International Journal of Computer Engineering in Research Trends
Multidisciplinary, Open Access, Peer-Reviewed and fully refereed
Research Paper Volume-9, Issue-12, 2022 Regular Edition E-ISSN: 2349-7084

Predicting Possible Loan Default Using Machine Learning

[1] Isha Reddy ,[2] Madhavi Nirati , [3]K. Venkatesh Sharma

[1,2,]B.Tech , IV[th] Year , Department of Computer Science and Engineering, CVR College of Engineering, Vastunagar, Mangalpally, Ibrahimpatnam, T.S., India – 501510.

[3]Professor, Department of Computer Science and Engineering, CVR College of Engineering, Vastunagar, Mangalpally, Ibrahimpatnam, T.S., India – 501510

Email ID:[3]*venkateshsharma.cse@gmail.com*

Corresponding author : ***K. Venkatesh Sharma***
Available online at: http://www.ijcert.org

Received: 21/10/2022, Revised: 08/11/2022, Accepted: 18/12/2022, Published: 28/12/2022

Abstract:.. Loan lending has been an important business activity for both individuals and financial institutions. Profit and loss of financial lenders to an extent depend on loan repayment. Loan default prediction is a crucial process that should be carried out by financial lenders to help them find out if a loan can default or not. The aim of this paper is to use data mining techniques to bring out insight from data then build a loan prediction model using machine learning algorithms and find the best-suited model for the given dataset. The four algorithms used are Decision Tree Classifier, Random Forest Classifier, AdaBoost classifier, Bagged classifier, and Gradient Boost Classifier. The results show that the bagging classifier is the most stable model with the highest mean of weighted F1 scores and the least variance.

Keywords: Machine Learning, Loan Default, AdaBoost classifier, Bagged classifier, Gradient Boost Classifier.

---

1. Introduction

With increasing competition in the financial world and due to severe financial constraints, taking a loan has become certain. Individuals and organizations rely on loans for reasons such as overcoming financial limits to achieve their personal goals or for the basic purpose of managing their affairs in times when there are financial constraints. Though loan lending is quite beneficial for both the lenders and the receivers and is considered an essential part of financial transactions, it does carry some great risks. This risk is termed credit risk or loan default. Murray defines loan default as when a borrower does not make required payments or does not comply with the terms of a loan. Profit or loss of the financial lender to a large extent depends on loan repayments, that is whether customers are paying back the loans or not (defaulting). Therefore, when loans default, financial institutions will lose money, and it might even lead to bankruptcy and collapse of the institution. By predicting loan default, financial institutions (lenders) can reduce credit risk, prevent loan default and increase profit by evaluating the ability of the borrower to deliver on their obligation of loan repayment i.e., loan default prediction. The process of forecasting when a loan will default or not was initially done manually or semi-manually. With the advancement of statistical computing packages, several machine learning algorithms are used to calculate and predict loan default by evaluating an individual's historical data. But with an ever-increasing amount of data for loan default prediction, there is the need to use faster and more accurate algorithms. In this paper, we solve this problem by building high-performing machine learning classifier models using algorithms like decision tree classifier, random forest classifier, Gradient boost, Ada boost, and bagging classifier to predict loan default.

Founded in 2006, Lending Club is the world's largest peer-to-peer lender. They disrupted the traditional bank-based personal lending market by allowing retail investors to lend directly to individuals wanting to borrow. The Lending Club loan pool has grown steadily since its founding. In the last 3 years, the platform originated nearly $18B in new loans.

Lending Club is a lending platform that lends money to people in need at an interest rate based on their credit history and other factors. In this paper, we will analyze this data and pre-process it based on our need and build a machine learning model that can identify a potential defaulter based on his/her history of transactions with Lending Club. We wanted to take a deeper dive into the actual default rates within Lending Club so investors can make a more informed decision about their risks. We also wanted to see whether risk was pooled within a certain borrower type or loan purpose so we can make sure to have a diversified portfolio. We wanted to find the best classification model to determine whether we can beat the average total return for a randomly selected pool of loans.

This paper aims to demonstrate the application of machine learning in the finance industry. First, exploratory data analysis using data mining techniques is carried out to bring out insights from the dataset. Secondly, we employ machine learning algorithms and python libraries to make accurate loan default predictions. Five supervised machine learning classification algorithms are applied to predict loan default, and we achieve the highest accuracy of 99.62% using the Bagging classifier model.

The remaining paper is organized as follows: Section 2 represents a literature review; Section 3 presents a proposed model; Section 4 presents a result analysis; and Section 5 presents concussion.

2. Literature Review

Better predictive modeling is always needed in the banking industry. Predicting people who will default on their credit is a challenging task for the banking industry. One indicator of a loan's quality is its current status. While it's only the first step in getting a loan, it does reveal some important information. A credit scoring model is developed based on the current loan status. Credit defaulters and legitimate customers can both be identified with the help of the credit scoring model. This paper aims to develop a credit scoring model for credit records. Many different types of machine learning financial credit scoring models are created using a variety of methods. In this work, the author [1] proposes a technique for using machine learning. Credit data analysis using a classifier-based model in this paper, we employ a hybrid approach involving the Min-Max normalization and the K Nearest Neighbor (K-NN) classifier. The plan is in motion and the goal is being met. Using R, a piece of software. With the highest accuracy, this proposed model delivers the most crucial data. Machine learning classifiers are employed in commercial banks to make loan status predictions.

Traditional user loan risk prediction models, such as KNN, have seen their accuracy decrease as data volumes have grown, while Bayesian and DNN-based models have shown no such trend. This research [2] presented here was originally published at Overdue Bank Prediction. Artificially intelligent loans In addition, we propose using the LSTM algorithm to analyse dynamic user behaviour and the SVM algorithm to analyse static user data in order to resolve the existing prediction issues. Users' delinquency is determined through the analysis of demographic data, financial transactions, web browsing patterns, credit card payments, and loan repayment histories. These unchanging details serve as SVM's foundational input. Authors use an LSTM model to predict the likelihood of users' overdue behaviour, and we feed it information about the users' most recent transactions gleaned from their web browsing activities. In the end, we take an average of the two algorithms' outputs. The experimental outcomes demonstrate that this LSTM-SVM model outperforms the state-of-the-art algorithms by a wide margin.

Networked-guarantee loans may cause the systemic risk related concern of the government and banks in China. The prediction of default of enterprise loans is a typical extremely imbalanced prediction problem, and the networked-guarantee make this problem more difficult to solve. Since the guaranteed loan is a debt obligation promise, if one enterprise in the guarantee network falls into a financial crisis, the debt risk may spread like a virus across the guarantee network, even lead to a systemic financial crisis. In this paper, the author [3] proposes an imbalanced network risk diffusion model to forecast the enterprise default risk in a short future. Positive weighted k-nearest neighbors (pwkNN) algorithm is developed for the stand-alone case – when there is no default contagious; then a data-driven default diffusion model is integrated to further improve the prediction accuracy. We perform the empirical study on a real-world three years loan record from a major commercial bank. The results show that our 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 could be useful to both regulators and stakeholders.

3. Proposed Model

The aim of this study is to predict whether a new loan applicant will default a loan or not. This dataset collected from Lending club contains 42538 rows and 144 columns. Out of these 144 columns, many columns have null values in majority. We will analyze this data and pre-process it based on our need and build a machine learning model that can identify a potential defaulter based on his/her history of transactions with Lending Club. Firstly we try to understand our data. There are many columns with categorical and numerical values. There are many columns with a large no of missing values. To make this dataset fit for machine learning models to use we have to do extensive data pre-processing. After that we will split the data into training and testing data. Then we will train the training data with five machine learning models like decision tree, random forest, Adaboost, Gradient Boost and Bagging Classifier.

After implementing the models, we have to find the models that have the most accurate results when compared

 https://doi.org/10.22362/ijcert/2022/v9/i12/v9i1202

to others, so we will use some of the accuracy metrics for classification, which will help us in deciding which models are giving good results. The metrics we will be using are accuracy, precision, recall, and F1 score.

After finding the most accurate model, we will train the model with testing data and keep observing the performance. The figure below shows the proposed workflow for this study.

The below Figure 1` shows the proposed workflow for the loan default prediction

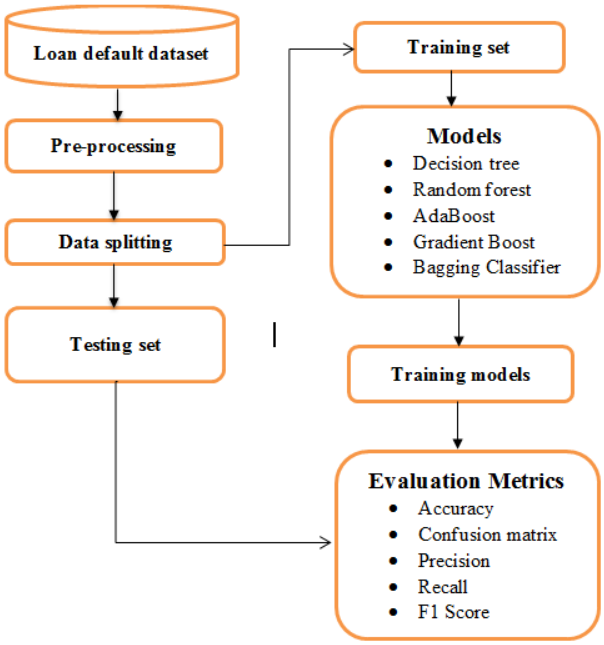

Figure 1: Proposed Model of the work

3.1 Dataset:

Lending Club is a lending platform that lends money to people in need at an interest rate based on their credit history and other factors. The dataset contains 42538 rows and 144 columns. Out of these 144 columns, many columns have null values in majority. In fact, 63.15% of the values in the overall data are null values. The attributes that have a major effect on the results are loan_status, sub_grade, funded_amnt_inv.

Data Preparation Process

The more disciplined you are in your handling of data, the more consistent and better results you are like likely to achieve. The process for getting data ready for a machine learning algorithm can be summarized in three steps:

Step 1: Select Data
Step 2: Preprocess Data
Step 3: Transform Data

You can follow this process in a linear manner, but it is very likely to be iterative with many loops.

3.2 Data Pre-Processing

Data pre-processing is a step in the data mining and data analysis process that takes raw data and transforms it into a format that can be understood and analyzed by computers and machine learning.

Raw, real-world data in the form of text, images, video, etc., is messy. Not only may it contain errors and inconsistencies, but it is often incomplete, and doesn't have a regular, uniform design.

Machines like to process nice and tidy information – they read data as 1s and 0s. So, calculating structured data, like whole numbers and percentages is easy. However, unstructured data, in the form of text and images must first be cleaned and formatted before analysis. When using data sets to train machine learning models, you'll often hear the phrase "garbage in, garbage out" This means that if you use bad or "dirty" data to train your model, you'll end up with a bad, improperly trained model that won't actually be relevant to your analysis.

Good, pre-processed data is even more important than the most powerful algorithms, to the point that machine learning models trained with bad data could actually be harmful to the analysis you're trying to do – giving you "garbage" results.

Data Preprocessing Steps:

1. Data quality assessment: There are a number of data anomalies and inherent problems to look out for in almost any data set, for example:

- **Mismatched data types:** When you collect data from many different sources, it may come to you in different formats. While the ultimate goal of this entire process is to reformat your data for machines, you still need to begin with similarly formatted data.
- **Mixed data values:** Perhaps different sources use different descriptors for features – for example, *man* or *male*. These value descriptors should all be made uniform.
- **Data outliers:** Outliers can have a huge impact on data analysis results. For
- **Missing data:** Take a look for missing data fields, blank spaces in text, or unanswered survey questions. This could be due to human error or incomplete data. To take care of missing data, you'll have to perform data cleaning.

2. Data cleaning: Data cleaning is the process of adding missing data and correcting, repairing, or removing incorrect or irrelevant data from a data set. Dating cleaning is the most important step of pre-processing because it will ensure that your data is ready to go for your downstream needs.

Data cleaning will correct all of the inconsistent data you uncovered in your data quality assessment. Depending on the kind of data you're working with, there are a number of possible cleaners you'll need to run your data through.

Missing data:

- There are a number of ways to correct missing data, but the two most common are:

 https://doi.org/10.22362/ijcert/2022/v9/i12/v9i1202

- Ignore the tuples
- Manually fill in missing data

Noisy data

Data cleaning also includes fixing "noisy" data. This is data that includes unnecessary data points, irrelevant data, and data that's more difficult to group together.

- Binning
- Regression
- Clustering

After data cleaning, you may realize you have insufficient data for the task at hand. At this point you can also perform data wrangling or data enrichment to add new data sets and run them through quality assessment and cleaning again before adding them to your original data.

3. Data transformation: With data cleaning, we've already begun to modify our data, but data transformation will begin the process of turning the data into the proper format(s) you'll need for analysis and other downstream processes.
This generally happens in one or more of the below:

1. Aggregation
2. Normalization
3. Feature selection
4. Discreditization
5. Concept hierarchy generation

Aggregation: Data aggregation combines all of your data together in a uniform format.
Normalization: Normalization scales your data into a regularized range so that you can compare it more accurately.
Feature selection: Feature selection is the process of deciding which variables (features, characteristics, categories, etc.) are most important to your analysis. These features will be used to train ML models. It's important to remember, that the more features you choose to use, the longer the training process and, sometimes, the less accurate your results are because some feature characteristics may overlap or be less present in the data.

Discreditization: Discretization pools data into smaller intervals. It's somewhat similar to binning but usually happens after data has been cleaned.
Concept hierarchy generation: Concept hierarchy generation can add a hierarchy within and between your features that weren't present in the original data.

Data reduction: The more data you're working with, the harder it will be to analyze, even after cleaning and transforming it. Depending on your task at hand, you may actually have more data than you need. Data reduction not only makes the analysis easier and more accurate but cuts down on data storage.
It will also help identify the most important features to the process at hand.

- **Attribute selection:** Similar to discreditization, attribute selection can fit your data into smaller pools.
- **Numerosity reduction:** This will help with data storage and transmission.
- **Dimensionality reduction:** This, again, reduces the amount of data used to help facilitate analysis and downstream processes.

3.3 Exploratory Data Analysis

A critical step in the data science process is the Exploratory Data Analysis or EDA. Examining your data thoroughly to understand the underlying data structure, is imperative to building good and even better models. There is no set on stone rules to exactly how to explore data. It is important to understand that the EDA process is iterative. It is not just a one-time step in the data science process, but rather something you come back to again and again, till the data science process is complete.

Look for missing values:
"*Are there missing values in my data?*" Once you have investigated the shape of the data set, identified the number of observations and independent variables or features, investigated the data types of each of the variables or features, you need to look for missing values in your data set. If any missing or '*Nan'* values are discovered, you are left with three options. Firstly, you can replace or impute them. Second option is to remove it altogether. Sometimes, you will encounter missing values for which any imputation might skew the distribution or the meaning of the data. Then, it might be best to remove the observation altogether. The third option is to keep it as is. Sometimes, missing values provide meaning in the data set and removing them could actually skew the distribution or the meaning of the observation. Then, keeping it as is, might be the best approach.

Look at the descriptive statistics:
"What are the descriptive points in my data?" Descriptive statistics describe the data within the subareas of central tendency, ma easure of spread or dispersion, and shapes of distributions. Generating the descriptive statistics of a dataset provides information on the key statistics; count, mean, standard deviation, min, ma,x and the values at different quartiles.

Look at the Variation:
"What type of variation occurs within my variables?" Variation is the measure of dispersion or spread of data. It tells you of the tendency of the values of a variable to change from observation to observation. The best way to understand variation of a variable is to visualize it. Use a barchart to visualize categorical variables or features and a histogram to examine the distribution of continuous variables or features. Other helpful data viz are scatterplots and boxplots. If you notice clusters of similar values with multiple peaks in the visuals, this might suggest that subgroups exist in your data. There may be certain unusual data points that linger far away or separately from all the other data points. These data points are called outliers.

Look at the correlation:
"What type of correlation occurs between my variables?" Correlation describes the relationship between two variables. It tells you how the change in variable A is related to the change in variable B. When variable A increases, does variable B also increase, decrease, or does not change? The correlation coefficient ranges from a value of 0 to 1. As the value approaches 1, the correlation increases or gets stronger. A positive value indicates a positive relationship while a negative value indicates an inverse or negative relationship. At 0, there is no relationship between the variables. Viewing the relation between two continuous variables can be achieved using scatterplots.
Realizing meaningful and significant relationships between variables can help in feature engineering and transformations. You can create new features depending on your end goal of a strong predictive model or a highly interpretable model etc.

3.4 Data Features

Converting of categorical columns to numerical columns: We have converted categorical columns to numerical by either performing one-hot encoding or label encoding depending on the kind of data they represent.

Converting Date Time columns to numerical columns: The columns ['issue_d','last_paymnt_d','last_credit_pull_d'] which are date 'int_ratetime' columns are further divided into month and year by using pandas datetime module. The new columns are named as 'issue_d_year', 'issue_d_month', 'last_paymnt_d_year', 'last_paymnt_d_', 'last_paymnt_d_month' respectively.

Converting objects to numerical columns: The columns 'int_rate' and 'term' are stored as objects. We have performed necessary string operations to convert them into numerical columns.

3.5 Feature Engineering

Feature engineering is the pre-processing step of machine learning, which is used to transform raw data into features that can be used for creating a predictive model using Machine learning or statistical Modelling**.** Feature engineering in machine learning aims to improve the performance of models. The predictive model contains predictor variables and an outcome variable, and while the feature engineering process selects the most useful predictor variables for the model.
These processes are described as below:

1. **Feature Creation**: Feature creation is finding the most useful variables to be used in a predictive model. The process is subjective, and it requires human creativity and intervention. The new features are created by mixing existing features using addition, subtraction, and ration, and these new features have great flexibility.
2. **Transformations**: The transformation step of feature engineering involves adjusting the predictor variable to improve the accuracy and performance of the model.
3. **Feature Extraction**: Feature extraction is an automated feature engineering process that generates new variables by extracting them from the raw data. The main aim of this step is to reduce the volume of data so that it can be easily used and managed for data modelling. Feature extraction methods include cluster analysis, text analytics, edge detection algorithms, and principal components analysis (PCA).
4. **Feature Selection:** While developing the machine learning model, only a few variables in the dataset are useful for building the model, and the rest features are either redundant or irrelevant. If we input the dataset with all these redundant and irrelevant features, it may negatively impact and reduce the overall performance and accuracy of the model. Hence it is very important to identify and select the most appropriate features from the data and remove the irrelevant or less important features, which is done with the help of feature selection in machine learning.

3.6 Model Building

Machine learning is a subfield of artificial intelligence, which is broadly defined as the capability of a machine to imitate intelligent human behaviour. With the help of sample historical data, which is known as training data, machine learning algorithms build a mathematical model that helps in making predictions or decisions without being explicitly programmed. Machine learning brings computer science and statistics together for creating predictive models. Machine learning constructs or uses the algorithms that learn from historical data. The more we will provide the information, the higher will be the performance.

A machine has the ability to learn if it can improve its performance by gaining more data. There are both supervised and unsupervised classifiers. Supervised learning is a type of machine learning method in which we provide sample labelled data to the machine learning system in order to train it, and on that basis, it predicts the output.

The system creates a model using labelled data to understand the datasets and learn about each data, once the training and processing are done then we test the model by providing a sample data to check whether it is predicting the exact output or not.Supervised and semi- supervised classifiers are fed training datasets, from which they learn to classify data according to predetermined categories.

The goal of supervised learning is to map input data with the output data. The supervised learning is based on supervision, and it is the same as when a student learns things in the supervision of the teacher.Unsupervised machine learning classifiers are fed only unlabelled datasets, which they classify according to pattern recognition or structures and anomalies in the data

 https://doi.org/10.22362/ijcert/2022/v9/i12/v9i1202

3.7 Classifiers

Classification belongs to the category of supervised learning where the targets also provided with the input data. Classification is the process of predicting the class of given data points. Classes are sometimes called as targets/ labels or categories. Classification predictive modeling is the task of approximating a mapping function (f) from input variables (X) to discrete output variables (y). There are many applications in classification in many domains such as in credit approval, medical diagnosis, target marketing, etc.
A classifier in machine learning is an algorithm that automatically orders or categorizes data into one or more of a set of "classes." Machine learning algorithms are helpful to automate tasks that previously had to be done manually.
They can save huge amounts of time and money and make businesses more efficient. A classifier is the algorithm itself the rules used by machines to classify data. A classification model, on the other hand, is the end result of your classifier's machine learning. The model is trained using the classifier, so that the model, ultimately, classifies your data.
Sentiment analysis is an example of supervised machine learning where classifiers are trained to analyze text for opinion polarity and output the text into the class: Positive, Neutral, or Negative.

Machine learning classifiers go beyond simple data mapping, allowing users to constantly update models with new learning data and tailor them to changing needs. Self-driving cars, for example, use classification algorithms to input image data to a category; whether it's a stop sign, a pedestrian, or another car, constantly learning and improving over time.

3.7.1 Decision Tree

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. The decisions or the test are performed on the basis of features of the given dataset.
It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions. It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm. A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees. This can be better understood by studying the below algorithm.

Algorithm:
Step-1: Begin the tree with the root node, says S, which contains the complete dataset.
Step-2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).
Step-3: Divide the S into subsets that contains possible values for the best attributes.
Step-4: Generate the decision tree node, which contains the best attribute.
Step-5: Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

This algorithm is easy to understand and can be used to solve decision-making problems. It helps to think about all the possible outcomes for a problem and there is fewer requirements of data cleaning as compared to other algorithms. The disadvantage of decision tree is that it is complex as it has many layers and may have overfitting issue.

3.7.2 Random Forest

3.7.3 Adaboost

AdaBoost is short for Adaptive Boosting. Boosting is an ensemble modeling technique that was first presented by Freund and Schapire in the year 1997, since then, Boosting has been a prevalent technique for tackling binary classification problems. These algorithms improve the prediction power by converting a number of weak learners to strong learners.

The principle behind boosting algorithms is first we built a model on the training dataset, then a second model is built to rectify the errors present in the first model. This procedure is continued until and unless the errors are minimized, and the dataset is predicted correctly.

Algorithm
The Adaboost working can be explained on the basis of the below algorithm:
Step 1 – Creating the First Base Learner
Step 2 – Calculating the Total Error (TE)
Step 3 – Calculating Performance of the Stump
Step 4 – Updating Weights
Step 5 – Creating a New Dataset

3.7.4 Bagging Classifier

Bagging is a technique for improving the accuracy of predictions made by a supervised learning algorithm. The basic idea is to train a number of different models on different randomly selected subsets of the training data, and then to combine the predictions of these models using some sort of voting scheme. The main advantage of bagging is that it can reduce the variance of the predictions made by a supervised learning algorithm without significantly compromising its accuracy. This makes it an attractive technique for problems where the cost of making a mistake is high (e.g., in medical diagnosis or credit card fraud detection), since it allows us to trade off some accuracy for increased robustness. The basic idea behind bagging is to train a number of different models on different randomly

 https://doi.org/10.22362/ijcert/2022/v9/i12/v9i1202

selected subsets of the training data. This can be done in several ways, but the most common is to use a different randomly selected subset for each model.

Once the models have been trained, we can combine their predictions using some sort of voting scheme. The most common way to do this is to take the majority vote, but there are many other options available.

Finally, we can apply the ensemble to new data in order to make predictions. This works by first splitting the new data into a number of training and testing sets, just as we did with the original training data. We then train the models on the training sets and combine their predictions using the voting scheme. The predictions for the testing sets are then averaged to get the final result.

3.7.5 Gradient Boost Classifier

Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees. When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms random forest. A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.

Algorithm:

Step 1: Initialize model with a constant value:

Step 2: For $m = 1$ to M:

1. Compute so-called *pseudo-residuals*:
2. Fit a base learner (or weak learner, e.g., tree) closed under scaling to pseudo-residuals, i.e., train it using the training set.
3. Compute multiplier by solving the following one-dimensional optimization problem:
4. Update the model.

5. Output

3.8 Evaluation Parameters

Accuracy - Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same. Therefore, you have to look at other parameters to evaluate the performance of your model. For our model, we have got 0.803 which means our model is approx. 80% accurate.

Accuracy = TP+TN/TP+FP+FN+TN

Confusion matrix - Confusion Matrix is a summary of predicted results in specific table layout that allows visualization of the performance measure of the machine learning model for a binary classification problem (2 classes) or multi-class classification problem (more than 2 classes)

Precision - Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is of all passengers that were labelled as survived, how many actually survived? High precision relates to the low false positive rate. We have got 0.788 precision which is pretty good.

Precision = TP/TP+FP

Recall - Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes. The question recall answers is: Of all the passengers that truly survived, how many did we label? We have got recall of 0.631 which is good for this model as it's above 0.5.

Recall = TP/TP+FN

F1 score - F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall. In our case, F1 score is 0.701.

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

4. Result and Analysis

From the above figure, we can say that the bagging classifier is the most stable model with the highest mean of weighted F1 scores and least variance.

Table 1.Classifciation report

	PRECISIONP	RECALL	F1-SCORE	SUPORT
0	0.75	0.62	0.67	1134
1	0.70	0.61	0.65	152
2	0.76	0.70	0.73	392
3	0.93	0.96	0.94	6824
accuracy			0.90	8502
Macro avg	0.78	0.72	0.75	8502

Weighted avg	0.89	0.90	0.89	8502

The below Figure 2. Shows the Final confusion matrix

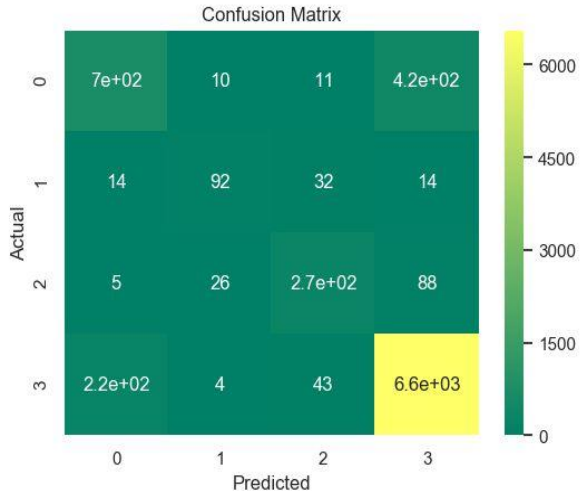

Figure 2. Final Confusion Matrix

4.1 Limitations and Future Research

The conducted study has limitations, which affect the study results. The data set consisted of three different Excel files. One data set contained information about the latest application; the second had information on previous applications; and the third contained a column description. Only the data set with information on the latest application was used. According to the researcher, the information on previous applications was irrelevant.

The data set used in the research had many important variables that are used in Finnish banks' loan applications. In turn, many of the variables had to be removed as descriptions were unclear or missing. Also, it was mentioned that the data set is a real-life data set, but the origin or country was not mentioned. The data set, however, did not include all important variables.A very interesting variable would have been the credit score. A credit score could be a number between 0 and 5, meaning that if an applicant has a credit score of 0, he or she has had problems with previous loans or account overdrafts. If the score is 5, the applicant has paid every payment when due and has maintained accounts accordingly.

Another limitation is the computing power of the computer the research was executed on. Oversampling the data set for random forest made the data too large, and it could not be executed. The issue arose when the second data set was attempted to be used for greater predictive power.

For future research, the loan default prediction should be executed for a mortgage data set. The rules and restrictions on mortgages are different and stricter than those on revolving loans. Revolving loans do not need collateral in Finnish society, so the loan-to-value ratio does not apply to them. The loan-to-value ratio is calculated based on the purchase price of the house or other collateral. A mortgage data set would enable studying the significance of the new restriction further. In addition, section 5.3.2 introduced the issue of values "365243" in the column "DAYS_EMPLOYED," which could be addressed in future research. The values were deleted from the data set in this research for simplicity. However, deleting the values led to deleting 51 232 observations, which was 71.7% of all deleted observations. Future research should address how to maintain these values.

5. Conclusion

The results suggest that lenders have various reasons to utilize machine learning in their loan application processes, and machine learning enables classifying the majority of qualified and unqualified applicants correctly. Previous research on loan defaults has compared different learning algorithms based on evaluation metrics similar to those in this research. This study was conducted with a literature review and an empirical study where different algorithms were compared. A literature review explained how machine learning is utilized in loan granting. In addition, it explained different types of machine learning, learning algorithms, how data is prepared, and how models are built for the predictions. The empirical study was executed with five machine learning models, and the aim was to identify the most powerful model. The models used in the study were the decision tree classifier, random forest, AdaBoost, gradient boost, and bagging classifier, which were compared based on chosen evaluation metrics like accuracy, precision, recall, and F1 scores. Finally, it is found that the bagging classifier gives the best results compared to the other models, with 89% accuracy and a 0.75 F1 score. It is followed by adaboost and gradient boost algorithms, but decision tree and random forest algorithms fall behind.

Based on the results obtained, machine learning-based models have shown a promising result in the prediction of loan default. It allows financial institutions (lenders) to be informed beforehand of defaults in issued loans, which will help them reduce financial loss and the cost associated with loan recovery. This will increase profits.

References

[1] Adewusi, A.O., Oyedokun, T.B., Bello, M.O.: Application of artificial neural network to loan recovery prediction. International Journal of Housing Marke Analysis (2016)

[2] Chambers, B., Zaharia, M.: Spark: The definitive guide: Big data processing made simple. " O'Reilly Media, Inc." (2018)

[3] Hamid, A.J., Ahmed, T.M.: Developing prediction model of loan risk in banks using data mining. Machine Learning and Applications: An International Journal (MLAIJ) Vol 3(1) (2016)

[4] Hassan, A.K.I., Abraham, A.: Modeling consumer loan default prediction using neural netware. In: 2013 INTERNATIONAL CONFERENCE ON COMPUTING, ELECTRICAL AND ELECTRONIC ENGINEERING (ICCEEE). pp. 239–243. IEEE (2013)
[5] Klaas, J.: Loan default model trap. https://www.kaggle.com/jannesklaas/modeltrap, (Accessed on 13/10/2021)
[6] Lai, L.: Loan default prediction with machine learning techniques. In: 2020 International Conference on Computer Communication and Network Security (CCNS). pp. 5–9. IEEE (2020)
[7] Marqu´es Marzal, A.I., Garc´ıa Jim´enez, V., S´anchez Garreta, J.S.: Exploring the behaviour of base classifiers in credit scoring ensembles (2012)
[8] Meer, K.: Machine learning models for mortgage default prediction in pakistan. In: 2021 International Conference on Artificial Intelligence (ICAI). pp. 164–169. IEEE (2021)
[9] Murray, J.: Default on a loan, united states business law and taxes guide national credit act (2005). act no. 34 of 2005, republic of south africa (2011)
[10] Odegua, R.: Predicting bank loan default with extreme gradient boosting. arXiv preprint arXiv:2002.02011 (2020)
[11] Patel, B., Patil, H., Hembram, J., Jaswal, S.: Loan default forecasting using data mining. In: 2020 International Conference for Emerging Technology (INCET). pp. 1–4. IEEE (2020)
[12] Reddy, M.J., Kavitha, B.: Neural networks for prediction of loan default using attribute relevance analysis. In: 2010 International Conference on Signal Acquisition and Processing. pp. 274–277. IEEE (2010)
[13] Rendle, S.: Factorization machines. In: 2010 IEEE International conference on data mining. pp. 995–1000. IEEE (2010)
[14] Sheikh, M.A., Goel, A.K., Kumar, T.: An approach for prediction of loan approval using machine learning algorithm. In: 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC). pp. 490–494. IEEE (2020)
[15] Turkson, R.E., Baagyere, E.Y., Wenya, G.E.: A machine learning approach for predicting bank credit worthiness. In: 2016 Third International Conference on Artificial Intelligence and Pattern Recognition (AIPR). pp. 1–7. IEEE (2016)
[16] Wang, B., Liu, Y., Hao, Y., Liu, S.: Defaults assessment of mortgage loan with rough set and svm. In: 2007 International Conference on Computational Intelligence and Security (CIS 2007). pp. 981–985. IEEE
[17] Loan Default Prediction Using Spark Machine Learning Algorithms Aiman Muhammad Uwais and and Hamidreza Khaleghzadeh

