ML GROUP ASSIGNMENT FINAL REPORT

pendyala@pdx.edu - Lasya Pendyala
thoutam@pdx.edu - Soumya Thoutam
gurram@pdx.edu - Mounisha Gurram

TABLE OF CONTENT:

INTRODUCTION:

Our lives have become easier with credit cards but to get an approval for one is a very long process. Banks need to undergo many background checks to give an approval for a credit card because without this company or banks may be bankrupt. Though it's a tough job for banks to be done, it is a must! To make the task a bit simpler we need a sophisticated method to automate the process and speed it up. This helps to avoid organization losses by avoiding potential defaulters. In our project we focused on personal attributes like gender, age, occupation etc to cut down the weeks-long process and provide a faster credit decision.
We extracted the data from the UCI Credit Approval Data Set to train the model. In the process of proceeding with the project challenges first we analyzed the data and did the data transformation. After exploratory analysis of data we prepared and applied different models to our transformed dataset, for this we split the dataset into train and test datasets. We allocated 75% to the training dataset and 25% to the testing dataset. After generating an analytic model the model is created to work on with the data to produce the required output.

FEATURE ENGINEERING:

1. Loading and viewing the dataset:

   We used the UCI Credit Approval DataSet from the UCI Machine Learning Repository. We started off by loading and viewing the dataset which is a mixture of both numerical and non numerical features.
2. Inspecting the dataset:

   Through the 1st step we got to know the features in a typical credit card application. The dataset has features like Gender, age, Debt, Married, BankCustomer. EducationLevel, Ethnicity, YearsEmployed, PriorDefault, Employed, CreditScore, DriversLicense, Citizen, ZipCode, Income and ApprovalStatus. During inspecting the data we felt DriversLicense and ZipCode are not so important features for predicting credit card approvals, so we've dropped them from the data.
3. Handling the missing data values:

   We've uncovered some issues that will affect the performance of our machine learning models if they go unchanged. There are few missing values in the dataset, these missing values in the dataset are labeled with '?', firstly we replaced the labels with NaN. In the process of treating the missing data values we secondly choose the mean imputation strategy to avoid the problem. Numerical data is handled by mean imputation strategy and for non-numerical data we have imputed missing values with the most frequent values since mean imputation would not work for categorical data.
4. Preprocessing the data:

   After handling the missing values in the data, minor preprocessing steps need to be processed to proceed towards building a machine learning model. In this preprocessing stage we first converted the non-numeric data into numeric data using label encoding technique because it faster the computation and also the algorithms developed using scikit-learn required the data to be in a strictly numeric format. After encoding the data the main task is to split the data into train dataset and test dataset. The next important step after splitting the encoded data is to scale the data. Here we scaled the feature value to a uniform range. This is the end of data preprocessing.

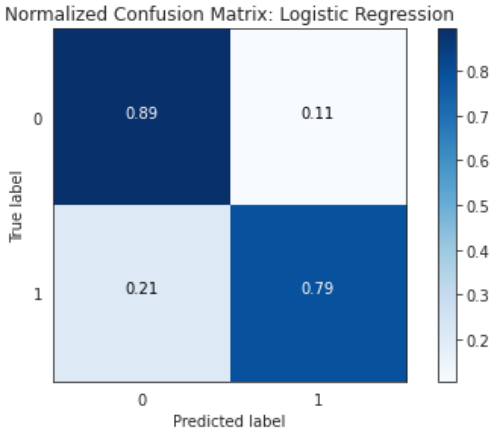
ALGORITHMS:

A good machine learning model should be able to accurately predict the status of the applications with respect to the feature statistics. In our project we worked on 4 different machine learning models to get the best results from the acquired dataset.

1. Logistic regression Model:

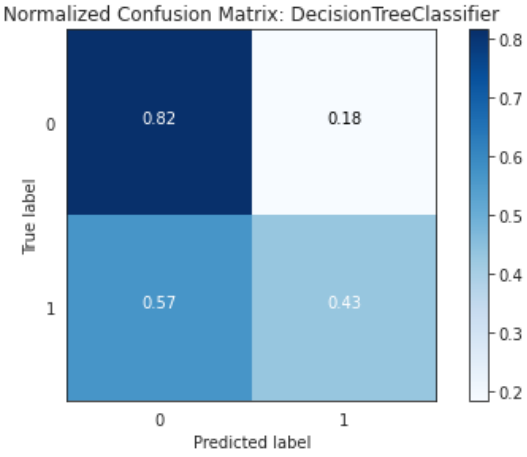
   We started off picking a machine learning model for our dataset with a generalized linear model i.e logical regression model. On performing a logistic regression algorithm on the dataset we got an accuracy score of almost 84%, to improve the models performance we performed a grid search on the model parameters. After grid search the accuracy we got

was 85%. We also plotted the confusion matrix for our model. Confusion matrix goes as below.

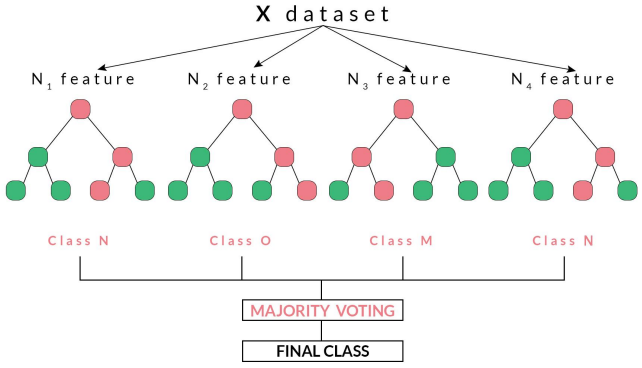


2. Decision tree Classifier:

   On performing a decision tree classifier model on the dataset, it was able to yield 60% of accuracy which was pretty bad compared to the logistic regression algorithm. After applying grid search the best accuracy the model could yield was 86%.

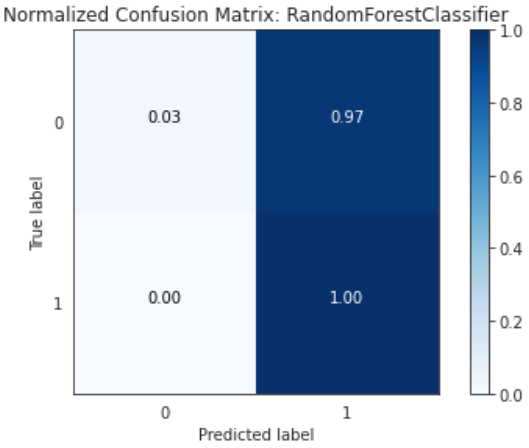


3. Random Forest Classifier:

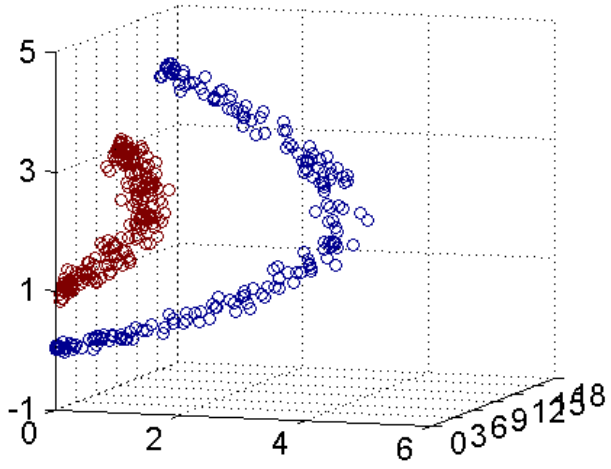


Random Forest model on dataset could yield 87% accuracy. Which is way better than

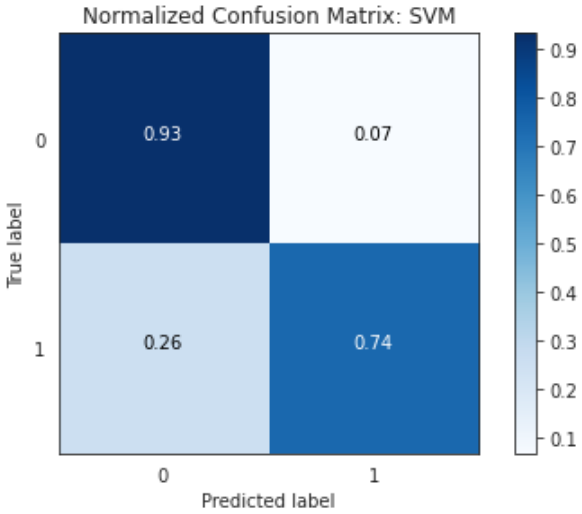
logistic regression and decision tree. Confusion matrix of the model goes as below.



4. Support Vector Machines (SVM) Model:



SVM model on the dataset gave us an accuracy score of 83% and the confusion matrix goes as below.



CONCLUSION:

By the end of our project we were able to preprocess the data and perform machine learning models on the preprocessed data. After performing four different machine learning models on the data we got to know that the Random Forest Classifier model yields the best performance when compared to other models. We plotted the accuracy scores of each algorithm we performed, it goes as below.