

Predicting Defaulter in Loan Approval Using Machine Learning

S. SHIYAM¹, DR. JIBRAEL JOS²

¹ Master of Science (Data Science), CHRIST (Deemed to be University), Pune Lavasa Campus, India

² Project Guide, CHRIST (Deemed to be University), Pune Lavasa Campus, India

Abstract- In wide range of improving banking sector in recent times and the increasing trend of taking loans is one of a large population applies for bank loans. But the major problem all banking sectors face in this ever-changing economy is the increasing rate of loan defaults, and the banking authorities are finding it more difficult to correctly assess loan requests and tackle the risks of people defaulting on loans. The analysis of risks and assessment of default becomes crucial thereafter. Banks hold huge volumes of customer behavior related data from which they are unable to arrive at a judgement if an applicant can be defaulter or not. In light of the given problems, this paper proposes two machine learning models to predict whether an individual should be given a loan by assessing certain attributes and therefore help the banking authorities by easing their process of selecting suitable people from a given list of candidates who applied for a loan. This paper does a comprehensive and comparative analysis between two algorithms (i) Decision Trees (ii) Ensemble Boosting

I. INTRODUCTION

1. Project Description

Most banks' main line of business is loan distribution. The main source of a bank's assets is the profit made from loans distributed by the bank. In the financial world, the fundamental purpose is to put your money in secure hands. The banking industry provides the majority of a country's revenue. The loan is sanctioned based on the customer's criteria. Every bank will support both open-ended and closed-ended loans. Despite the fact that giving loans is beneficial to both parties, the activity is fraught with dangers. Credit Risk refers to a borrower's inability to repay a loan within the agreed-upon time frame, which was set by both the lender and the borrower. Machine learning is the most prominent new technology trend that is transforming every recognized domain and technical innovation in a variety of sectors and businesses. It has the potential to bring unique advantages in banks.

The Loan Prediction System can determine the weight of each characteristic involved in loan processing automatically, and the same features are processed according to their associated weight on new test data. The applicant can be given a deadline to determine whether or not his or her loan will be approved. The Loan Prediction System allows you to jump to a specific application and review it on a priority basis. This paper is intended just for the management of a bank or finance firm; the entire prediction process is conducted in private, and no stakeholders will be able to influence the outcome. The results for a specific Loan Id can be sent to other bank departments, allowing them to take appropriate action on the application. This facilitates the completion of other formalities by all other departments.

When applying for a loan, you must first complete paperwork, which results in a large amount of data. Banking has improved over the previous few years. Before issuing a loan, it is critical to assess the clients' credit eligibility. To evaluate a borrower in the traditional loan procedure, banking authorities primarily use the '5C concept,' i.e., Character, Capital, Capacity, Collateral, and Conditions [2]. This assessment was based primarily on personal experience and expertise of customer service. This approach has a lot of drawbacks. Although banks and large financial businesses grant loan requests after a regressive screening and validation process, there is no guarantee that the applicant chosen at the end of the process will be able to repay the loan on time. Previously, banks would recruit highly qualified individuals whose primary function was to examine applicants and determine whether or not they were suitable for a loan following a thorough examination. To establish a candidate's worthiness for loan acceptance or rejection, a numerical score called a "Credit Score" was employed. In general, a high credit score helps authorities calculate the likelihood of a

borrower repaying a loan on time based on their credit history or payment history, as well as their background [3].

Credit scoring needed the use of experts and statistical algorithms to accurately predict an applicant's creditworthiness. However, researchers and banking authorities have recently chosen to train classifiers based on various machine learning and deep learning algorithms to automatically predict an applicant's credit score based on their credit history and other historical data, making the process of selecting eligible candidates much easier before the loan is approved. As a result, the goal of this paper is to discuss the application of several machine learning models to the aforementioned case in the loan lending process in order to determine the best approach for a financial institution that accurately identifies whom to lend to and assists banks in identifying loan defaulters for significantly reduced credit risk. Ensemble boosting, Random Forest and Decision Trees were the classifiers we utilized to create the model. They'll each be utilized individually to analyze the dataset, find patterns in it, and learn from them. Predict whether a new applicant is likely to default on a loan based on that information. The following sections make up the remainder of the paper. Section 2 contains a brief review of the approaches to credit risk analysis that have been employed. The Random Forest algorithm and Decision Trees are used in Section 3 to explain machine learning and detail the two methods used in the paper. The dataset used to train and evaluate the model is introduced in Section 4. Section 5 of this paper describes our methodology, which includes data analysis, dataset cleaning, and a walkthrough of the model we created utilizing the two classifiers. Section 6 summarizes the model's outcomes after training and testing, as well as the results of a comparative examination of the two classifiers' performance differences. Section 7 finishes the report with closing remarks and recommendations for future research. There are specific features to be used from the dataset

```
features = ['grade'          # grade of the loan
            'sub grade'      # sub-grade of the loan
            'short emp'      # one year or less of
            employment
```

```
            'emp length num' # number of years of
            employment
            'home ownership' # home ownership
            status: own, mortgage or rent
            'dti'            # debt to income ratio
            'purpose'        # the purpose of the loan
            'term'           # the term of the loan
            'last delinq none' # has borrower had a
            delinquency
            'last major derog none' # has borrower had
            90 day or worse rating
            'revol util'      # percent of available credit
            being used
            'total rec late fee' # total late fees received
            to day
        ]
```

Machine learning is an arena of computer science that involves the learning of pattern identification and computational learning theory in AI. Machine learning basically refers to the changes in systems that carry out tasks associated with artificial intelligence (AI). Such tasks embody recognition, analysis, planning, mechanism management, forecasting, etc. It explores the study and construction of formula that may build prediction on knowledge. Machine Learning is employed to make programs with its tuning parameters that are custom-made consequentially thus on increase their functioning by adapting to earlier data. Machine learning can be broken into two categories supervised unsupervised machine learning. the steps of building a machine are given below

A financial fraud is any criminal conduct committed by humans or robots that results in personal benefit at the expense of institutions or legitimate human benefactors, although an error should not be mistaken for a fraud. It is referred to as economic sabotage when considering the entire effect of financial scams. Money laundering, bank credit fraud, pension fraud, co-operative society fraud, tax evasion, telecommunications fraud, credit card fraud, inflated contracts, financial reports fraud, health insurance fraud, automobile insurance fraud, and mortgage insurance fraud are just a few examples of financial fraud. Credit card fraud, telecommunication fraud, computer infiltration, bankruptcy fraud, theft fraud or counterfeit fraud, and application fraud are all

examples of fraud, according to. The impact of fraud on a country's economy is real, and various measures have been tried, yet they all have flaws. On the other hand, machine learning has proven to be more reliable. Machine learning mining approaches leverage data to identify hidden patterns in big, dynamic, and diverse datasets and make intelligent judgments based on the disclosed insights. It's worth noting that, according to, a high rate of default has been documented in several countries, which can be lowered with the help of information technology. The following is how the rest of the paper is structured: Section 2 contains the materials and techniques, Section 3 contains the results and discussion, and Section 4 contains the paper's conclusion. Using tree structure techniques, the decision tree divides data into discrete categories. It emphasizes the structural information in the data and categorizes it from root to leaf node.

The advantages of employing decision trees are their unparalleled simplicity and quickness, there is no need for domain knowledge or parameter settings; it also comfortably handles high-dimensional data with many attributes; the way it is represented allows for enhanced comprehensibility; it has fantastic accuracy, though this is dependent on the data in use; it supports incremental learning; and it is unvaried because it is used based on a single feature at each interval node. They operate well for both classification and regression issues; they can manage missing information; trees are graphically plotted and easy to read; and, perhaps most importantly, trees are simple to explain to others. Credit default occurs when a client fails to meet the legal responsibilities or terms of a loan as stated in the promissory note. In other words, loan or credit default occurs when a borrower fails to repay a loan according to the terms agreed to prior to the loan's acceptance. A non-performing loan is one in which a borrower has taken out a certain amount of credit but has failed to pay back the loan in the agreed-upon time frame of 90 days for commercial banking loans and 180 days for consumer loans. Non-payment means that, depending on the type of loan, purpose, or industry, neither the interest nor the principle for that credit is paid within 90 to 180 days. Any definition of a non-performing loan is dependent on the loan's conditions and the existing agreement, as the definition is conditional and relies on promissory notes and agreements.

This Problem is done by mining the Big Data of the previous records of the people to whom the loan was granted before and on the basis of these records/experiences the machine was trained using the machine learning model which give the most accurate result. The main objective of this paper is to predict whether assigning the loan to a particular person will be safe or not. We have implemented this loan prediction problem using Decision tree algorithm and data cleaning in Python as there are missing values in the dataset. We use map function for the missing values. The aim of this paper is to apply machine learning technique on dataset which has 1000 cases and 7 numerical and 6 categorical attributes. The creditability of a customer for sanctioning loan depend on several parameters, such as credit history, Installment etc

1.2 Literature Review

This section summarizes some of the work that has already been done to develop machine learning models utilizing various algorithms to improve loan prediction and assist banking authorities and financial organizations in selecting a qualified candidate with very low credit risk.

Arutjothi, G., & Senthamarai, C. (2017). Prediction of loan status in commercial bank using machine learning classifier. 2017 International Conference on Intelligent Sustainable Systems (ICISS). The R programming language was utilised to create the K-NN credit scoring model since it has been widely used for academic purposes in the field of data mining. The R software packages used in creating the model are knn, ggplot2, ca Tools, and Desc tools. The credit scoring model is built using K-NN based on Euclidean distance. The data was divided into two subsets: 70% of the observations were utilized for training, while 30% were used for model testing. Furthermore, the partitioning was done at random such that both the training and testing sets include roughly the same percentage of default situations. Before splitting the dataset, we utilize Min-Max normalization to normalize it. We suggested a loan status model in this research to forecast whether a loan application is a genuine client or a default customer. Credit lenders may use this strategy to make a lending decision on loan proposals. Furthermore, the comparative research

was conducted with various levels of iteration. The iteration level 30 of the kNN model provides much higher accuracy than prior levels. This approach can be utilized to avert commercial banks' massive losses.

Mehul Madaan and colleagues 2021 This research, Loan default prediction using decision trees and random forest, employed two machine learning algorithms, Random Forest and Decision Trees, to develop a model for loan prediction and credit risk assessment. To have a better grasp of the accuracy and other scores of the two models, the results of both models are provided below, together with their classification report and confusion matrix. The Decision Tree classifier provided us with an accuracy score of 73%. The Random Forest Classifier provided us with an accuracy score of 80%. The purpose of this study was to investigate, analyse, and develop a machine learning algorithm that would properly determine if a person, given certain characteristics, has a high likelihood of defaulting on a loan. This type of algorithm might be used by Lending Club to discover certain financial characteristics of prospective borrowers who may be at danger of default. and fail to repay their debt by the due date The Random Forest Classifier delivered an accuracy of 80%, whereas the Decision Tree technique delivered an accuracy of 73%. As a result, the Random Forest model looks to be a superior choice for such data.

Li, X., Long, X., Sun, G., Yang, G., & Li, H. (2018). Overdue Prediction of Bank Loans Based on LSTM-SVM. 2018 IEEE Smart world, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation. This article evaluates whether users are overdue by using basic user information, bank data, user surfing activity, credit card billing records, and loan time information. These static data serve as the foundation for SVM. To forecast the probability of users' late behaviour, we extract the user's recent transaction type from browsing behaviour as input to the LSTM model. Finally, as the final result, we compute the average of the two techniques. According on the experimental data, this LSTM-SVM model outperforms previous techniques. Following the processing of the data set and the establishment of the model, the algorithm's correctness is tested. We

examined the aforementioned algorithms to compare the learning capacity of the various algorithms in user loan and the test's correctness. The findings are presented in the tables below.

Eweoya et al 2019 Journal of Physics: Conference Serie, Fraud prediction in bank loan administration using decision tree. Loan defaults have caused bank insolvency and nations to enter recession, which has a negative impact on individuals. Principal Component Analysis was used to extract important features (features engineering), cross validation was utilized to minimize overfitting in the model constructed, and data splitting was done to separate testing data from training data. This enables the model to test itself on a subset of the data that was previously unknown. The training and testing produced an accuracy of 75.9 percent with a high true positive ratio. In addition, based on a python [28] programmed built for this purpose, 80.04 percent of the occurrences were successfully categorised, and 129 of the testing data were detected as false. As shown in [29,30], the current study did not use hypothesis in its model formation and testing, instead employing a machine learning approach such as a decision tree in the model development and prediction.

Sheikh, M. A., Goel, A. K., & Kumar, T. (2020). An Approach for Prediction of Loan Approval using Machine Learning Algorithm. 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC).

Ke, L., Li, C., Zhong, T., Cai, Z., Wen, J., Wang, R., ... Tang, H. (2021). Loan Repayment Behavior Prediction of Provident Fund Users Using a Stacking-Based Model. 2021 IEEE 6th International Conference on Cloud Computing and Big Data Analytics. proposed a model based on stacking ensemble method, and compared with six individual models: Cart decision tree, Random Forest, AdaBoost, XGBoost, CatBoost and LightGBM. After experiments, our stacking model has a good improvement under the classification performance evaluation metrics such as accuracy score, recall score, F1 score, AUC score, and kappa score. Furthermore, we also analyzed the importance of features in the process of training the loan repayment behavior prediction model. Through our stacking model and research analysis, the

provident fund centers can reduce loan risks and provide better services to their customers. (ICCCBDA).

P M, S., & Paul, V. (2019). A Novel Optimized Classifier For the Loan Repayment Capability Prediction System. 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC). doi:10.1109/iccmc.2019.8819772

Amin, R. K., Indwiarti, & Sibaroni, Y. (2015). Implementation of decision tree using C4.5 algorithm in decision making of loan application by debtor (Case study: Bank pasar of Yogyakarta Special Region). 2015 3rd International Conference on Information and Communication Technology (ICoICT). doi:10.1109/icoict.2015.7231400

Singh, D. K., & Goel, N. (2020). Analyzing Data Mining Techniques on Bank Customers for Credit Score. 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO). doi:10.1109/icrito48877.2020.9197909

Patel, Bhoomi; Patil, Harshal; Hembram, Jovita; Jaswal, Shree (2020). [IEEE 2020 International Conference for Emerging Technology (INCET) - Belgaum, India (2020.6.5-2020.6.7)] 2020 International Conference for Emerging Technology (INCET) - Loan Default Forecasting using Data Mining.” have used data mining algorithms to predict the likely defaulters from a dataset that contains information about home loan applications, thereby helping the banks for making better decisions in the future.”, (), 1–4. Several algorithms were used in this work to forecast loan defaulters. Logistic Regression, Random Forest, Gradient Boosting, and CatBoost Classifier were used to achieve the best results. In comparison to Logistic Regression, the Gradient Boosting technique produces better or equal results. Gradient boosting is a procedure that combines many models. It is rare to remove a variable since the variables' meaning is not straightforward. On the other hand, even though it reduces overall model accuracy and prediction power, it is common practise to exclude variables while fitting logistic regression. CatBoost converts categorical values to numbers by employing a range of statistics on categorical feature

combinations and numerical and categorical feature combinations. With regard to the given dataset, the CatBoost classifier and Gradient Boost provide almost comparable accuracy. Furthermore, these models could also be utilizedaaaaaaaaa to form better lending selections and spare any financial institution from incurring massive losses.

Archana et al applied Decision Tree (DT), Support Adaptive Boosting Model (ABM), Vector Machine (SVM), rectilinear regression (LR), Neural Network (NN) and Random Forest (RF) algorithms for build loan predictive model on a banking data from UCI machine learning data repository, and showed that SVM had the simplest accuracy, but low run time performance, and tuned it to only incorporate most vital features

Zhang, Zenglian (2011). [IEEE 2011 3rd International Workshop on Intelligent Systems and Applications (ISA) - Wuhan, China (2011.05.28-2011.05.29)] 2011 3rd International Workshop on Intelligent Systems and Applications - Research of Default Risk of economic Bank's consumer loan supported Rough Sets and Neural Network. Results showed that rough set and BP neural network test samples of prediction accuracy during this paper, a billboard bank consumer loan default information, for instance , rough set of attributes went to streamline, re-use of BP neural networks, early warning of private loan default risk. The results showed that the reduction of condition attributes after reduction, but the accuracy of early warning is higher. (), 1–4.

Yu Jin and Yudan Zhu [7] applied different attribute selection methodologies: Random Forest and matrix within the model preparation stage to work out the highest attributes that determine loan default, and used different data processing classification techniques for prediction model building; Decision Tree (DT), Neural Network (NN) and Support Vector Machine (SVM) and concluded that SVM showed the simplest performance.

Amira Hassan and Ajith Abraham [5] used different algorithms with ensemble methods to create a loan default prediction model; scaled conjugate gradient backpropagation (SCG), Levenberg-Marquardt algorithm (LM), and One-step secant backpropagation

(OSS), and applied different filtering methodologies on dataset then used different parameters for comparison. They concluded that the Levenberg-Marquardt algorithm (LM) with PLs filter produced the simplest model.

Gupta, A., Pant, V., Kumar, S., & Bansal, P. K. (2020). loan Prediction System using Machine Learning. during this work, we use a machine learning technique which will predict the one that is reliable for a loan, supported the previous record of the person whom the loan amount is accredited before. This work's primary objective is to predict whether the authorization to a selected individual is safe or not. t. From the analysis of the info, it's very clear that it reduces all the frauds done at the time of authorization. 2020 9th International Conference System Modeling and Advancement in Research Trends (SMART).

Arutjothi, G.; Senthamarai, C. (2017). [IEEE 2017 International Conference on Intelligent Sustainable Systems (ICISS) - Palladam, India (2017.12.7-2017.12.8)] 2017 International Conference on Intelligent Sustainable Systems (ICISS) - Prediction of loan status in full service bank using machine learning classifier. propose a machine learning classifier based analysis model for credit data. We use the mixture of Min-Max normalization and KNearest Neighbor (K-NN) classifier. the target is implemented using the software package R tool. This proposed model provides the important information with the very best accuracy. it's wont to predict the loan status in commercial banks using machine learning classifier., (), 416–419.

Cheng, Dawei; Niu, Zhibin; Tu, Yi; Zhang, Liqing (2018). [IEEE 2018 24th International Conference on Pattern Recognition (ICPR) - Beijing, China (2018.8.20-2018.8.24)] 2018 24th International Conference on Pattern Recognition (ICPR) - Prediction Defaults for Networked-guarantee Loans. perform the empirical study on a real-world three years loan record from a serious full service bank. The results show that the proposed method outperforms conventional credit risk methods in terms of AUC. In summary, our quantitative risk evaluation model shows promising prediction performance on real-world data, which might be useful to both regulators and stakeholders. perform empirical experiments, give

the simplest parameter settings for the networked-guarantee loan record, and compare it with state-of-the-art credit rating approaches. Future work will include refining the approach and experime, (), 361–366. r, the most challenges are the imbalanced data and therefore the diffusion risk. We first compare our proposed P-w kNN with start-of-the-art imbalanced learning approaches, like SMOTE [5], over-sampling, under-sampling, One Sided Selection (OSS) [15], Edited Nearest Neighbor (ENN) [22] and Neighborhood Cleaning Rule (NCL) [3]. Figure 6 gives the AUC curves. it's clear that our proposed pw kNN outperforms classic approaches with a way larger AUC value. additionally, we increase the worth of k to ascertain it Maheswari, P.; Narayana, CH. V. (2020). [IEEE 2020 5th International Conference on Computing, Communication and Security (ICCCS) - Patna (2020.10.14-2020.10.16)] 2020 5th International Conference on Computing, Communication and Security (ICCCS) - Predictions of Loan Defaulter - a knowledge Science Perspective. This paper uses statistical measures to preprocess the info and build an efficient model which will predicts the loan defaulter accurately thanks to huge volume processing, built the models in cross validation approach using Grid Search CV. The classification techniques logistic regression, random forest and KNN model are built, thus far three algorithms result similarly. Among them Logistic regression with SGD training results better predictions than the others., (),

Sheikh, Mohammad Ahmad; Goel, Amit Kumar; Kumar, Tapas (2020). [IEEE 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC) - Coimbatore, India (2020.7.2-2020.7.4)] 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC) - the method of prediction starts from cleaning and processing of knowledge, imputation of missing values, experimental analysis of knowledge set then model building to evaluation of model and testing on test data. On Data set, the best-case accuracy obtained on the first data set is 0.811. the subsequent conclusions are reached after analysis that those applicants whose credit score was worst will fail to urge authorization, thanks to a better probability of not return the loan amount. Most of the time, those applicants who have high income and demands for lower amount of loan are more likely to urge approved

which is sensible, more likely to pay back their loans. another characteristic like gender and legal status seems to not be taken

and model validation strategies. Finally, chapters 6 and 7 discuss the results of this project and the conclusion made out of them respectively.

1.3 PROBLEM STATEMENT AND OBJECTIVE

- This paper aims to solve the major complexity faced by the banking sector in which not all the applicants are reliable to approve the loan
- The idea of this project is to gather data from the customer and use machine learning techniques on this data to extract important information and predict if a customer would be able to get a loan from a bank or not

1.4 Purpose, Scope and Applicability

1.4.1 Purpose

The objective of this thesis is to investigate which method from a chosen set of machine learning techniques performs the best default prediction. The specific algorithms included in the project are – decision tree, Ensemble Boosting

1.4.2. Scope

To determine if the bank can approve a loan with the help of machine learning the reasons for choosing these metrics will be explained in more detail in section 2

1.5 OVERVIEW OF THE REPORT

The next chapter will discuss the system analysis and requirements of this project followed by chapter 3, which discusses the datasets used for this project along with information on how the data was prepared to use in this project (like handling NAN values, etc.). Chapter 4 of this report discusses details on the model building that includes information on the model category, model training, model validation and model scoring. This chapter is followed by the chapter on model implementation which further discusses outliers

II. SYSTEM ANALYSIS AND REQUIREMENTS

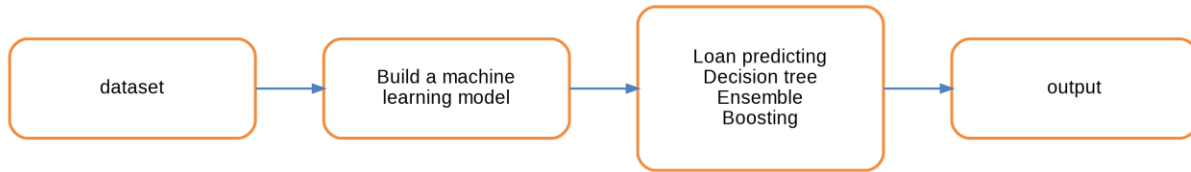
2.1 PROBLEM DESCRIPTION

It's a notion that ML allows robots to learn from real-world interactions and observations, act like humans, and improve their capacity to learn and perform with data provided as input. In recent years, machine learning has attracted the attention and interest of researchers and engineers, who are attempting to implement various machine learning models and algorithms in domains that will make a variety of critical activities and the lives of ordinary people much easier. The banking and finance sectors are two popular examples. Banking authorities and financial corporations are monitoring patterns and drawing conclusions in areas such as credit card fraud and loan default prediction with the use of various machine learning models. It has simplified and improved the procedure. The aforementioned models are built using a variety of machine learning techniques. It's practically hard to compile and present a comprehensive list of all machine learning methods. A model's name usually consists of a combination of data structure, design, estimator, ensemble method, and other terms. Random Forest and Decision Trees are the two machine learning techniques that are employed in this paper.

2.2 REQUIREMENT SPECIFICATION

Jupyter notebook is required to create and run a machine learning model. The code is written in Python 3.7 (or above). Installation of required libraries for the machine learning algorithms

BLOCK DIAGRAM



2.4 System Modules

Datasets to be used in building the model: “lending club data”. The dataset will be divided into a training and testing dataset. Then the dataset will be used to train machine learning models - like decision trees, and ensemble Boost for classifying whether the target bought from the marketing campaign. Then the outputs will be generated using the test data set

In this project, I used Landing Club data (<https://www.lendingclub.com/info/download-data.action>) to predict default bank loans.

3.2 Description of Dataset

The number of instances and attributes in the dataset are as follows: Number of instances: 122605
Number of attributes: 64

III. DATA DESIGN

3.1 Data Collection

Machine Learning Algorithm to Predict Default Bank Loans

The dataset was later divided into an “x-safe loans” and a “y” column, where “y” contains value of the target variable (whether the customer will be able pay the interest amount or not) containing binary values (0 or 1).

Feature name	Description	Data type
grade	grade of the loan	Categorical
Sub grade	sub-grade of the loan	Categorical
Short emp	one year or less of employment	numeric
Emp length num	number of years of employment	numeric
Home ownership	Home ownership status: own, mortgage or rent	Categorical
DTI	debt to income ratio	numeric
purpose	the purpose of the loan	Categorical
term	the term of the loan	Categorical
Last delinq none	has borrower had a delinquency	numeric
Last major derog none	has borrower had 90 day or worse rating	numeric
Revol util revol util	percent of available credit being used	numeric
Total rec late fee	total late fees received to day	numeric

3.3 Feature Selection

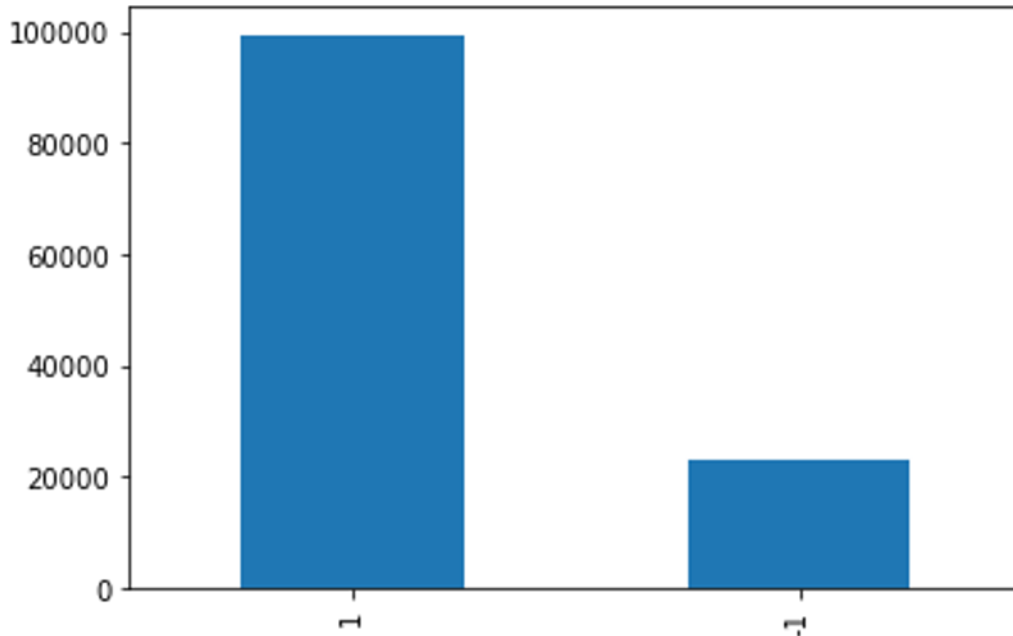
The target column (label column) of the dataset that I am interested in is called bad loan. In this column 1 means a risky (bad) loan 0 means a safe loan.

I put this in a new column called safe loan.

In order to make this more intuitive and consistent with the lectures, I reassigned the target to be:

The dataset has been created into two variable, feature column and target variable as safe loan which contain the target variable (whether the customer can repay the loan amount or not) containing binary values (-1 and 1) and the feature column contains the variable mentioned above.

- +1 as a safe loan,
- -1 as a risky (bad) loan.



3.4 Algorithm

Machine learning is the process of predicting and detecting patterns in order to generate appropriate results after fully comprehending them. Algorithms that use machine learning look for patterns in data and learn from them. With each attempt, an ML model will learn and improve. To determine a model's effectiveness, the data must first be divided into training and test sets. So, before training our models, we divided the Lending Club data into two sets: the Training set, which comprised 70% of the total dataset, and the Test set, which comprised the remaining 30%.

It was therefore necessary to apply a variety of performance indicators to our model's predictions. In this scenario, we were attempting to predict whether or not a person would default on a loan. Model accuracy may not be the only statistic to use to determine how well our model performed; the F1 score and confusion matrix should also be considered. What matters is that the appropriate performance measures be selected for the appropriate scenarios. For our modelling, we employed two algorithms:

IV. MODEL BUILDING

4.1 REVIEW OF MACHINE LEARNING ALGORITHMS

Decision Trees: Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too. The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data (training data). In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

Ensemble boosting: The ensemble boosting method used in our research was designed differently. Boosting has been modified by many researchers leading to the Adaboost method. In essence, that method focuses on the weak learners of every iteration over the data set $S = \{(x_i, y_i), i = 1 \dots n\}$ with n examples. During each iteration t , the algorithm tries to find a new classifier $h_t : x \rightarrow y$ while minimizing an error δ_t over the data set S_t . Throughout the iterations,

the Adaboost method gives more weight to the weak learned examples, but on the other hand, gives less weight to the correctly classified ones. These weights are initialized before starting iterations to $W_1(n) = 1$, $n \in \{1, 2, \dots, N\}$. In each iteration, we train a base learner h_m using the data set with distribution D_m and calculate the error δ_m such that: $\delta_m = \sum_{i=1}^N W_m(i) (1 - y_i, h_m(i))$ (2) $y_i, m = \{1, D_m \text{ gives the correct label } 0, \text{ otherwise } 1\}$ (3) It worth mentioning that $\delta_m \in (0, 0.5)$, therefore, in case that $\delta_m \geq 0.5$, the weights $W_m(i)$ are initialized to 1. Then, we calculate a parameter α_m that will be used to calculate the new weight for the iteration $m + 1$: $\alpha_m = 1 - \delta_m$ (4) $W_{m+1}(i) = W_m(i) \alpha_m (1 - y_i, h_m(i)) + \sum_{j=1}^M W_m(j) \alpha_m (1 - y_j, h_j(i))$ (5) The AdaBoost algorithm stops the ensemble construction process, presumably, at an iteration T , resulting in an ensemble learner of the classifiers h_1 to h_T . The Final classifier output could be given by: $H(x) = \arg \max_{y \in Y} \sum_{m=1}^T \log \frac{1 - \delta_m}{\delta_m} I(h_m(x) = y)$

4.2 MACHINE LEARNING MODEL

4.2.1 MODEL CATEGORY AND RATIONALE FOR CHOOSING THE SAME

Machine learning models will be used to evaluate the same problem of classifying whether the customer be able to repay the loan amount or not. Different models

are being used in order to understand how supervised algorithm explain the outcomes of the model.

4.2.2 MODEL NAME AND RATIONALE FOR CHOOSING THE SAME

The aim of the project is to observe the working of the algorithms with different instances of outputs. The following models are taken into consideration: Decision Trees, ensemble Boosting. Decision tree algorithms is considered to be white-box models (easy to interpret) whereas ensemble boosting is considered to be black-box models (tough to interpret). There are multiple kinds of machine learning models that could've been used for this project. Since it is not possible to incorporate all kinds of machine learning models, only the mentioned models were chosen. Any different model type can be used in the future work for this project.

4.2.4 THEORETICAL DESCRIPTION OF MODEL

The algorithms used in the project generate their feature attribution reports by creating a linear model in the vicinity of the datapoint mentioned to it and then tweaking input given to the model and receiving outputs from the model. To enable them, 3 different models are created upon the same dataset.

4.2.4 MODEL ARCHITECTURE



STEPS IN BUILDING THE MODEL

Required libraries and the dataset are imported into the Jupyter notebook.

Some features by default would be binary features, but since they have a third class "unknown" we'll process them as non-binary categorical features

The 2 mentioned models are then created (Decision Trees, Ensemble boosting)

The dataset is then split into 70:30 ratio for training and testing respectively.

After training, testing and validating the model, an instance of the dataset is taken for which we would like the interpretation to be created.

Required parameters are parsed through the explainer function to generate the explanations for an instance.

4.3 MODEL TRAINING

Best parameters are found with Grid Search CV and then passed to the model(s) to train the final estimator with these new-found parameters

4.4 MODEL VALIDATION

All the proposed models are then validated by scoring on the validation dataset.

4.5 MODEL SCORING

To score the models, a classification report on the validation dataset is generated and hence accuracy, precision, recall and f-1 score are produced to evaluate the model's performance.

Once the model has achieved a good score, then we move onto explaining the outcomes of the model using the interpretability algorithms

V. MODEL IMPLEMENTATION

5.1 DATA INVESTIGATION

The dataset consists of data from the customer detail of the previous bank transaction and their income. Upon this data, the aim is to build classifiers that can predict whether the client can properly repay the given loan amount or not. The dataset is publicly available on GitHub

In [91]: `loans.head()`

Out[91]:

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	term	int_rate	installment	grade	sub_grade	...	sub_grade_num	delinq_2yrs_zero	pub
0	1077501	1296599	5000	5000	4975	36 months	10.65	162.87	B	B2	...	0.4	1.0	
1	1077430	1314167	2500	2500	2500	60 months	15.27	59.83	C	C4	...	0.8	1.0	
2	1077175	1313524	2400	2400	2400	36 months	15.96	84.33	C	C5	...	1.0	1.0	
3	1076863	1277178	10000	10000	10000	36 months	13.49	339.31	C	C1	...	0.2	1.0	
4	1075269	1311441	5000	5000	5000	36 months	7.90	156.46	A	A4	...	0.8	1.0	

5 rows × 68 columns

5.2 DATA PREPARATION

The dataset was imported and checked for null values but it had none as can be seen in the figure below:

5.2.1 Checking for NAN values

In [4]: `loans.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 122607 entries, 0 to 122606
Data columns (total 68 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   id                                     122607 non-null  int64
1   member_id                             122607 non-null  int64
2   loan_amnt                             122607 non-null  int64
3   funded_amnt                            122607 non-null  int64
4   funded_amnt_inv                       122607 non-null  int64
5   term                                  122607 non-null  object
6   int_rate                              122607 non-null  float64
7   installment                           122607 non-null  float64
8   grade                                 122607 non-null  object
9   sub_grade                             122607 non-null  object
10  emp_title                             115767 non-null  object
11  emp_length                            118516 non-null  object
12  home_ownership                        122607 non-null  object
13  annual_inc                            122603 non-null  float64
14  is_inc_v                               122607 non-null  object
15  issue_d                               122607 non-null  object
16  loan_status                           122607 non-null  object
17  pymnt_plan                            122607 non-null  object
18  url                                    122607 non-null  object
19  desc                                  60705 non-null  object
20  purpose                               122607 non-null  object
21  title                                 122596 non-null  object
22  zip_code                              122607 non-null  object
23  addr_state                            122607 non-null  object
24  dti                                    122607 non-null  float64
25  delinq_2yrs                           122578 non-null  float64
26  earliest_cr_line                       122578 non-null  object
27  inq_last_6mths                         122578 non-null  float64
28  mths_since_last_delinq                 50500 non-null  float64
29  mths_since_last_record                 12531 non-null  float64
30  open_acc                               122578 non-null  float64
31  pub_rec                                122578 non-null  float64
32  revol_bal                              122607 non-null  int64
33  revol_util                             122607 non-null  float64
34  total_acc                              122578 non-null  float64
35  initial_list_status                    122607 non-null  object
36  out_prncp                              122607 non-null  float64
37  out_prncp_inv                          122607 non-null  float64
38  total_pymnt                            122607 non-null  float64
39  total_pymnt_inv                       122607 non-null  float64
40  total_rec_prncp                       122607 non-null  float64
41  total_rec_int                          122607 non-null  float64
42  total_rec_late_fee                    122607 non-null  float64
43  recoveries                             122607 non-null  float64
44  collection_recovery_fee                122607 non-null  float64
45  last_pymnt_d                           122271 non-null  object
46  last_pymnt_amnt                       122607 non-null  float64
47  next_pymnt_d                           2907 non-null  object
48  last_credit_pull_d                     122601 non-null  object
```

5.2.2 Feature Engineering

A new object Column transformer was defined that keeps the numerical features and apply one hot encoding on our categorical features that allows to

create a clean pipeline that includes both features engineering (one hot encoding here) and training the model.

```
In [13]: # Features to used in this analysis

features = ['grade',           # grade of the loan
            'sub_grade',       # sub-grade of the loan
            'short_emp',       # one year or less of employment
            'emp_length_num',  # number of years of employment
            'home_ownership',  # home_ownership status: own, mortgage or rent
            'dti',             # debt to income ratio
            'purpose',         # the purpose of the loan
            'term',            # the term of the loan
            'last_delinq_none', # has borrower had a delinquency
            'last_major_derog_none', # has borrower had 90 day or worse rating
            'revol_util',      # percent of available credit being used
            'total_rec_late_fee', # total late fees received to day
            ]

target = 'safe_loans'          # prediction target (y) (+1 means safe, -1 is risky)

# Extract the feature columns and target column
loans_data = loans_data[features + [target]]
```

5.3 MODEL TRAINING AND VALIDATION

All the three models were then built (Decision Tree, Random Forest, Ensemble boosting).

```
In [20]: train_data.shape
Out[20]: (37040, 13)

In [21]: validation_data.shape
Out[21]: (9260, 13)

In [22]: # Convert data into numpy

def get_numpy_data(data_frame, features, label):
    features_array = np.array(data_frame[features])
    label_array = np.array(data_frame[label])
    return(features_array, label_array)

train_feature_matrix, train_target_matrix = get_numpy_data(train_data, features, target)
validation_feature_matrix, validation_target_matrix = get_numpy_data(validation_data, features, target)
```

5.3.1 Data Splitting

To train the model, the dataset was divided into a training and testing ratio of 70:30 using sklearn's

method train test split. The following image shows the same

```
In [23]: print(train_feature_matrix[1:5,])
print(validation_feature_matrix[1:5])

[['B' 'B2' 0 7 'RENT' 24.21 'major_purchase' ' 36 months' 1 1 89.3
 14.9425]
['A' 'A2' 0 7 'RENT' 10.38 'credit_card' ' 36 months' 1 1 28.1 0.0]
['C' 'C4' 0 11 'RENT' 8.74 'debt_consolidation' ' 36 months' 0 1 34.4
 0.0]
['A' 'A1' 1 0 'RENT' 0.4 'major_purchase' ' 36 months' 1 1 2.3 0.0]]
[['C' 'C1' 0 11 'RENT' 20.35 'debt_consolidation' ' 60 months' 1 1 38.5
 0.0]
['B' 'B4' 0 3 'OWN' 17.35 'credit_card' ' 36 months' 1 1 90.3 0.0]
['C' 'C2' 0 6 'RENT' 17.07 'moving' ' 36 months' 1 1 84.7 0.0]
['B' 'B2' 0 11 'MORTGAGE' 13.96 'debt_consolidation' ' 60 months' 1 1
 29.1 0.0]]
```

5.3.2 Model specification

The decision trees model was trained in the same way by finding the best parameters and the fitting the model on the dataset.

Decision Tree

```
In [63]: from sklearn import tree
from sklearn.tree import DecisionTreeClassifier

dt = tree.DecisionTreeClassifier(max_depth=6)
clf = dt.fit(train_feature_matrix, train_target_matrix)
```

```
In [29]: from sklearn import tree
dtree = tree.DecisionTreeClassifier()
```

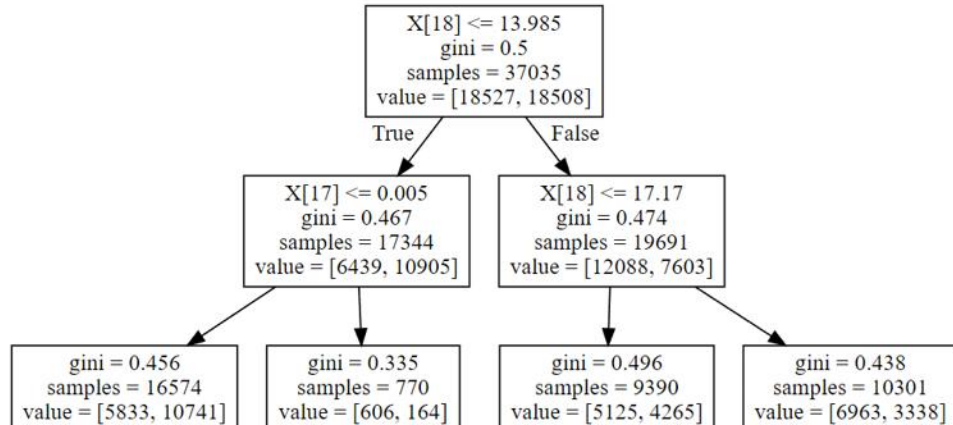
```
In [30]: tree.export_graphviz(clf, out_file='tree.dot')
```

```
dt = DecisionTreeClassifier(max_depth=2)
clf2 = dt.fit(train_feature_matrix, train_target_matrix)

from sklearn.tree import export_graphviz
import graphviz

export_graphviz(clf2, out_file="mytree.dot")
with open("mytree.dot") as f:
    dot_graph = f.read()
graphviz.Source(dot_graph)
```

Out[81]:



the data is continuously split according to a certain parameter it can also explain by two entities, namely decision tree nodes and leaves it force the consideration of all possible outcome of a decision and traces each path to a conclusion

Ensemble Boosting

```
In [40]: loans_data2 = loans[[target] + features].dropna()
```

```
In [41]: safe_loans_raw = loans_data2[loans_data2[target] == 1]
risky_loans_raw = loans_data2[loans_data2[target] == -1]

percentage = len(risky_loans_raw)/float(len(safe_loans_raw))
safe_loans = safe_loans_raw.sample(frac = percentage)

# Append the risky_loans with the downsampled version of safe_loans
loans_data2 = risky_loans_raw.append(safe_loans)
```

```
In [42]: safe_loans_raw = loans_data2[loans_data2[target] == 1]
risky_loans_raw = loans_data2[loans_data2[target] == -1]
```

```
In [43]: len(risky_loans_raw)
```

Out[43]: 23147

```
In [44]: len(safe_loans_raw)
```

Out[44]: 23147

```
In [45]: from sklearn.model_selection import train_test_split
train_data2, validation_data2 = train_test_split(loans_data2, test_size = 0.2)

In [46]: def get_numpy_data(data_frame, features, label):
features_array = np.array(data_frame[features])
label_array = np.array(data_frame[label])
return(features_array, label_array)

train_feature_matrix, train_target_matrix = get_numpy_data(train_data2, features, target)
validation_feature_matrix, validation_target_matrix = get_numpy_data(validation_data2, features, target)

In [47]: train_feature_matrix[1:2]

Out[47]: array([[ 'C', 1.0, 1, 1, 'MORTGAGE', 17.83, 'debt_consolidation', 7.8991,
0.0, 1.0, 1.0, 1, 1, 10.0, 0.0, 1.0, 52.4, 0.0, 16.78, 904.79,
62000.0, 16500, 16500, 408.12]], dtype=object)

In [85]: import sklearn
from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier(n_estimators=100,n_jobs=4)
clf5 = rf.fit(train_feature_matrix, train_target_matrix)

In [ ]:

In [86]: import sklearn.metrics as metrics

In [87]: predicted_rf = clf5.predict(validation_feature_matrix)
calculate_accuracy=("predicted_rf,validation_target_matrix")

In [ ]:

In [88]: importances = clf5.feature_importances_

In [89]: importances
std = np.std([tree.feature_importances_ for tree in clf5.estimators_],axis=0)
indices = np.argsort(importances)[::-1]
```

5.2.1 Model Validation strategies

The model will be validated on the test data set (30% of the dataset) and will be evaluated based on its accuracy and classification report.

The models were scored on the basis of their accuracy

Decision tree

The logistic regression model achieved an accuracy of 63%. Precision, recall and f1- score for the model are as follows

VI. RESULTS AND INTERPRETATION

6.1 PERFORMANCE METRICS

```
In [63]: from sklearn import tree
from sklearn.tree import DecisionTreeClassifier

dt = tree.DecisionTreeClassifier(max_depth=6)
clf = dt.fit(train_feature_matrix, train_target_matrix)
```

```
In [28]: predicted = clf.predict(train_feature_matrix)

num_mistakes = (predicted != train_target_matrix).sum()
num_correct = len(train_target_matrix) - num_mistakes
accuracy = num_correct/float(len(train_target_matrix))
print ('Accuracy on training data = %.2f' % accuracy)
```

Accuracy on training data = 0.64

```
In [82]: predicted = clf.predict(validation_feature_matrix)

num_mistakes = (predicted != validation_target_matrix).sum()
num_correct = len(validation_target_matrix) - num_mistakes
accuracy = num_correct/float(len(validation_target_matrix))
print ('Accuracy on validation data = %.2f' % accuracy)
```

Accuracy on validation data = 0.65

Ensemble Boosting

```
In [50]: from sklearn import tree
from sklearn.ensemble import GradientBoostingClassifier, GradientBoostingRegressor

dt = GradientBoostingClassifier(n_estimators=200, max_depth=3)
clf4 = dt.fit(train_feature_matrix, train_target_matrix)
```

```
In [51]: predicted = clf4.predict(validation_feature_matrix)

num_mistakes = (predicted != validation_target_matrix).sum()
num_correct = len(validation_target_matrix) - num_mistakes
accuracy = num_correct/float(len(validation_target_matrix))
print ('Accuracy on validation data = %.2f' % accuracy)
```

Accuracy on validation data = 0.68

Sl.no	Model used	accuracy
1	Decision tree	63%
2	Ensemble boosting	68%

In this prediction the model used to find the exact comparison the tested model in which the accuracy shows the model of the prediction which has been done from that the prefect model has been choosen

6.1 COMPARISON OF MODELS

Model 2 with total number of trees = 6 is 63% accurate on validation dataset.

In [36]: # ROC curve

```
from sklearn.metrics import roc_curve, auc
import matplotlib.pyplot as plt
import random

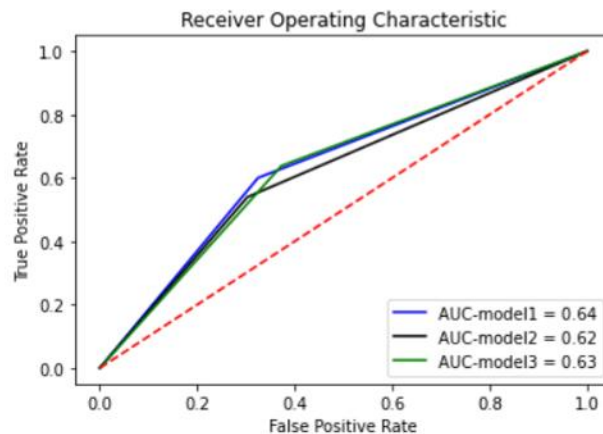
predicted1 = clf.predict(validation_feature_matrix)
false_positive_rate1, true_positive_rate1, thresholds1 = roc_curve(validation_target_matrix, predicted1)
roc_auc1 = auc(false_positive_rate1, true_positive_rate1)

predicted2 = clf2.predict(validation_feature_matrix)
false_positive_rate2, true_positive_rate2, thresholds2 = roc_curve(validation_target_matrix, predicted2)
roc_auc2 = auc(false_positive_rate2, true_positive_rate2)

predicted3 = clf3.predict(validation_feature_matrix)
false_positive_rate3, true_positive_rate3, thresholds3 = roc_curve(validation_target_matrix, predicted3)
roc_auc3 = auc(false_positive_rate3, true_positive_rate3)
```

```
In [37]: plt.title('Receiver Operating Characteristic')
plt.plot(false_positive_rate1, true_positive_rate1, 'b', label='AUC-model1 = %0.2f' % roc_auc1)
plt.plot(false_positive_rate2, true_positive_rate2, 'k', label='AUC-model2 = %0.2f' % roc_auc2)
plt.plot(false_positive_rate3, true_positive_rate3, 'g', label='AUC-model3 = %0.2f' % roc_auc3)
plt.plot([0,1],[0,1], 'r--')
plt.legend(loc='lower right')
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
```

Out[37]: Text(0.5, 0, 'False Positive Rate')



Ensemble Boosting

out the errors with the number of trees and classification errors

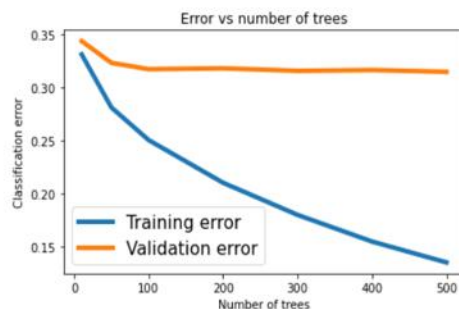
This method is based on the comparison of training error and validation errors in which the model will find

```
In [92]: def calculate_accuracy(predicted, true_label):  
        num_mistakes = (predicted != true_label).sum()  
        num_correct = len(true_label) - num_mistakes  
        accuracy = num_correct/float(len(true_label))  
        return(accuracy)
```

```
In [93]: ntree = [10,50,100,200,300,400,500]  
        training_err = []  
        validation_err = []  
  
        for i in ntree:  
            dt = GradientBoostingClassifier(n_estimators=i, max_depth=6)  
            clf = dt.fit(train_feature_matrix, train_target_matrix)  
  
            train_predict = clf.predict(train_feature_matrix)  
            t_err = 1 - calculate_accuracy(train_predict, train_target_matrix)  
            training_err.append(t_err)  
  
            validation_predict = clf.predict(validation_feature_matrix)  
            v_err = 1 - calculate_accuracy(validation_predict, validation_target_matrix)  
            validation_err.append(v_err)
```

```
In [94]: import matplotlib.pyplot as plt  
        %matplotlib inline  
        def make_figure(dim, title, xlabel, ylabel, legend):  
            plt.rcParams['figure.figsize'] = dim  
            plt.title(title)  
            plt.xlabel(xlabel)  
            plt.ylabel(ylabel)  
            if legend is not None:  
                plt.legend(loc=legend, prop={'size':15})  
            plt.rcParams.update({'font.size': 16})  
            plt.tight_layout()
```

```
In [55]: plt.plot([10,50,100,200,300,400,500], training_err, linewidth=4.0, label='Training error')  
        plt.plot([10,50,100,200,300,400,500], validation_err, linewidth=4.0, label='Validation error')  
  
        make_figure(dim=(10,5), title='Error vs number of trees',  
                    xlabel='Number of trees',  
                    ylabel='Classification error',  
                    legend='best')
```



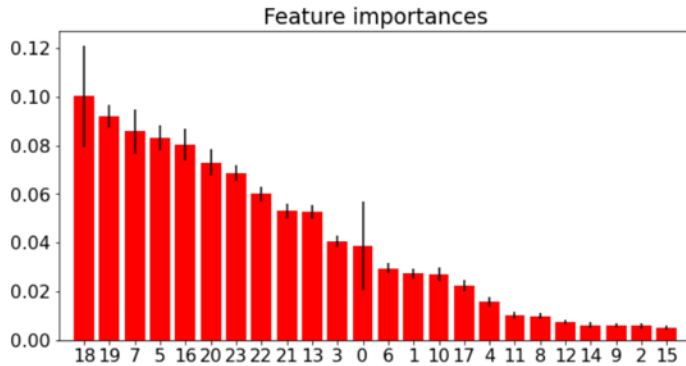
Boosting method improved accuracy to ~70% and is quite robust to overfitting.

```
In [62]: # Print the feature ranking
print("Feature ranking:")

for f in range(train_feature_matrix.shape[1]):
    print(features[indices[f]], importances[indices[f]])

# Plot the feature importances of the forest
plt.figure()
plt.title("Feature importances")
plt.bar(range(train_feature_matrix.shape[1]), importances[indices],
        color="r", yerr=std[indices], align="center")
plt.xticks(range(train_feature_matrix.shape[1]), indices)
plt.xlim([-1, train_feature_matrix.shape[1]])
plt.show()
```

```
Feature ranking:
int_rate 0.10017704045207844
total_rec_int 0.0918120800478945
payment_inc_ratio 0.08572177393252463
dti 0.08313702994532429
revol_util 0.08017479008289072
annual_inc 0.07284080452868719
installment 0.06849180835718252
funded_amnt_inv 0.05996582841561933
funded_amnt 0.05294552123165885
open_acc 0.05272985375458834
emp_length_num 0.04069102860079612
grade 0.038712458665732176
purpose 0.029548738613978543
sub_grade_num 0.027270011190817723
inq_last_6mths 0.027042671483867745
total_rec_late_fee 0.022298171559484388
home_ownership 0.01573073212985721
last_delinq_none 0.010346635946748848
delinq_2yrs 0.009926885949126323
last_major_derog_none 0.007261484453854746
pub_rec 0.0061661037929137035
delinq_2yrs_zero 0.006071446625051056
short_emp 0.005834836998648386
pub_rec_zero 0.005102263240674268
```



Interest rate, interest received to date, ratio of the monthly payment to income are some of the most important features to predict default loans.

VII. CONCLUSION

7.1 Research Project Overview

The goal of this study was to investigate, analyse, and develop a machine learning algorithm that could correctly determine whether a person, given certain characteristics, has a high risk of defaulting on a loan. Lending Club might utilise this type of model to identify specific financial characteristics of future borrowers who may be at risk of defaulting and not repaying their loan on time. The accuracy of the Decision Tree technique was 63 percent. As a result, the ensemble boosting gives us a 68 percent accuracy. When it comes to identifying potential borrowers who meet certain requirements, Lending Club must be cautious. Borrowers who do not own a home and apply for a small company or wedding loan, for example, may face a bad combination that leads to a default on the loan.

7.2 Evaluation of Results

This paper use different model to tested on a data set of bank on that basis a model is selected for the bank. Using that model bank can easily predict which customer will be beneficial for them and which can be defaulter or non-performing asset. Three different model is used here for prediction in which decision tree perform best on the data set which accuracy is higher than ensemble boosting.

7.3 Advantages and Limitations

One of the drawbacks is the limited number of people who defaulted on their debts in the first place during the course of the 8 years of data (2007-2015). We could look at how many of the current loans were paid off, defaulted, or even charged off using an updated data frame that includes the next three years' values (2015-2018). These fresh data points can then be used to make predictions or train new models for more accurate outcomes. Because the algorithm places certain non-defaulters in the default category, we should investigate this further to improve the model accurately identify capable borrowers.

VIII. FUTURE WORK

As this project could not incorporate all kinds of machine learning models, the other kinds of machine learning models can be used as the future work of this project. Apart from machine learning models, neural networks can also be attempted in to explain in the future using the mentioned algorithms. The problem type can also be changed from a classification type to a regression type or a clustering-based model that might enable a different point of view while observing the interpretation reports. Also, the algorithms are not just limited to these three algorithms.

REFERENCES

- [1] Mehul Madaan et al 2021 "Loan default prediction using decision trees and random forest": A comparative study, *IOP Conference*

- Series: Materials Science and Engineering: Mater. Sci. Eng.* 1022 012042. [IOP Publishing](#)
- [2] Arutjothi, G., & Senthamarai, C. (2017). Prediction of loan status in commercial bank using machine learning classifier. *2017 International Conference on Intelligent Sustainable Systems (ICISS)*.
- [3] Sheikh, M. A., Goel, A. K., & Kumar, T. (2020). An Approach for Prediction of Loan Approval using Machine Learning Algorithm. *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*. [IEEE](#)
- [4] Li, X., Long, X., Sun, G., Yang, G., & Li, H. (2018). Overdue Prediction of Bank Loans Based on LSTM-SVM. *2018 IEEE Smart World, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (Smart World/SCALCOM/UIC/ATC/CBD.Com/IOP/SC I)*. [IEEE](#)
- [5] P M, S., & Paul, V. (2019). A Novel Optimized Classifier For the Loan Repayment Capability Prediction System. *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*. [IEEE](#)
- [6] Amin, R. K., Indwiarti, & Sibaroni, Y. (2015). Implementation of decision tree using C4.5 algorithm in deciding of application by debtor (Case study: Bank pasar of Yogyakarta Special Region). *2015 3rd International Conference on Information and Communication Technology (ICoICT)*.
- [7] Singh, D. K., & Goel, N. (2020). Analysing data processing Techniques on Bank Customers for Credit Score. *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*. [IEEE](#)
- [8] Dawei Cheng*, Zhibin Niu†, Yi Tu* and Liqing Zhang* “Prediction Defaults for Networked-guarantee Loans” *2018 24th International Conference on Pattern Recognition (ICPR)* Beijing, China, August 20-24, 2018. [IEEE](#)
- [9] Bhoomi Patel¹, Harshal Patil², Jovita Hembram³, Shree Jaswal⁴ “Loan Default Forecasting using Data Mining” *2020 International Conference for Emerging Technology (INCET)* Belgaum, India. Jun 5-7, 202. [IEEE](#)
- [10] Xin Li, Xianzhong Long, Guozi Sun, Geng Yang, and Huakang Li Jiangsu Key Lab of Big Data and Security and Intelligent Processing Nanjing University of Posts and Telecommunications, Nanjing, 210023 “Overdue Prediction of Bank Loans Based on LSTM-SV” *2018 IEEE Smart World, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovations*.
- [11] Dr. Dahlia Sam¹, Dr. K. C. Suresh², Dr.N.Kanya³, C. Tamilselvi⁴, M.V.S.L Tejasria⁵ “AN IMPROVED BANK CUSTOMER CHURN AND LOAN PREDICTION MODEL USING SUPERVISED MACHINE LEARNING APPROACH” *Turkish Journal of Physiotherapy and Rehabilitation*; 32(3) ISSN.
- [12] Rising Odegua Department of Computer Science Ambrose Alli University Ekpoma, Edo state, Nigeria “Predicting Bank Loan Default with Extreme Gradient Boosting”. [arXiv](#)
- [13] Pratik Ratadiya¹, Khushi Asawa¹, Omkar Nikha¹ Dept. of Computer Engineering, Pune Institute of Computer Technology, Maharashtra, India. “A decentralized aggregation mechanism for training deep learning models using smart contract system for loan prediction”.
- [14] YA-QI CHEN, JIANJUN ZHANG, WING W. Y. NG Guangdong Provincial Key Lab of Computational Intelligence and Cyberspace Information, School of computing and Engineering “LOAN DEFAULT PREDICTION USING DIVERSIFIED SENSITIVITY UNDERSAMPLING”. [IEEE](#)
- [15] Ahmad Al-qerem Computer Science Department Zarqa University Zarqa, Jordan Ahmad_qerm@zu.edu.j “Loan Default Prediction Model Improvement through Comprehensive Preprocessing and Features Selection” *2019 International Arab Conference on Information Technology*.
- [16] G. Arutjothi Department of Computer Applications Government Arts College (Autonomous) Salem, India “Prediction of Loan

- Status in Commercial Bank using Machine Learning Classifier” *Proceedings of the International Conference on Intelligent Sustainable Systems (ICISS 2017)* IEEE Xplore Compliant - Part Number: CFP17M 19-ART, ISBN:978-1-5386-1959-9. [IEEE](#)
- [17] Amira Kamil Ibrahim Hassan Department of computing Sudan University of Science and Technology, Sudan Khartoum, Sudan “Modeling personal loan Default Prediction Using Neural Netware” 2013 *INTERNATIONAL CONFERENCE ON COMPUTING, ELECTRICAL AND ELECTRONIC ENGINEERING*.
- [18] Amira Kamil Ibrahim Hassan Department of computing Sudan University of Science and Technology, Sudan Khartoum, Sudan “Modeling personal loan Default Prediction Using Ensemble Neural Network” 2013 *INTERNATIONAL CONFERENCE ON COMPUTING, ELECTRICAL AND ELECTRONIC ENGINEERING (ICCEEE)*.
- [19] Shubham Chaudhary Student, IT Dept. of Galgotias College of Engg and Tech Greater Noida, UP “Loan Prediction System Using Decision Tree and Random Forest Algorithms” *International Journal of Emerging Technology and Innovative Engineering* Volume 6, Issue 06, June 2020 (ISSN: 2394 – 6598).
- [20] Pidikiti Supriya¹, Myneedi Pavani², Nagarapu Saisushma³, Namburi Vimala Kumari⁴, K Vikas⁵ “Loan Prediction by using Machine Learning Models” *International Journal of Engineering and Techniques* - Volume 5 Issue 2, Mar-Apr 2019.
- [21] Anirudh Bindal Department of computing and Engineering Manipal University Jaipur, Rajasthan anirudh312bindal@gmail.com “Predictive Risk Analysis For Loan Repayment of mastercard Clients” 2018 *3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT-2018)*, MAY 18th & 19th 2018.
- [22] Hossam Meshref, SMIEEE, computing Department College of Computers and knowledge Technology, Taif University Kingdom of Saudi Arabia “Predicting authorization of Bank marketing Data Using Ensemble Machine Learning Algorithms” *INTERNATIONAL JOURNAL OF CIRCUITS, SYSTEMS AND SIGNAL PROCESSING* DOI: 10.46300/9106.2020.14.117.
- [23] Eweoya et al 2019 *J. Phys.: Conf. Ser.* 1299 012037 “Fraud prediction in loan administration using decision tree” *Journal of Physics: Conference Series*.
- [24] P. Maheswari Department of CSE Lakireddy Bali Reddy College of Engineering (A) Mylavaram, Krishna, Andhra Pradesh “Predictions of Loan Defaulter - a knowledge Science Perspective”.
- [25] Kumar Arun, Garg Ishan, Kaur Sanmeet “Loan Approval Prediction supported Machine Learning Approach” *IOSR Journal of Computer Engineering (IOSR-JCE)* e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 18, Issue 3, Ver. I (May-Jun. 2016), PP 79-81.
- [26] Soni P M1 Professor, MCA, SNGIST N.Parur, Kerala, India “A Novel Optimized Classifier For the Loan Repayment Capability Prediction System” *Proceedings of the Third International Conference on Computing Methodologies and Communication (ICCMC 2019)* IEEE Xplore Part Number: CFP19K25-ART; ISBN: 978-1-5386-7808-4.
- [27] Yashna Sayjadah Computing & Technology Asia Pacific University Technology Park Malaysia Kuala Lumpur, Malaysia yashna.sayjadah@gmail.com “Credit Card Default Prediction using Machine Learning Techniques”.
- [28] Durgesh Kumar Singh Research Scholar, Department of Computer Applications, VBS PU, Jaunpur, “Analysing data processing Techniques on Bank Customers for Credit Score” 2020 *8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)* Amity University, Noida, India. June 4-5, 2020.
- [29] Xiao_jie Zhang School of Economy Shandong University of Technology' Zibo, China “Analysing data processing Techniques on Bank Customers for Credit Score” 2020 *8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)* Amity University, Noida, India. June 4-5, 2020.

- [30] Mehul Madaan et al 2021 *IOP Conf. Ser.: Mater. Sci. Eng.* 1022 012042 “Loan default prediction using decision trees and random forest: A comparative study” *IOP Conference Series: Materials Science and Engineering*. [IOP Publishing](#)