Project Report [Group - 02]
CSE 445.2 - Machine Learning

Project Title: Bank Loan Approval Prediction using Machine Learning based classification

Submitted By

Project Members	Contribution Part
A. S. M. Sabiqul Hassan NSU ID - 1812442042 sabiqul.hassan@northsouth.edu	Design Methodology (data processing part), Results & Discussion , Conclusion & Future Work
Sazzad Hossain Sabbir NSU ID – 1612229042 sazzad.sabbir@northsouth.edu	Abstract, Introduction, Design Methodology (2 classification models)
Aiaj Uddin Bhuiyan NSU ID – 1621696042 aiaj.bhuiyan@northsouth.edu	Background, Design Methodology (2 classification models)

Submitted To

Syed Athar Bin Amir (SAA3)
Lecturer

ELECTRICAL AND COMPUTER ENGINEERING
NORTH SOUTH UNIVERSITY

Fall 2020

Table of Contents

Abstract
Many people are taking loans every day for different reasons. Their applications have to undergo many steps to fulfill the required criteria for loan approval. If we can build a better and reliable machine learning model for that purpose, it would make our life easier and reduce time to know the result of approval for a loan application.

Keywords
Bank Loan, Approval, Prediction, Machine Learning, Train and Test

Introduction
The banking sector is connected with our life in every aspect. Every day we need to exchange money as an individual or organization for different purposes. Sometimes we need to take some amount of loans to implement different projects. We request our desired loan amount into many financial organizations according to their rules and regulations.

After application our job is complete, the next part is the concern of the particular company we applied for a loan. They have to review every application and filter that according to their required criteria. Suppose, a bank assigned a digital team to increase their number of individual customers and provided them a record of the previous loan request. Now the main goal is to make a machine learning model that would be reliable and can be applied in the future to filter a loan application for approval.

Background [1]
One of the hottest trends right now is machine learning in banking. The eye-opening implications technological advances like machine learning have on construction loan administration and process automation are now reaching the surface. Many of us experience the outcomes of machine learning every day.

Historically, due to flexibility and familiarity, 33-year-old Excel has been the First Federal construction loan administration tool of choice for years. Meanwhile, recognition of potential errors and manual overload has often been overlooked. With the volume of work involved in administering their portfolio of loans,

First Federal has embraced the benefits of modern machine learning, process automation, and construction loan software. Through a combination of optical character recognition, computer vision, machine learning algorithms, and rule-based predictive modeling, loan management software parses the information from the countless emails and PDFs included within a construction loan draw request. It analyzes the data and performs “searches” to identify errors. Spreadsheets and forms capture the clean data, and the software creates recommendation reports based on rules and the data.

The applications for machine learning in banking are benefiting some of the most prominent banks in the world, as well as banks without billion dollars in IT budgets.

Design Methodology

This report utilized a methodology to predict the bank loan approval of an applicant. Firstly, we collected the dataset. Secondly, we pre-processed the dataset and prepared it to implement machine learning models. Finally, we applied some machine learning classification models to predict the result and claimed which model would perform better on that dataset.

Project Workflow

The workflow below describes the total process to make our desired machine learning model.

Dataset collection

Preprocess the dataset

- features selection
- categorical data handling
- missing numeric value handling

Dataset split into train and test

Train dataset with different classification ML models

Compare the better ML model according to the test result

Dataset

The dataset has been collected from Kaggle [2]. The dataset was divided into two CSV files train.csv and test.csv and there were about 70k instances for the training dataset and 30k instances for the test dataset.

Data Cleaning and Preprocessing

Features Selection

There was a total of 21 features before preprocessing our dataset. Moreover, our dataset was biased and we did the data processing part manually. We found that 7 features were irrelevant and similar to unique id. We ensured that those features wouldn't affect our trained model using the confusion matrix later.

The table below represents the most important features including the label of our dataset that we have used:

Feature Name	Description	Type
Gender	Gender of the applicant	String
City_Category	Anonymised City Feature	Character
Employer_Category1	Anonymized Employer Feature	Character
Employer_Category2	Anonymized Employer Feature	Integer
Monthly_Income	Monthly Income in Dollars	Integer
PrimaryBankType	Anonymised Bank Feature	Character
Contacted	Contact Verified (Y/N)	Character
Source_Category	Type of Source	Character
Existing_EMI	EMI of Existing Loans in Dollars	Integer
Loan_Amount	Loan Amount Requested	Integer
Loan_Period	Loan Period (Years)	Integer
Interest_Rate	Interest Rate of Submitted Loan Amount	Integer
EMI	EMI of Requested Loan Amount in dollars	Integer
Var1	Anonymized Categorical variable with multiple levels	Integer
Approved (Target)	Whether a loan is approved or not (1-0). Customer is Qualified Lead or not (1-0)	Integer

NB: Due to the wrong format of the date we could not use two features named 'DOB' and 'Lead_Creation_Date'. We could use them to find out the age of the applicants on the loan requested date which could be an important feature for our model.

Categorical Data Handling

There were categorical data in some features which are defined as character and string. We replaced those features using the one-hot encoding method [3].

One hot encoding is a process by which categorical variables are converted into a form that could be provided to ML algorithms to accomplish a better prediction.

Missing and Null Integer Data Handling

There were some null or missing values in numerical data. For example, loan_amont, EMI, loan_preriod, etc. features. Most of the time, we drop those columns containing some missing or null values but these features are important for our model and we couldn't drop them as the dataset was large enough which could impact the prediction result.

Therefore, we replaced those features value with the median value of that respective column. Now our dataset is pre-processed and ready to split into train and test set.

Dataset Split Part

As our training dataset was comparatively larger, we used the train.csv file for both train and test purposes. The train part contains 80% and the test part contains 20% of the total dataset.

Our next work is to use different machine learning models to train the dataset and test the prediction result.

Classification Models

There are four machine learning classification algorithms we applied for that project to predict the approval of a loan application, including logistic regression, decision tree classifier, random forest classifier, and XGBoost Classifier.

Decision Tree Classifier [5]

Decision Trees is a Supervised Machine Learning algorithm where the data is continuously split according to a certain parameter. It can be explained by two entities, namely decision nodes and leaves. The leaves are the decisions or the outcomes. And the decision nodes are where the data is split.

A decision tree classifier is one type of decision tree for discrete value prediction. For example, if we want to know if a person is fit for a job post then we have to measure him based on some criteria, like some specified skills, test results and the final selection outcome would be true or false. Those factors would be considered as a node of the classification decision tree and the leaf node would tell us the result.

XGBoost Classifier [6]

XGBoost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. In prediction problems involving unstructured data, artificial neural networks tend to outperform all other algorithms or frameworks. However, when it comes to small-to-medium structured/tabular data, decision tree-based algorithms are considered best-in-class right now.

XGBoost algorithm was developed as a research project at the University of Washington. Since its introduction, this algorithm has not only been credited with winning numerous Kaggle competitions but also for being the driving force under the hood for several cutting-edge industry applications. As a result, there is a strong community of data scientists contributing to the XGBoost open-source projects with ~350 contributors and ~3,600 commits on GitHub.

Logistic Regression [7, 8]

Logistic Regression is a Machine Learning algorithm that is used for classification problems, it is a predictive analysis algorithm and based on the concept of probability. We can call a Logistic Regression a Linear Regression model but the Logistic Regression uses a more complex cost function, this cost function can be defined as the 'Sigmoid function' or also known as the 'logistic function' instead of a linear function.
The hypothesis of logistic regression tends to limit the cost function between 0 and 1. Therefore linear functions fail to represent it as it can have a value greater than 1 or less than 0 which is not possible as per the hypothesis of logistic regression.

Actually, a cost function is a measure of how wrong the model is in terms of its ability to estimate the relationship between X and y. This is typically expressed as a difference or

distance between the predicted value and the actual value. The cost function (we may also see this referred to as loss or error.)

Random Forest Classifier [9, 10, 11]

Random forest is a supervised learning algorithm. The "forest" it builds, is an ensemble of decision trees, usually trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result. One big advantage of random forest is that it can be used for both classification and regression problems.

A random forest classifier is used to solve classification related problems. It consists of a large number of individual decision trees that operate as an ensemble. Each tree in the random forest spits out a class prediction and the class with the most votes becomes our model’s prediction. The fundamental concept behind random forest is a simple but powerful one — the wisdom of crowds. In data science, the reason that the random forest model works so well is a large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models.”

Results & Discussion

The report evaluates the algorithms’ performance by comparing some factors like accuracy, precision, recall, and f1 score on the test dataset. The table below represents those performance metrics of our designed model.

ML Model Name	Accuracy	Precision	Recall	F1 Score
Decision Tree Classifier	0.971168	0.046296	0.048544	0.047393
XGBoost Classifier	0.985010	0.285714	0.009709	0.018779
Logistic Regression	0.985226	0.500000	0.004854	0.009615
Random Forest Classifier	0.984078	0.055556	0.004854	0.008929

Accuracy: It is also known as classification accuracy. It is the ratio of the number of correct predictions to the total number of input samples. It works well only if our dataset is stratified. We can see that the accuracy of all the machine learning models is almost similar and about 98%.

F1 Score: F1 Score is used to measure the accuracy of the test dataset. It is the Harmonic Mean between precision and recall. The range for the F1 Score is [0, 1]. It describes how many instances it classifies correctly (measuring how precise it is), as well as it does not miss a significant number of instances (measuring how robust it is).

The greater the F1 Score, the better is the performance of our model. According to that, the f1 score of the decision tree classifier algorithm is about 4.7% which is greater than that of others.

Conclusion & Future Work

The report proposes a methodology to predict the approval for a loan application. As we observed the performance of all applied machine learning algorithm, we can claim that the decision tree classifier algorithm would work better on our dataset through the accuracy level are almost same in all cases. This designed model can be used at the production level for testing purposes and in the future, it would be available to use for commercial use after doing some tasks described below.

We have the plan to work in the future with a feature named 'Age_on_application_date' and we would try to implement k-fold cross-validation, hyperparameter tuning, and add neural network algorithms to make our designed model more efficient.

GitHub Code Link

https://github.com/SabiqulHassan13/cse445.2-fall20-project-dataset/blob/main/bank-loan-prediction-v1-C2.ipynb

References

1. https://rabbet.com/blog/machine-learning-in-banking-and-construction-loan-administration/
2. https://www.kaggle.com/arashnic/banking-loan-prediction
3. https://medium.com/m/global-identity?redirectUrl=https%3A%2F%2Ftowardsdatascience.com%2Fwhat-is-one-hot-encoding-and-how-to-use-pandas-get-dummies-function-922eb9bd4970
4. https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234
5. https://medium.com/swlh/decision-tree-classification-de64fc4d5aac
6. https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d
7. https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148
8. https://en.wikipedia.org/wiki/Logistic_regression
9. https://en.wikipedia.org/wiki/Random_forest
10. https://builtin.com/data-science/random-forest-algorithm
11. https://towardsdatascience.com/understanding-random-forest-58381e0602d2