.6.1 Sample Split

The selected approach involves partitioning the original dataset into two distinct parts, namely the training dataset and the testing dataset. Approximately 80% of the data is allocated to the training set, with the remaining proportion assigned to the testing set. This segregation serves to mitigate issues such as model overfitting, which can arise when the model is too closely tailored to the training data and may not generalize well to new, unseen data.

The testing section of the dataset is designed to ensure a sufficient amount of data for obtaining statistically meaningful results. By reserving a portion of the data for testing, the model's performance can be effectively evaluated on unseen data, providing a robust assessment of its predictive capabilities.

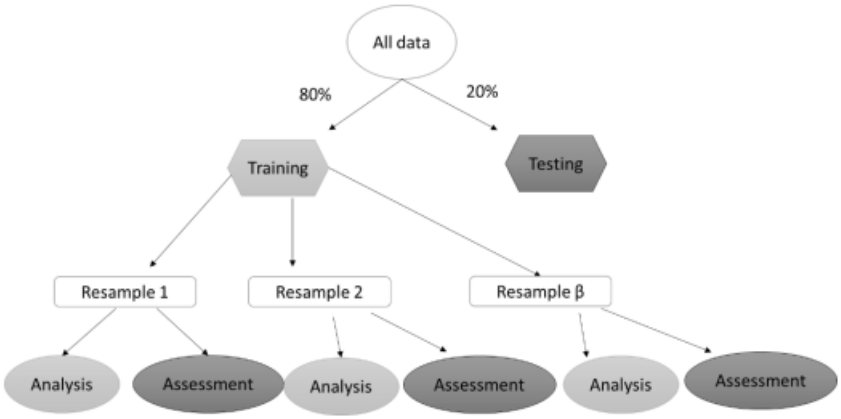

Figure 3.9: Predictive modelling workflow

Source: Author

3.6.2 Hyperparameters Optimisation

The hyperparameter tuning process was carried out using a cross-validated grid search, which involved selecting parameters from a predefined parameter grid in order to maximize the score of the underlying estimator (Pedregosa et al., 2011). The accuracy metric was used to evaluate the performance of each combination of hyperparameters.

In order to determine the appropriate range of values for each parameter when training the model, the author conducted an analysis of the Validation Curve separately. It is crucial to identify the optimal values for parameters when using them in combination. To achieve this, the GridSearchCV function was utilized, which includes inputs such as the model (Random Forest Classifier), parameters, range of parameter values, and train-test split method, among others. GridSearchCV evaluates all possible combinations of parameter values and retains the best combination. Subsequently, the author obtained the optimal parameter values and is now ready to run the Random Forest model with these optimized parameters.

Details of the hyperparameters, along with their respective values used in the grid search, were provided in Table 3.1, after the explanation of each classifier. The input hyperparameter values for the grid search are listed in the tables below.

Table 3.1: Hyperparameters optimisation

Hyperparameter	**Set of Values**
Logistic Regression	
alpha	1e-05
penalty	l1
Decision Tree	
criterion	entropy
max_depth	3
random_state	0
Random Forest	

max_depth	15
max_features	10
min_samples_split	8
n_estimators	100
XGBoost	
n_estimators	500
learning_rate	0,05

Source: Author

3.6.3 Buiding Pipeline

In the context of machine learning projects, pipelines become increasingly important as the project grows. They offer several benefits, especially when dealing with large datasets or resource-intensive tasks. The approaches I have discussed allow for easy scalability of infrastructure to accommodate such requirements. Moreover, pipelines enable repeatability through automation and provide an audit trail for machine learning processes, which is crucial for ensuring consistency and reproducibility in the results.

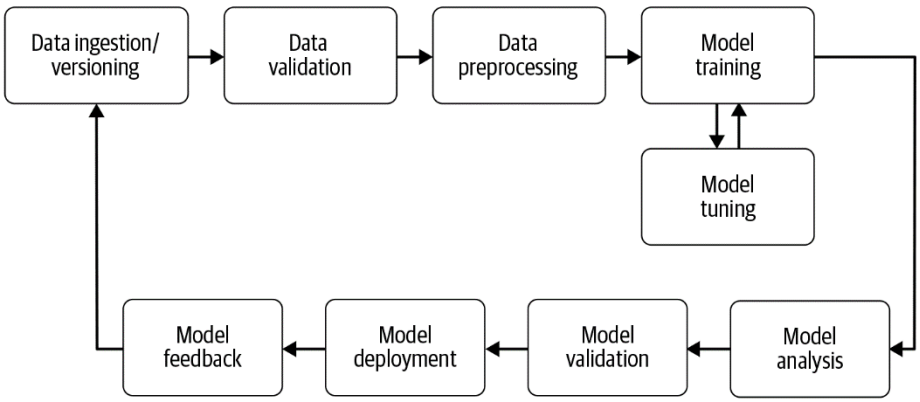

Figure 3.10: Building pipeline process

Source: Author

To combine numeric and categorical variables, several approaches can be employed. One approach is to use feature engineering techniques, where new features are created by combining existing numeric and categorical variables in meaningful ways. For example, numeric variables such as age or income can be combined with categorical

variables such as gender or education level to create new features that capture interactions or relationships between the variables.

Another approach is to use techniques such as one-hot encoding or label encoding to convert categorical variables into numeric representations that can be used in machine learning algorithms. One-hot encoding creates binary indicator variables for each category in a categorical variable, while label encoding assigns numeric labels to each category. These numeric representations can then be combined with the existing numeric variables to create a unified dataset for model training.

It is important to note that the choice of approach for combining numeric and categorical variables depends on the specific dataset and the machine learning algorithm being used. Care must be taken to ensure that the combined dataset is properly encoded, scaled, and normalized to prevent any bias or distortion in the modeling process.

3.6.4 Handling Imbalanced Dataset

In many real-world applications involving classification problems, the datasets tend to be imbalanced, meaning that the number of examples belonging to one class (referred to as the minority class) is significantly lower than another class (referred to as the majority class). To address this issue, in the present problem, I have chosen to utilize the technique of upsampling.

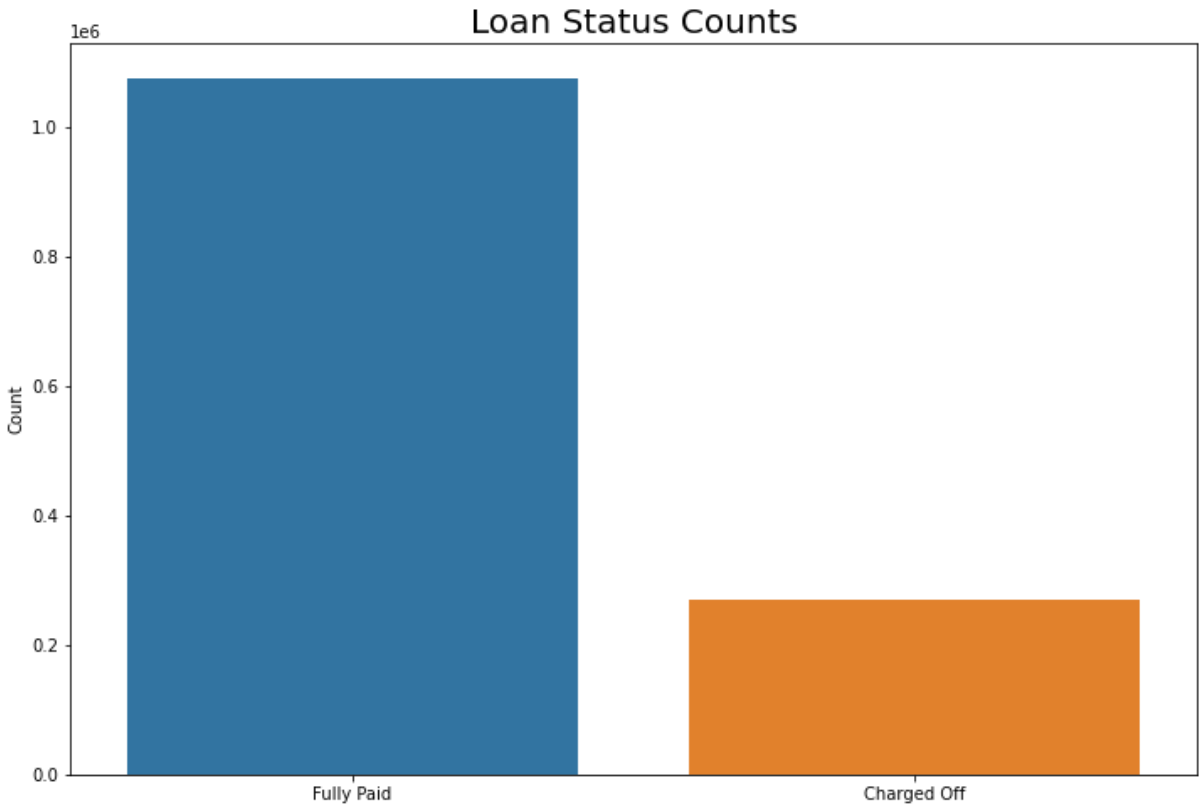

Figure 3.11: Loan status counts

Source: Author

Upsampling is a procedure in which synthetic data points corresponding to the minority class are generated and added to the dataset. This process helps to balance the counts of both class labels, bringing them to a similar level. By equalizing the class distribution, this procedure aims to prevent the model from being biased towards the majority class during the training process.

Through the use of upsampling, the imbalanced nature of the dataset can be mitigated, allowing the model to better capture patterns and relationships in the minority class, and improving its ability to accurately classify instances from both the majority and minority classes.

3.7 Applied Algorithms

The classifier described above is known as super-vised learning in machine learning. There is a predefined set of classes in supervised learning, and example objects are labeled with the appropriate class. The goal in this case is to determine whether the lender will be able to repay the loan. Several supervised models will be applied to the loan repayment

dataset in this chapter. Logistic Regression, Decision Tree, Random Forest, and XGBoost. Because the author are solving the classification problem with skewed data, the author will use more than just the accuracy score as the criterion, but also, I will consider the Precision and Recall as the performance index.

4. RESEARCH RESULTS

4.1 Detailed Results Of The Hyperparameter Optimization Process

The author present the accuracy results for each method, both before and after the tuning procedures (upsample). The findings reveal that the Logistic Regression algorithm experienced the most significant decrease in accuracy after tuning. Also, the smallest accuracy improvements were observed in the Logistic Regression method. Notably, the XGBoost algorithm exhibited the highest accuracy among all the methods both before and after, although it should be noted that this accuracy was calculated using the training set.

Table 4.1: The accuracy results before and after tuning

Algorithm	**Accuracy before tunning**	**Accuracy after tunning**	**%**
Logistic Regression	0.89	0.79	-12.65%
Decision Tree	0.86	0.84	-2.44%
Random Forest	0.80	0.82	-2.44%
XGBoost	0.91	0.89	-2.44%

Source: Author

However, despite the fact that accuracy did not improve, both recall and prediction for both classes were significantly improved due to the handling of the imbalanced dataset.

4.2 Detailed Results Of The Evaluation Measures

Table 4.2: The index results before and after tuning

Before tunning					
	%	**Logistic Regression**	**Decision Tree**	**Random Forest**	**XGBoost**
Precision	**Fully Paid (0)**	0.91	0.86	0.80	0.93
	Charged Off (1)	0.76	0.87	0.00	0.84
Recall	**Fully Paid (0)**	0.95	0.99	1.00	0.97

	Charged Off (1)	0.64	0.37	0.00	0.70
F1-score	**Fully Paid (0)**	0.93	0.92	0.89	0.95
	Charged Off (1)	0.69	0.51	0.00	0.76
Accuracy		0.89	0.84	0.80	0.91
After tunning					
	%	**Logistic Regression**	**Decision Tree**	**Random Forest**	**XGBoost**
Precision	**Fully Paid (0)**	0.72	0.88	0.83	0.94
	Charged Off (1)	0.93	0.81	0.81	0.86
Recall	**Fully Paid (0)**	0.95	0.80	0.80	0.85
	Charged Off (1)	0.64	0.89	0.83	0.94
F1-score	**Fully Paid (0)**	0.82	0.83	0.81	0.89
	Charged Off (1)	0.75	0.85	0.82	0.90
Accuracy		0.79	0.84	0.82	0.89

Source: Author

In summary, based on the results presented in Table 4.2, XGBoost is identified as the best performing classifier, followed by Decision Tree and Random Forest. On the other hand, Logistic Regression exhibit weaker performance. These findings align with previous studies such as Lessmann et al. (2015), Malekipirbazari and Aksakalli (2015), and Thanawala (2019), where Random Forest has been shown to deliver superior results, but in this case, Random Forest is ranked second. Similarly, ElMasry (2019) also classified Decision Tree as less accurate classifiers. However, this results solved that Logistic Regression had the worst results.

It is worth noting that most studies lack detailed presentation of results, often only reporting a single performance measure such as accuracy. It is suggested that studies should include other types of performance measures, similar to what has been presented in this research, or consider different approaches such as Amaro (2020). Exploring newer

and less institutionalized classifiers and learning approaches could also be interesting, as well as investigating the effects of different hyperparameter settings on performance measures.

Mostly, the problems using classification models, the accuracy index in prediction is often used as a standard measure to evaluate the effectiveness of the classification model. Due to the nature of the topic, in this study, the author not only considered the accuracy as a criterion to evaluate the model but also focused on improving the precision index in the confusion matrix. The reason comes from the point of view of "do not lose money" when lending, an ordinary lenders when making lending decision will avoid the highest possible risk and minimize the loss. In that direction, the model will be flexibly tested on many algorithms to find which algorithms suitable for prior data. Therefore, the result becomes more accurate and lenders can be more confident.

Compared with the previous studies cited in the study, the random forest algorithm in this study gives outstandingly high accuracy results.

Through experiments, it is noticed that the model was found which best fits the dataset with highest accuracy is the XGBoost Model. As the author expected, borrowers with higher annual income and higher FICO scores are morelikely to repay the loan fully. In addition, borrowers with lower interest rates and smaller installments are more likely to pay the loan fully.

Moreover, before tunning model, the recall for the "fully paid" class is high, while the recall for the "charged off" class is low, even, the Random Forest algorithm may still miss all cases of "charged off" class in credit default prediction.

In a credit default prediction problem, it may indicate some insights and evaluations about the performance of the classification model. With a high recall for the "fully paid" class, it means that the model correctly predicts almost all actual cases belonging to the "fully paid" class (true positives), and only misses a very small number of actual "fully paid" cases (false negatives).

However, with a low recall for the "charged off" class, it indicates that the model has the potential to miss a significant number of actual cases belonging to the "charged off"

class (false negatives) during the prediction process and fails to accurately predict these cases.

This may suggest that the model is performing well in accurately predicting customers who are likely to fully pay their debts (fully paid), but needs improvement in detecting customers who are likely to default on their debts (charged off), in order to detect P2P Lending credit fraud.

After tunning model, the recall had improvement for both classes. This indicates that the model is effectively detecting instances from both classes without favoring one class over the other, and is capable of identifying true positives from both classes while minimizing false negatives.

A fair recall between the two classes is desirable in many classification tasks to ensure balanced performance and avoid biased predictions. It indicates that the model is not overfitting to one class or neglecting the other, but rather providing reliable predictions for both classes.

Table 4.3 presents the pros and cons of each classifier, along with a summary of their performance. The "Pros" column highlights the advantages of each classifier with a "+" sign, while the "Cons" column outlines the downsides with a "-" sign.

Table 4.3: Performance analysis

Ranking	Algorithm	Performance	Pros&Cons
1	XGBoost	High	+ Fast training + Outliners have less impact to model - Complex interpretation
2	Decision Tree	High	+ Easy to explain + No need normalization

			- Easy to overfitting
3	Random Forest	High	+ No overfitting + Good performance in imbalanced dataset - Black box
4	Logistic Regression	Medium	+ Simple to implement - Need to normalization and check correlation

Source: Author

4.3 XGBoost

After training, the author reached the following result with the highest accuracy is 89% among other algorithms:

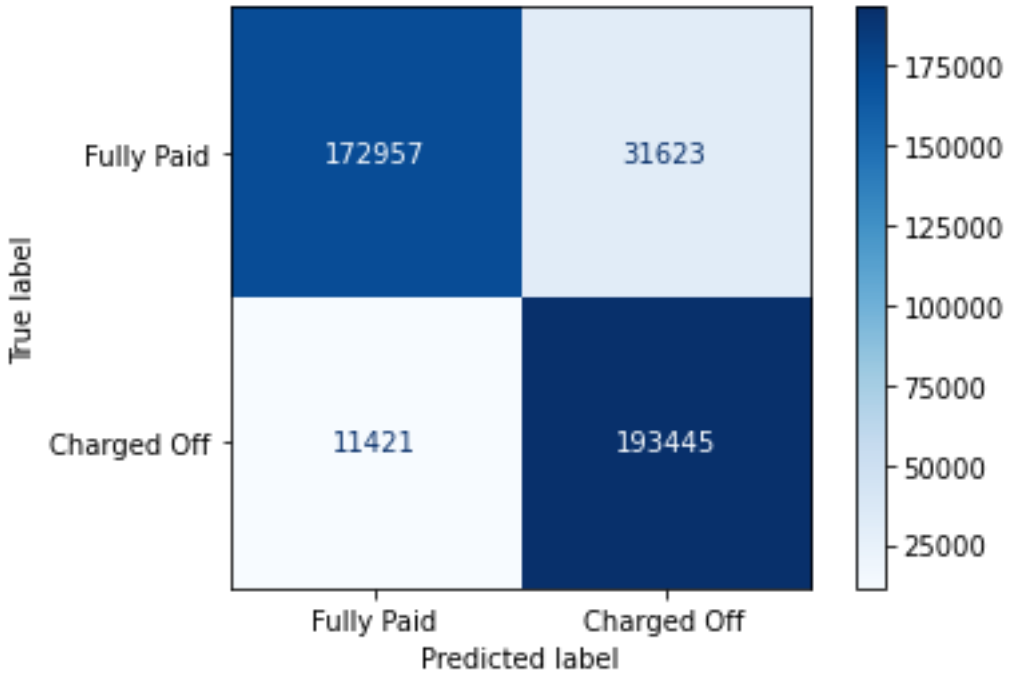

Figure 4.1: Confusion matrix of XGBoost

Source: Author

• The model correctly predicted 172957 cases of fully paid and actual fully paid

• Model misses 11421 cases actual charged off

• The model correctly predicted 193445 cases of charged off and actual charged off

• The model incorrectly predicted 31623 cases actual fully paid.

Table 4.4: Result of XGBoost

	Precision	Recall	F1-Score	Support
Fully paid (0)	0.94	0.85	0.89	204580
Charged off (1)	0.86	0.94	0.90	204866
Accuracy			0.89	409446
Macro Avg	0.90	0.89	0.89	409446
Weighted Avg	0.90	0.89	0.89	409446

Source: Author

4.4 Logistic Regression

Confusion matrix results are shown in Figure 4.2

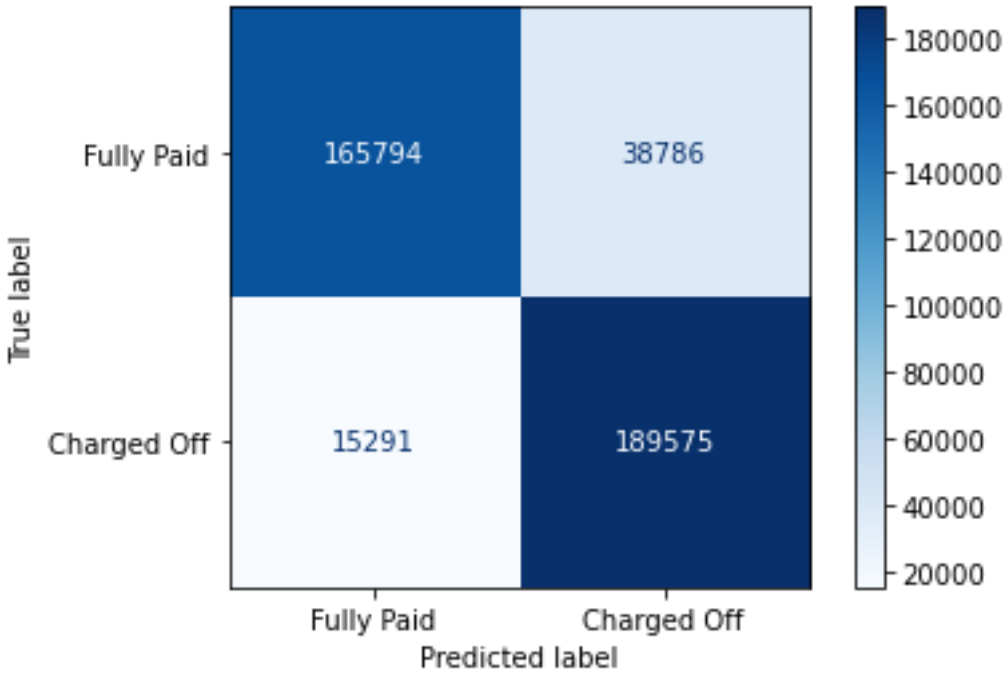

Figure 4.2: Confusion matrix of Logistics Regression

Source: Author

It can be seen that the result in test set, after running Logistic Regression with the above setting for a maximum of 1000 iterations, the author arrived at the following results: As the author can see, Logistic Regression is doing somewhat the author compared to naive models that blindly predict positive for all examples, or randomly guess positive and negative with 50% chance. Thanks to L1 regularization, the author did not observe overfitting issues. One thing that the author noticed and would like to improve upon is accuracy. The accuracy is quite low: 79%, it is hard to trust and use the model. Although the author used a balanced type the authorights to offset data imbalance, the prediction precision is only slightly better than randomly guessing. Therefore, the author suspects there may be non-linear relationships in the dataset that is not learned by Logistic Regression, which leads to our exploration with other algorithms. Specifically:

- The model correctly predicted 165798 cases of fully paid and actual fully paid
- Model misses 15291 cases actual charged off
- The model correctly predicted 189575 cases of charged off and actual charged off
- The model incorrectly predicted 38786 cases actual fully paid.

Table 4.5: Result of Logistics Regression

	Precision	Recall	F1-Score	Support
Fully paid (0)	0.72	0.95	0.82	204580
Charged off (1)	0.64	0.64	0.75	204866
Accuracy			0.79	409446
Macro Avg	0.83	0.79	0.79	409446
Weighted Avg	0.83	0.79	0.79	409446

Source: Author

In the confusion matrix figure 4.2, if you look at the accuracy index of only 79%, the index is quite in acceptable level. And the precision of variable 1 (charged off) is 72% and variable 0 (fully paid) is 64%, a quite rate but acceptable.

4.5 Decision Tree

The confusion matrix results are shown in Figure 4.3

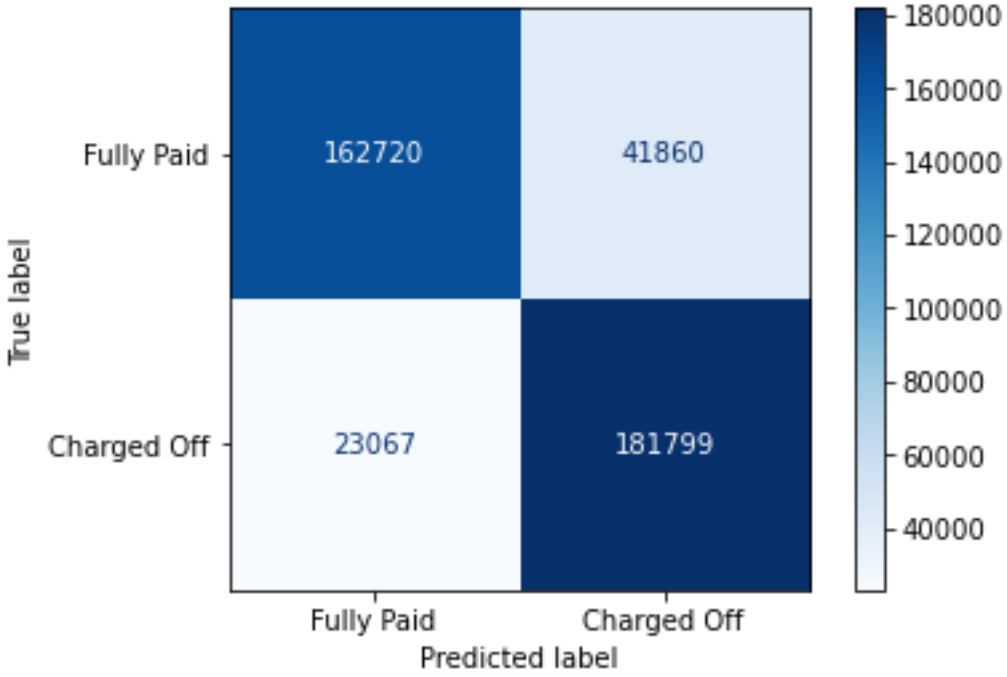

Figure 4.3: Confusion matrix of Decision Tree

Source: Author

• The model correctly predicted 162720 cases of fully paid and actual fully paid

• Model misses 23067 cases actual charged off

• The model correctly predicted 181799 cases of charged off and actual charged off

• The model incorrectly predicted 41860 cases actual fully paid.

Table 4.6: Result of Decision Tree

	Precision	Recall	F1-Score	Support
Fully paid (0)	0.88	0.80	0.83	204580
Charged off (1)	0.81	0.89	0.85	204866
Accuracy			0.84	409446
Macro Avg	0.84	0.84	0.84	409446
Weighted Avg	0.84	0.84	0.84	409446

Source: Author

Results that are more pleasing, with a high accuracy index of 84%. One advantage of a straightforward Decision Tree is that the model is simple to understand. The author able to quickly predict the outcome because I know which variable and which value the variable uses to split the data when I am building the Decision Tree. The models created by the Random Forest algorithm, on the other hand, are more intricate because they combine different Decision Trees.

4.6 Random Forest

The confusion matrix results are shown in Figure 4.4

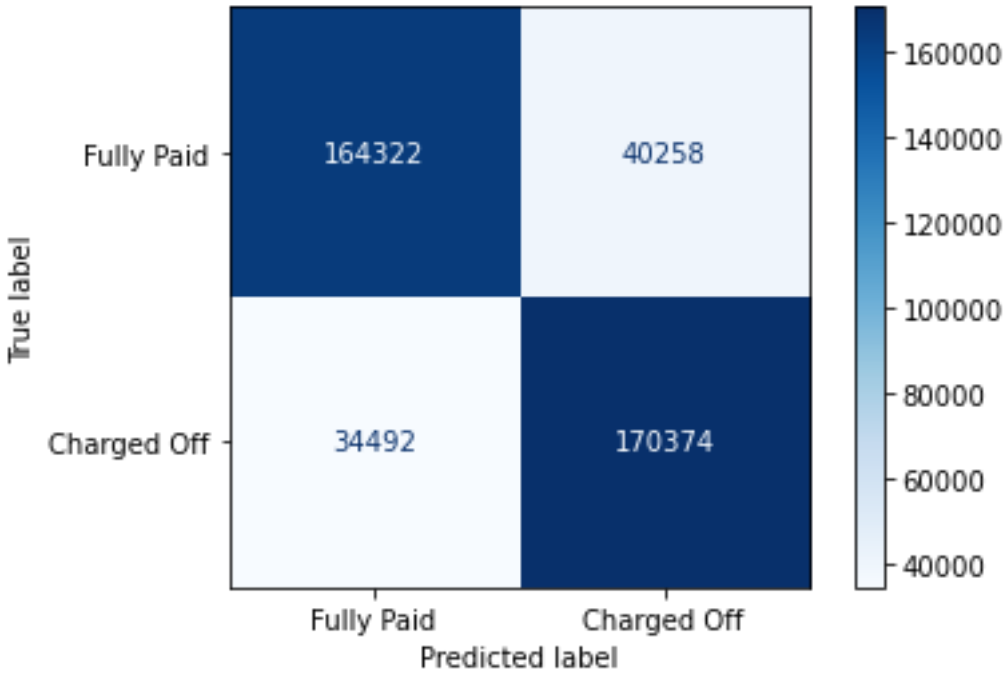

Figure 4.4: Confusion matrix of Random Forest

Source: Author

• The model correctly predicted 164322 cases of fully paid and actual fully paid

• Model misses 34492 cases actual charged off

• The model correctly predicted 170374 cases of charged off and actual charged off

• The model incorrectly predicted 40258 cases actual fully paid.

Table 4.7: Result of Random Forest

	Precision	Recall	F1-Score	Support
Fully paid (0)	0.83	0.80	0.81	204580
Charged off (1)	0.81	0.83	0.82	204866
Accuracy			0.82	409446
Macro Avg	0.82	0.82	0.82	409446
Weighted Avg	0.82	0.82	0.82	409446

Source: Author

Results that are more pleasing, with a high accuracy index of 82%.

5. CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions and Limitations

5.1.1 Conclusion

Nowadays, the loan business becomes more and popular, and many people apply for loans for various reasons. Ho the authorver, there are cases where people do not repay the bulk of the loan amount to the bank which results in huge financial loss. Hence, if there is a way that canefficiently classify the loaners in advance, it would greatly prevent the financial loss. In this study, the dataset was cleaned first, and the exploratory data analysis and featureengineering were performed. The strategies to deal with both missing values and imbalanced data sets were covered. Then the author propose fỏ machine learning models to predict if theapplicant could repay the loan, which are Random Forest, Logistic Regression, Decision Tree and XGBoost.

When tuning parameters, both RandomizedSearch Cross Validation and Grid Search Cross Validation methods are applied in different situations. Through experiments, it is noticed that the model was found which best fits the dataset with highest accuracy is the XGBoost Model: 89%.

As the author expected, borrowers with higher annual income and higher FICO scores are morelikely to repay the loan fully. In addition, borrowers with lower interest rates and smaller installments are more likely to pay the loan fully. And from this synthetic model, in general, the study obtained some conclusions as follows:

- First, it can be concluded that past loans have a significant evidence impact on the evolution of P2P lending, and recommend that lenders pay attention to the effect of past loans on the present.

- Secondly, it gives a accuracy of 89% and the recall for “fully paid: is 0.85 and “charged off” is0.94, which is an acceptable high rate.

- Thirdly, upsample technique immediately impact the improvement of the recall of the model.

- Finally, the predictive model can be seen as a tool for lenders to refer to with a variety of algorithms chosen which they can determine how to invest and profit for themselves.

5.1.2 Limitations

Although the research results have many positive results as well as quite other achievements, the objectives set out by the group from the beginning, however, nothing can be perfect, under the objective impacts, the topic still has some incomplete issues, the research still has the following limitations:

- The model results developed in the study compared to the results of previous studies are somewhat higher. However, the algorithm has not been carefully refined to find the highest accuracy index.

- The research scope of the study is still quite small, using secondary data, historical. The model needs to be developed to up to date.

In addition to these inherent limitations, there are other limitations that exist inside the model in particular and the research paper in general. Because there are many difficulties in implementing the topic, the author has not yet produced surveys. The group sees this not only as a limitation but also as an incentive to make further improvements to the topic to advance to the next research based on the available research.

5.2 Recommendations

In order to further advance the development of this study, the team has outlined several plans for future incubation as follows:

- It is acknowledged that there may be factors contributing to default that are not captured by the features in our current dataset. To address this, the author can consider incorporating external features, such as macroeconomic metrics that have historically

shown correlation with bond default rates. For categorical features like employment title, the author can explore merging them with signals such as average income by industry, similar to what Chang et al (2015) did with zip codes and average income by neighborhood. Additionally, the author can explore better utilization of existing features in the LendingClub dataset, such as the loan description provided by borrowers during the loan application process. Instead of discarding these freeform features, the author can investigate applying statistical natural language processing techniques, such as term frequency-inverse document frequency (TFIDF), as employed by Chang (2015).

- In addition to continuously improving model performance to enhance forecast results, the team also aspires to develop an application or web app that would serve as a platform for users to download, utilize, and view the developed model as a valuable reference for P2P Lending Default prediction. This would require further learning and implementation of interfaces and functionalities in the application, and transforming it into a complete product. This endeavor is seen as both a challenge and an opportunity for the author's personal development. Once completed, the application is expected to serve as a useful investment tool and contribute to the reputation of the University of Economics and Law as a research-oriented institution with practical applications and a wide scope of influence.

- Furthermore, the author aims to develop additional models to build more datasets and improve predictive performance, ensuring consistency across different types of predictions, including longer-term forecasts.

To pursue these directions, the first step will be to create a detailed plan with a specific schedule to minimize downtime and allocate ample time for resolving potential implementation difficulties in the upcoming tasks.

REFERENCES

Akindaini, B. (2017). Machine learning applications in mortgage default prediction (Master's thesis).

Anderson, R. (2007). The credit scoring toolkit: theory and practice for retail credit risk management and decision automation. Oxford University Press.

Bachmann, A., Becker, A., Buerckner, D., Hilker, M., Kock, F., Lehmann, M., ... & Funk, B. (2011). Online peer-to-peer lending-a literature review. Journal of Internet Banking and Commerce, 16(2), 1.

Brown, I., & Mues, C. (2012). An experimental comparison of classification algorithms for imbalanced credit scoring data sets. Expert Systems with Applications, 39(3), 3446-3453.

Cox, D. R. (1958). The regression analysis of binary sequences. Journal of the Royal Statistical Society: Series B (Methodological), 20(2), 215-232.

Chang, S., Kim, S. D., & Kondo, G. (2015). Predicting default risk of lending club loans. Machine Learning, 1-5.

Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).

Dhingra, C. (2020). A Visual Guide to Gradient Boosted Trees (XGBoost). Towards Data Science. Retrieved April, 4, 2022.

ElMasry, M. H. A. M. T. (2019). Machine learning approach for credit score analysis: a case study of predicting mortgage loan defaults (Doctoral dissertation).

Garson, G. D. (2012). Discriminant function analysis. Asheboro, NC: Statistical Associates Publishers.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning (Vol. 112, p. 18). New York: springer.

Kuhn, M., & Johnson, K. (2013). Applied predictive modeling (Vol. 26, p. 13). New York: Springer.

Kuncheva, L. I. (2014). Combining pattern classifiers: methods and algorithms. John Wiley & Sons.

Lee, T. S., & Chen, I. F. (2005). A two-stage hybrid credit scoring model using artificial neural networks and multivariate adaptive regression splines. Expert Systems with applications, 28(4), 743-752.

Lessmann, S., Baesens, B., Seow, H. V., & Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. European Journal of Operational Research, 247(1), 124-136.

Liu, X. Y., & Zhou, Z. H. (2013). Ensemble methods for class imbalance learning. Imbalanced learning: Foundations, algorithms, and applications, 61-82.

Malekipirbazari, M., & Aksakalli, V. (2015). Risk assessment in social lending via random forests. Expert Systems with Applications, 42(10), 4621-4631.

Pujun, B., Nick, C., & Max, L. (2016). Demystifying the workings of Lending Club. CS229 Stanford.

Saha, S. (2018). Understanding the log loss function of XGBoost. Medium.[Blog Post] Retrieved February, 21, 2021.

Tsai, K., Ramiah, S., & Singh, S. (2014). Peer lending risk predictor. CS229 Autumn.

Vijh, M., Chandola, D., Tikkiwal, V. A., & Kumar, A. (2020). Stock closing price prediction using machine learning techniques. Procedia computer science, 167, 599-606.

Xia, Y. (2019). A novel reject inference model using outlier detection and gradient boosting technique in peer-to-peer lending. IEEE Access, 7, 92893-92907.

Yiu, T. (2019). Understanding random forest-towards data science. Understanding Random Forest How the Algorithm Works and Why it Is So Effective.

APPENDIX

Appendix 1: Result of Logistics Regression

```
[[194400  10180]
 [ 74625 130241]]
              precision    recall  f1-score   support

           0       0.72      0.95      0.82    204580
           1       0.93      0.64      0.75    204866

    accuracy                           0.79    409446
   macro avg       0.83      0.79      0.79    409446
weighted avg       0.83      0.79      0.79    409446

Logistic Regression accuracy:  0.7928786702031525
```

Appendix 2: Result of Random Forest

```
[[164322  40258]
 [ 34492 170374]]
              precision    recall  f1-score   support

           0       0.83      0.80      0.81    204580
           1       0.81      0.83      0.82    204866

    accuracy                           0.82    409446
   macro avg       0.82      0.82      0.82    409446
weighted avg       0.82      0.82      0.82    409446

Random Forest accuracy:  0.8174362431187507
```

Appendix 3: Result of XGBoost

```
[[172957  31623]
 [ 11421 193445]]
              precision    recall  f1-score   support

           0       0.94      0.85      0.89    204580
           1       0.86      0.94      0.90    204866

    accuracy                           0.89    409446
   macro avg       0.90      0.89      0.89    409446
weighted avg       0.90      0.89      0.89    409446
```

Appendix 4: Result of Decision Tree

```
[[162720  41860]
 [ 23067 181799]]
              precision    recall  f1-score   support

           0       0.88      0.80      0.83    204580
           1       0.81      0.89      0.85    204866

    accuracy                           0.84    409446
   macro avg       0.84      0.84      0.84    409446
weighted avg       0.84      0.84      0.84    409446

Decision Tree accuracy:  0.8414271967487776
```

Appendix 4: Variables description

LoanStatNew	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
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.
desc	Loan description provided by the borrower
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.*

fico_range_high	The upper boundary range the borrower’s FICO at loan origination belongs to.
fico_range_low	The lower boundary range the borrower’s FICO at loan origination belongs to.
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
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.
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
member_id	A unique LC assigned Id for the borrower member.
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
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_il_12m	Number of installment accounts opened in past 12 months
open_il_24m	Number of installment accounts opened in past 24 months
open_act_il	Number of currently active installment trades
open_rv_12m	Number of revolving trades opened in past 12 months
open_rv_24m	Number of revolving trades opened in past 24 months
out_prncp	Remaining outstanding principal for total amount funded
out_prncp_inv	Remaining outstanding principal for portion of total amount funded by investors
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
recoveries	post charge off gross recovery
revol_bal	Total credit revolving balance
revol_util	Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit.
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
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
verified_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.
revol_bal_joint	Sum of revolving credit balance of the co-borrowers, net of duplicate balances
sec_app_fico_range_low	FICO range (high) for the secondary applicant
sec_app_fico_range_high	FICO range (low) for the secondary applicant
sec_app_earliest_cr_line	Earliest credit line at time of application 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_open_acc	Number of open 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
sec_app_open_act_il	Number of currently active installment trades 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_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_med	Number of collections within last 12 months excluding medical collections at time of application for the secondary applicant
sec_app_mths_since_last_major_derog	Months since most recent 90-day or worse rating at time of application for the secondary applicant
hardship_flag	Flags whether or not the borrower is on a hardship plan
hardship_type	Describes the hardship plan offering
hardship_reason	Describes the reason the hardship plan was offered
hardship_status	Describes if the hardship plan is active, pending, canceled, completed, or broken
deferral_term	Amount of months that the borrower is expected to pay less than the contractual monthly payment amount due to a hardship plan
hardship_amount	The interest payment that the borrower has committed to make each month while they are on a hardship plan
hardship_start_date	The start date of the hardship plan period
hardship_end_date	The end date of the hardship plan period

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.
hardship_length	The number of months the borrower will make smaller payments than normally obligated due to a hardship plan
hardship_dpd	Account days past due as of the hardship plan start date
hardship_loan_status	Loan Status as of the hardship plan start date
orig_projected_additional_accrued_interest	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.
hardship_payoff_balance_amount	The payoff balance amount as of the hardship plan start date
hardship_last_payment_amount	The last payment amount as of the hardship plan start date
disbursement_method	The method by which the borrower receives their loan. Possible values are: CASH, DIRECT_PAY
debt_settlement_flag	Flags whether or not the borrower, who has charged-off, is working with a debt-settlement company.
debt_settlement_flag_date	The most recent date that the Debt_Settlement_Flag has been set
settlement_status	The status of the borrower's settlement plan. Possible values are: COMPLETE, ACTIVE, BROKEN, CANCELLED, DENIED, DRAFT
settlement_date	The date that the borrower agrees to the settlement plan
settlement_amount	The loan amount that the borrower has agreed to settle for
settlement_percentage	The settlement amount as a percentage of the payoff balance amount on the loan
settlement_term	The number of months that the borrower will be on the settlement plan

Appendix 5: Source code

```
#!/usr/bin/env python
# coding: utf-8

# In[1]:


import re
import os

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

import warnings
warnings.filterwarnings("ignore")

# "magic" command to make plots show up in the notebook
get_ipython().run_line_magic('matplotlib', 'inline')
```

```
# In[2]:


# Import library
from sklearn.metrics import classification_report, confusion_matrix,
accuracy_score, plot_confusion_matrix


# # 1. Exploratory Data Analysis

# > Get an understanding for which variables are important, view summary
statistics, and visualize the data

# In[3]:


data = pd.read_csv('LendingClub_data.csv')


# In[4]:


data.info()


# In[5]:


# print(f"\033[1m\033[94mTotal null:\n{55 * '-'}")
# print(f"\033[30m{data.isnull().sum()}")
# data.head()


# In[6]:


data.describe()


# In[7]:


data[data.isna().any(axis=1)]


# In[8]:


data.loan_status.value_counts()


# > Current status of the loan
# 1: Charged Off
# 0: Fully Paid

# In[9]:


data['charged_off'] = data.loan_status.map({'Fully Paid': 0, 'Charged Off':
1})
```

```
data[['loan_status', 'charged_off']].head(10)


# In[10]:


notPaid = data[data["charged_off"] == 1].shape[0]
fullPaid = data[data["charged_off"] == 0].shape[0]

print(f"Fully Paid = {fullPaid}");
print(f"Not Fully Paid (Charged Off) = {notPaid}");
print(f"% Charged Off/Fully Paid = {(notPaid/fullPaid) * 100:.2f}%");

plt.figure(figsize=(12, 8));
sns.countplot(data["charged_off"]);
plt.xticks((1, 0), ["Charged Off", "Fully Paid"]);
plt.xlabel("");
plt.ylabel("Count");
plt.title("Loan Status Counts", y=1, fontdict={"fontsize": 20});


# # 1.1 Correlation Matrix

# ![The-scale-of-Pearsons-Correlation-Coefficient.png](attachment:The-scale-
of-Pearsons-Correlation-Coefficient.png)

# In[11]:


selected_features=['loan_amnt','term','int_rate','installment','grade','sub_g
rade','emp_title','emp_length','home_ownership',

'annual_inc','verification_status','issue_d','purpose','title','dti','earlies
t_cr_line','inq_last_6mths',

'open_acc','pub_rec','revol_bal','revol_util','total_acc','initial_list_statu
s','application_type','mort_acc',
                    'last_pymnt_amnt',

'pub_rec_bankruptcies','addr_state','fico_range_low','fico_range_high','loan_
status']


# In[12]:


len(selected_features)


# In[13]:


df = pd.read_csv('LendingClub_data.csv',usecols=selected_features,
low_memory=False)
df.head()


# In[14]:


df['charged_off'] = df.loan_status.map({'Fully Paid': 0, 'Charged Off': 1})
df[['loan_status', 'charged_off']].head(10)
```

```
# In[15]:


notPaid = df[df["charged_off"] == 1].shape[0]
fullPaid = df[df["charged_off"] == 0].shape[0]

print(f"Fully Paid = {fullPaid}");
print(f"Not Fully Paid (Charged Off) = {notPaid}");
print(f"% Charged Off/Fully Paid = {(notPaid/fullPaid) * 100:.2f}%");

plt.figure(figsize=(12, 8));
sns.countplot(df["charged_off"]);
plt.xticks((1, 0), ["Charged Off", "Fully Paid"]);
plt.xlabel("");
plt.ylabel("Count");
plt.title("Loan Status Counts", y=1, fontdict={"fontsize": 20});


# In[16]:


plt.figure(figsize=(12,9))
df_corr = df.corr()
sns.heatmap(df_corr, annot=True, cmap='viridis')
plt.title('Feature Correlation')


# In[17]:


sns.scatterplot(data = df, x = 'installment', y = 'loan_amnt')
plt.title('installment / loan_amnt corr')


# In[18]:


df.drop(['installment',
         'total_acc',
         'pub_rec'],
         axis=1,
        inplace=True)


# In[19]:


df.shape


# In[20]:


sns.boxplot(data = df, x = 'charged_off', y = 'loan_amnt')
plt.title('Boxplot Charged Off / Loan Amount')


# In[21]:
```

```
df.grade.unique()


# In[22]:


plt.figure(figsize=(12,4))
subgrade_order = sorted(df['sub_grade'].unique())
sns.countplot(x='sub_grade',data=df,order =
subgrade_order,palette='coolwarm')
plt.title('Subgrade Distribution');


# In[23]:


plt.figure(figsize=(12,4))
subgrade_order = sorted(df['sub_grade'].unique())
sns.countplot(x='sub_grade',data=df,order =
subgrade_order,palette='coolwarm', hue='charged_off' )
plt.title('Subgrade Distribution by Charged_Off');


# In[24]:


df.drop('loan_status', axis=1, inplace=True)


# In[25]:


df.isnull().sum()/len(df)*100


# In[26]:


df.emp_title.value_counts()


# In[27]:


df.emp_title.value_counts().count()


# In[28]:


df.drop('emp_title', axis=1, inplace=True)


# In[29]:


sorted(df.emp_length.dropna().unique())


# In[30]:
```

```
length_order = ['< 1 year', '1 year',
 '2 years',
 '3 years',
 '4 years',
 '5 years',
 '6 years',
 '7 years',
 '8 years',
 '9 years',
 '10+ years']


# In[31]:


plt.figure(figsize=(10,6))
sns.countplot(data = df, x = 'emp_length', order=length_order)
plt.title('employee length Distribution')


# In[32]:


plt.figure(figsize=(10,6))
sns.countplot(data = df, x = 'emp_length', hue='charged_off',
order=length_order)
plt.title('employee length by Loan status')


# In[33]:


temp1 = df[df.charged_off == 0].groupby('emp_length').charged_off.count()
temp2 = df[df.charged_off == 1].groupby('emp_length').charged_off.count()


# In[34]:


(temp1/temp2).plot(kind='bar')
plt.title('Emp.Length Fully Paid/Charged Off Ratio')


# In[35]:


df.drop('emp_length', axis=1, inplace=True)


# In[36]:


round(df.isnull().sum()/len(df),2)


# In[37]:


df.dropna(inplace=True)


# In[38]:
```

```
print("Missing values: ",df.isnull().sum().sum())

# In[39]:

df.columns

# In[40]:

df.drop('title', axis=1, inplace=True)
df.drop('addr_state', axis=1, inplace=True)

# In[41]:

df['fico_range'] = (df.fico_range_high+df.fico_range_low)/2
df.drop(['fico_range_low','fico_range_high'], axis=1, inplace=True)

# In[42]:

df.select_dtypes(exclude=["category", "object"])

# # Feature Scaling

# In[43]:

df.select_dtypes(exclude=["category", "object"])

# In[44]:

target = df.charged_off.astype('int')
features = df.drop('charged_off', axis=1)
features.head(10)

# In[45]:

numeric_features = features.select_dtypes(exclude=['object', 'category'])
numeric_features.head()

# In[46]:

numeric_features.columns

# In[47]:
```

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
scaled = sc.fit_transform(numeric_features)
print(scaled)


# In[48]:


df.dtypes


# In[49]:


scaled_features = pd.DataFrame(scaled, columns=['loan_amnt', 'int_rate',
'annual_inc', 'dti', 'inq_last_6mths',
       'open_acc', 'revol_bal', 'revol_util', 'last_pymnt_amnt', 'mort_acc',
       'pub_rec_bankruptcies', 'fico_range'], index=df.index)
scaled_features.head()


# In[50]:


scaled_features.isnull().sum()


# In[51]:


scaled_features.shape


# In[52]:


object_features = df.select_dtypes(['object', 'category'])
one_hot_features=pd.get_dummies(object_features, drop_first=True)
one_hot_features.head()


# In[53]:


one_hot_features.shape


# In[54]:


one_hot_features.isnull().sum().sum()


# In[55]:


model_df = scaled_features.merge(one_hot_features, on=scaled_features.index)
model_df.head()
```

```
# In[56]:


model_features = df.drop('charged_off', axis=1)
model_target = df.charged_off


# In[57]:


from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(model_features,
model_target, test_size=0.2, random_state=42)
print("X_train shape: Obs- {} / col- {}".format(X_train.shape[0],
X_train.shape[1]))
print("X_test shape: Obs- {} / col- {}".format(X_test.shape[0],
X_test.shape[1]))
print("y_train shape: Obs- {} / col- {}".format(y_train.shape[0], 0))
print("y_test shape: Obs- {} / col- {}".format(y_test.shape[0], 0))


# In[58]:


model_numeric_features = X_train.select_dtypes(exclude=['object',
'category']).columns
model_categorical_features = X_train.select_dtypes(['object',
'category']).columns


# In[59]:


from sklearn.pipeline import Pipeline
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
from sklearn.linear_model import LogisticRegression

numeric_transformer = Pipeline(
    steps=[
        ('scaler', StandardScaler())
    ]
)

categorical_transformer = Pipeline(
    steps=[
        ('one_hot', OneHotEncoder(handle_unknown='ignore'))
    ]
)

preprocessor = ColumnTransformer(
    transformers=[
        ('numeric', numeric_transformer, model_numeric_features),
        ('categorical', categorical_transformer, model_categorical_features)
    ]
)

clf = Pipeline(steps=[('preprocessor', preprocessor),
                      ('classifier',
LogisticRegression(solver='liblinear'))])
```

```
clf.fit(X_train, y_train)
clf_preds = clf.predict(X_test)


# In[60]:


from sklearn.metrics import classification_report
print(confusion_matrix(y_test,clf_preds))
print(classification_report(y_test,clf_preds))
print('Logistic Regression accuracy: ', accuracy_score(y_test, clf_preds))
plot_confusion_matrix(clf, X_test,y_test,cmap= 'Blues',values_format="d",
display_labels = ['Fully Paid', 'Charged Off'])


# In[61]:


from sklearn.tree import DecisionTreeClassifier #import library

dt = Pipeline(steps=[('preprocessor', preprocessor),
                     ('classifier',
DecisionTreeClassifier(criterion='entropy', max_depth=3, random_state=0))])
dt.fit(X_train, y_train)
dt_preds = dt.predict(X_test)


# In[62]:


print(confusion_matrix(y_test,dt_preds))
print(classification_report(y_test,dt_preds))
print('Decision Tree accuracy: ', accuracy_score(y_test, dt_preds))
plot_confusion_matrix(dt, X_test,y_test,cmap= 'Blues',values_format="d",
display_labels = ['Fully Paid', 'Charged Off'])


# In[63]:


from sklearn.ensemble import RandomForestClassifier

clf_random = Pipeline(steps=[('preprocessor', preprocessor),
                     ('classifier', RandomForestClassifier(max_depth = 15,
                                max_features = 10,
                                min_samples_split = 8,
                                n_estimators = 100))])
clf_random.fit(X_train, y_train)
random_preds = clf_random.predict(X_test)


# In[64]:


print(confusion_matrix(y_test,random_preds))
print(classification_report(y_test,random_preds))
print('Random Forest accuracy: ', accuracy_score(y_test, random_preds))
plot_confusion_matrix(clf_random, X_test,y_test,cmap=
'Blues',values_format="d", display_labels = ['Fully Paid', 'Charged Off'])


# In[65]:
```

```
import xgboost as xgb

clf_xgb = Pipeline(steps=[('preprocessor', preprocessor),
                          ('classifier', xgb.XGBClassifier())])
clf_xgb.fit(X_train, y_train)
xgb_preds = clf_xgb.predict(X_test)


# In[66]:


print(confusion_matrix(y_test,xgb_preds))
print(classification_report(y_test,xgb_preds))
plot_confusion_matrix(clf_xgb, X_test,y_test,cmap= 'Blues',values_format="d",
display_labels = ['Fully Paid', 'Charged Off'])


# In[67]:


from sklearn.utils import resample

df_majority = df[df['charged_off']==0] #Fully Paid
df_minority = df[df['charged_off']==1] #Charged Off


# In[68]:


df_minority_upsampled = resample(df_minority,
                                 replace=True,     # sample with replacement
                                 n_samples=len(df_majority),    # to match
majority class
                                 random_state=1234) # reproducible results

# Combine majority class with upsampled minority class
df_upsampled = pd.concat([df_majority, df_minority_upsampled])

# Display new class counts
df_upsampled.value_counts('charged_off')


# In[69]:


features = df_upsampled.drop('charged_off', axis=1)
target = df_upsampled.charged_off


# In[70]:


# Splitting the dataset into the Training set and Test set
X_trains, X_tests, y_trains, y_tests = train_test_split(features, target,
test_size=0.2, random_state=42)
print("X_train shape: Obs- {} / col- {}".format(X_trains.shape[0],
X_trains.shape[1]))
print("X_test shape: Obs- {} / col- {}".format(X_tests.shape[0],
X_tests.shape[1]))
print("y_train shape: Obs- {} / col- {}".format(y_trains.shape[0], 0))
```

```
print("y_test shape: Obs- {} / col- {}".format(y_tests.shape[0], 0))


# In[71]:


model_numeric_feature = X_trains.select_dtypes(exclude=['object',
'category']).columns
model_categorical_feature = X_trains.select_dtypes(['object',
'category']).columns


# In[72]:


numeric_transformer = Pipeline(
    steps=[
        ('scaler', StandardScaler())
    ]
)

categorical_transformer = Pipeline(
    steps=[
        ('one_hot', OneHotEncoder(handle_unknown='ignore'))
    ]
)

preprocessor = ColumnTransformer(
    transformers=[
        ('numeric', numeric_transformer, model_numeric_feature),
        ('categorical', categorical_transformer, model_categorical_feature)
    ]
)

clf_lgt = Pipeline(steps=[('preprocessor', preprocessor),
                      ('classifier',
LogisticRegression(solver='liblinear'))])

clf_lgt.fit(X_trains, y_trains)
clf_pred = clf.predict(X_tests)


# In[73]:


print(confusion_matrix(y_tests,clf_pred))
print(classification_report(y_tests,clf_pred))
print('Logistic Regression accuracy: ', accuracy_score(y_tests, clf_pred))
plot_confusion_matrix(clf_lgt, X_tests,y_tests,cmap=
'Blues',values_format="d", display_labels = ['Fully Paid', 'Charged Off'])


# In[80]:


clf_dt = Pipeline(steps=[('preprocessor', preprocessor),
                      ('classifier',
DecisionTreeClassifier(criterion='entropy', max_depth=3, random_state=0))])
clf_dt.fit(X_trains, y_trains)
dt_pred = clf_dt.predict(X_tests)
```

```
# In[81]:


print(confusion_matrix(y_tests,dt_pred))
print(classification_report(y_tests,dt_pred))
print('Decision Tree accuracy: ', accuracy_score(y_tests, dt_pred))
plot_confusion_matrix(clf_dt, X_tests,y_tests,cmap=
'Blues',values_format="d", display_labels = ['Fully Paid', 'Charged Off'])


# In[74]:


clf_rf = Pipeline(steps=[('preprocessor', preprocessor),
                         ('classifier', RandomForestClassifier(max_depth = 15,
                                        max_features = 10,
                                        min_samples_split = 8,
                                        n_estimators = 100))])
clf_rf.fit(X_trains, y_trains)
rf_preds = clf_rf.predict(X_tests)


# In[75]:


print(confusion_matrix(y_tests,rf_preds))
print(classification_report(y_tests,rf_preds))
print('Random Forest accuracy: ', accuracy_score(y_tests, rf_preds))
plot_confusion_matrix(clf_rf, X_tests,y_tests,cmap=
'Blues',values_format="d", display_labels = ['Fully Paid', 'Charged Off'])


# In[76]:


clf_xgboost = Pipeline(steps=[('preprocessor', preprocessor),
                         ('classifier', xgb.XGBClassifier())])
clf_xgboost.fit(X_trains, y_trains)
xgboost_preds = clf_xgboost.predict(X_tests)


# In[77]:


print(confusion_matrix(y_tests,xgboost_preds))
print(classification_report(y_tests,xgboost_preds))
plot_confusion_matrix(clf_xgboost, X_tests,y_tests,cmap=
'Blues',values_format="d", display_labels = ['Fully Paid', 'Charged Off'])
```