TERM PROJECT

LOAN DEFAULT PREDICTION

SUBMITTED BY: MAHNOOR SHEIKH

COURSE: FOUNDATIONS TO DATA SCIENCE

INSTRUCTOR: IMRAN NASEEM

Table of Contents

Page

Problem Statement ---------- 2

Databases ---------- 3

I. Loan Data Set ---------- 3
II. Default of Credit Card Clients ---------- 3

Methodology ---------- 4

I. Loan Data Set ---------- 5
II. Default of Credit Card Clients ---------- 7

Results ---------- 10

I. Loan Data Set ---------- 10
II. Default of Credit Card Clients ---------- 11

Conclusion ---------- 13

Future Work ---------- 13

Problem Statement

Loan lending is a product that generates profits for institutions through differential borrowing/ lending rates, and is a means of aid for individuals in need. It has been an important part of the daily lives for financial organizations and individuals alike, and this activity has become more or less inevitable with the growing financial constraints. However, it does carry great risks.

Credit Risk is the inability of the receiver to pay back the loan at the designated time which was decided by the lender and the borrower during the loan agreement. This causes major concerns among the financial institutes as it can result in “credit defaulting”, which can prove to be drastic to the lending party, as it may lead to losses, and even bankruptcy. A thorough evaluation and verification of the ability of a borrower to repay their loan in the decided time period can result in minimized credit risk, and will prove beneficial for financial institutes around the world.

Machine Learning models can be deployed to predict risky customers and hence minimize losses of the lenders. By using algorithms to study the behaviour and demographics of previous customers, we can apply the findings to customers in the future and differentiate between risky and non-risky customers, resulting in efficient loan lending. For this purpose, I have conducted this study to find the most suitable model for loan default prediction.

Databases

I have made use of two datasets to benchmark my results.

I. **Loan Data Set**
https://www.kaggle.com/burak3ergun/loan-data-set#loan_data_set.csv
This dataset contains information about Dream Housing Finance company which gives out housing loans. The dataset has 614 instances and 13 attributes. The dataset has multivariate characteristics, and the attributes have integer, categorical and real data types. The attribute summary is as follows:
Loan_ID: ID of each client
Gender: Categorical variable (Male, Female)
Married: Marital Status (Yes, No)
Dependents: Categorical variable (0, 1, 2, 3+)
Education: Categorical variable (Graduate, Not Graduate)
Self_Employed: Categorical variable (Yes, No)
ApplicantIncome: Income of the applicant
CoapplicantIncome: Income of the person who applied, along with the borrower, for the loan
LoanAmount: The amount of loan applied for
Loan_Amount_Term: The repayment time for the loan
Credit_History: Categorical variable for availability of client's credit history (1, 0)
Property_Area: Categorical variable (Urban, Semiurban, Rural)
Loan_Status: Default payment (Y, N)

II. **Default of Credit Card Clients**
https://www.kaggle.com/uciml/default-of-credit-card-clients-dataset
(I decided to use this dataset because credit cards are a type of loan. Personal loans and credit cards, both offer a way to borrow funds. You typically find funds offered from a lender at a specified interest rate, monthly payments that include principal and interest, late fees, underwriting requirements, amount limits, and more. The only things different are the interest rates and the interest-free period.)

This dataset contains information on credit card clients in Taiwan from April 2005 to September 2005. The dataset has 30,000 instances and 25 attributes. The dataset has multivariate characteristics, and the attributes have both integer, categorical and real data types. The attribute summary is as follows:
ID: ID of each client
LIMIT_BAL: Amount of given credit in NT dollars (includes individual and family/supplementary credit
SEX: Gender (1=male, 2=female)

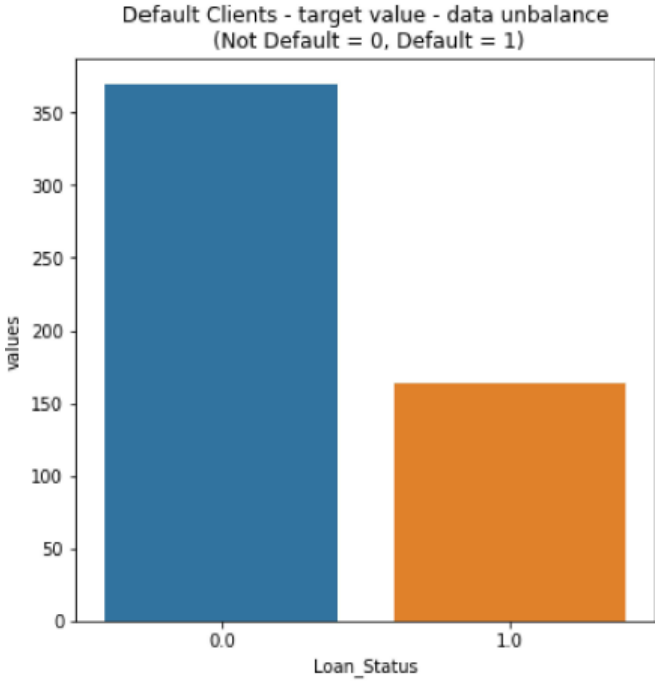
EDUCATION: (1=graduate school, 2=university, 3=high school, 4=others, 5=unknown, 6=unknown)
MARRIAGE: Marital status (1=married, 2=single, 3=others)
AGE: Age in years
PAY_0: Repayment status in September, 2005 (-1=pay duly, 1=payment delay for one month, 2=payment delay for two months, … 8=payment delay for eight months, 9=payment delay for nine months and above)
PAY_2: Repayment status in August, 2005 (scale same as above)
PAY_3: Repayment status in July, 2005 (scale same as above)
PAY_4: Repayment status in June, 2005 (scale same as above)
PAY_5: Repayment status in May, 2005 (scale same as above)
PAY_6: Repayment status in April, 2005 (scale same as above)
BILL_AMT1: Amount of bill statement in September, 2005 (NT dollar)
BILL_AMT2: Amount of bill statement in August, 2005 (NT dollar)
BILL_AMT3: Amount of bill statement in July, 2005 (NT dollar)
BILL_AMT4: Amount of bill statement in June, 2005 (NT dollar)
BILL_AMT5: Amount of bill statement in May, 2005 (NT dollar)
BILL_AMT6: Amount of bill statement in April, 2005 (NT dollar)
PAY_AMT1: Amount of previous payment in September, 2005 (NT dollar)
PAY_AMT2: Amount of previous payment in August, 2005 (NT dollar)
PAY_AMT3: Amount of previous payment in July, 2005 (NT dollar)
PAY_AMT4: Amount of previous payment in June, 2005 (NT dollar)
PAY_AMT5: Amount of previous payment in May, 2005 (NT dollar)
PAY_AMT6: Amount of previous payment in April, 2005 (NT dollar)
default payment next month: Default payment (1=yes, 0=no)

Methodology

I decided to apply multiple models on the datasets. This was done so that I was able to make an informed decision through the comparative analysis of all the models. My problem was a binary classification problem and so I used classifiers. Before the models were applied, each dataset went through a thorough process of Data Cleaning, Exploratory Data Analysis and Pre-processing. A total of six models were applied on each dataset, namely Logistic Regression, Decision Trees, Random Forest Classifier, Support Vector Machine (SVM), Naïve Bayes, and XGBoost Classifier.

Loan Data Set

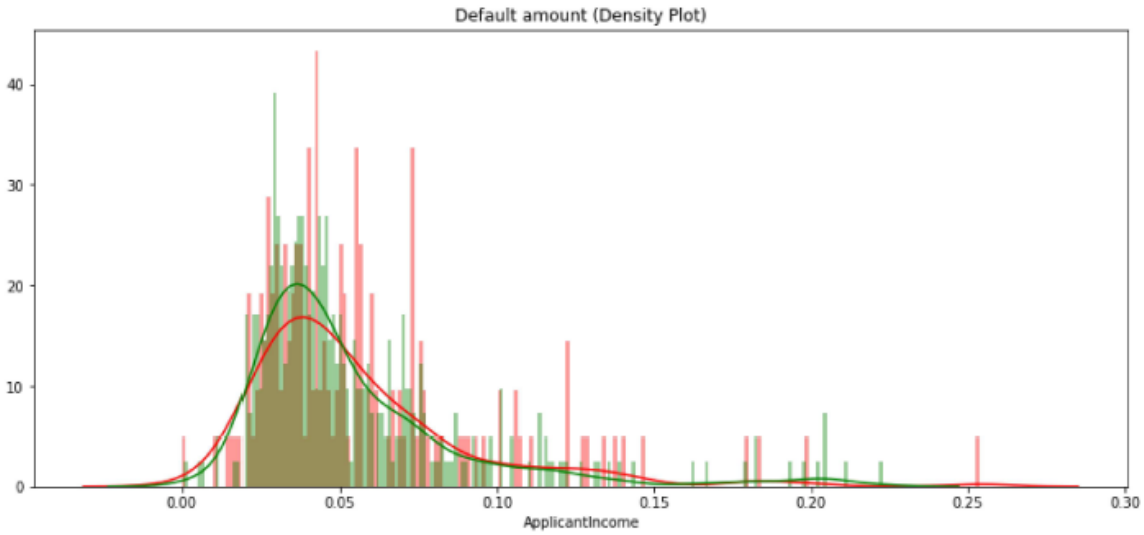
This was a comparatively small dataset with 614 instances and 13 attributes. I began with analyzing the data and putting it through a deep cleaning to make it easy for interpretation. I began by dropping the 'Loan_ID' attribute as it contained unique values and didn't contribute any information towards the final decision. I gave dummy values to categorical features with 2 categories, and turned 'Dependents' and 'Property_Area' into dummy variables, and dropped the duplicate columns. Upon checking for missing values, I found that the data had small percentage of missing values in 6 columns, the highest in 'Credit_History'. To fill these missing values, I decided to use K-Nearest Neighbor Imputation because it imputes missing values by matching a point with its closest k neighbors in a multi-dimensional space. I used this because loans with similar features would have the same default prediction. KNN Imputer works by measuring the distances between pairs of samples and these distances are influenced by the measurement units. Thus, as to not cause any bias during the clustering process, I normalized the data first using MaxMin normalization. Then, I removed any outliers in the data using Z-Scores.



During Exploratory Data Analysis, I obtained the description of the data, which included mean, median, standard deviation and other statistical measures to get a summary of all the attributes. I checked the distribution of the target variable and found that 69 percent samples belonged to the majority class (not default) and 31 percent samples belonged to the minority class (default). This showed that the data was mildly imbalanced, so I had to deal with this later.

I conducted a T Test and found that all numerical features had average importance, with 'CoapplicantIncome' being the most important, but none had statistical significance with respect to the target variable. On conducting a Chi-Square Test for categorical features, 'Credit-History', 'Property_Area_Semiurban' and 'Property_Area_Rural' were found to have statistical significance. This meant that all three of them held important information in regards to predicting a default client. 'Dependents_3+' had no importance so it was dropped. A correlation analysis found that no features had very strong correlations, i.e. greater than 0.95. (I tested the

models with different correlation cut-offs and 0.95 gave the best results. Any other value resulted in loss of information). Next, I checked the distribution of various variables with the 'Loan_Status' variable. I found that single people and graduates had a higher chance of defaulting. Even though, it makes more sense for non-graduates to default more, this trend could be explained by the graduates having a higher percentage in the dataset. I also found that those with and without dependents were equally likely to default. Moreover, the density plot for 'LoanAmount' had a normal distribution. Whereas, the density plot for 'ApplicantIncome' was right skewed and showed that those with low incomes had a higher defaulting tendency. This can be seen below. Everything seemed consistent and logically accurate in this dataset.



To prepare for model implementation, I separated the data into features and target variable. I split the data in a 7:3 ratio, i.e. 70% percent training data and 30% testing data. As the data was imbalanced, I applied Synthetic Minority Oversampling Technique (SMOTE) because it increased the minority class, while keeping the majority class the same so no important information was lost. I did not do Downsampling on this because the dataset was already small. Now I had two training datas to test the models on: Original data and SMOTE data.

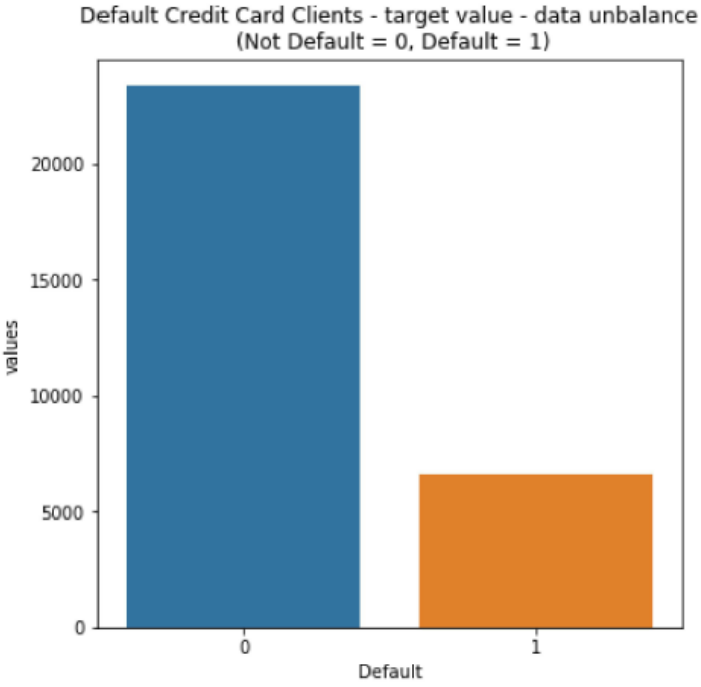
First, I applied Logistic Regression. I used GridSearchCV to find the best model parameters and fit them on the training data. The GridSearchCV tested the model with various values of 'c', an inverse regularization parameter that regulates against overfitting, and then used the best one. For the Decision Trees, I first tested the model with various depths. The training score increased with depth, but the testing score decreased. I fit the classifier at depth 2 because the testing score was the highest there, and the training score was good too. Next, I applied Random Forest Classifier with n_estimators=100. This gave the best results and was computationally fast too. I applied SVM with a Radial Basis Function (rbf) kernel. I did not apply Linear kernel as my data did not show a linear relationship (same reason why I did not use Linear Regression model). I tested SVM with increased values of C (penalty parameter)., and found that the rbf function with a low gamma of 0.01 and C=1000 gave the best results. At this value, the classifier starts to

become very intolerant to misclassified data points, and the decision boundary becomes less biased and has more variance, i.e. more dependent on the individual data points. After this, I fitted a simple Naïve Bayes model with default parameters as they gave the best results. Last was XGBoost Classifier. I used the Binary Logistic objective as the problem I was dealing with was a binary classification one.

Out of the 2 training datasets, all the models performed best on the original data, except XGBoost Classifier.

Default of Credit Card Clients

This was a large dataset with 30,000 instances and 25 attributes. I again began with analyzing the data and putting it through a deep cleaning. I removed the 'ID' attribute as it carried unique values and did not contribute any information towards the final decision. I renamed the target variable, i.e. default payment next month, to 'default' for ease in coding as I would be calling it frequently. The categorical variables (sex, education, marriage, default) had already been given dummy labels so I did not have to dummify these columns. Moreover, no missing data was found so I continued onto the data analysis.

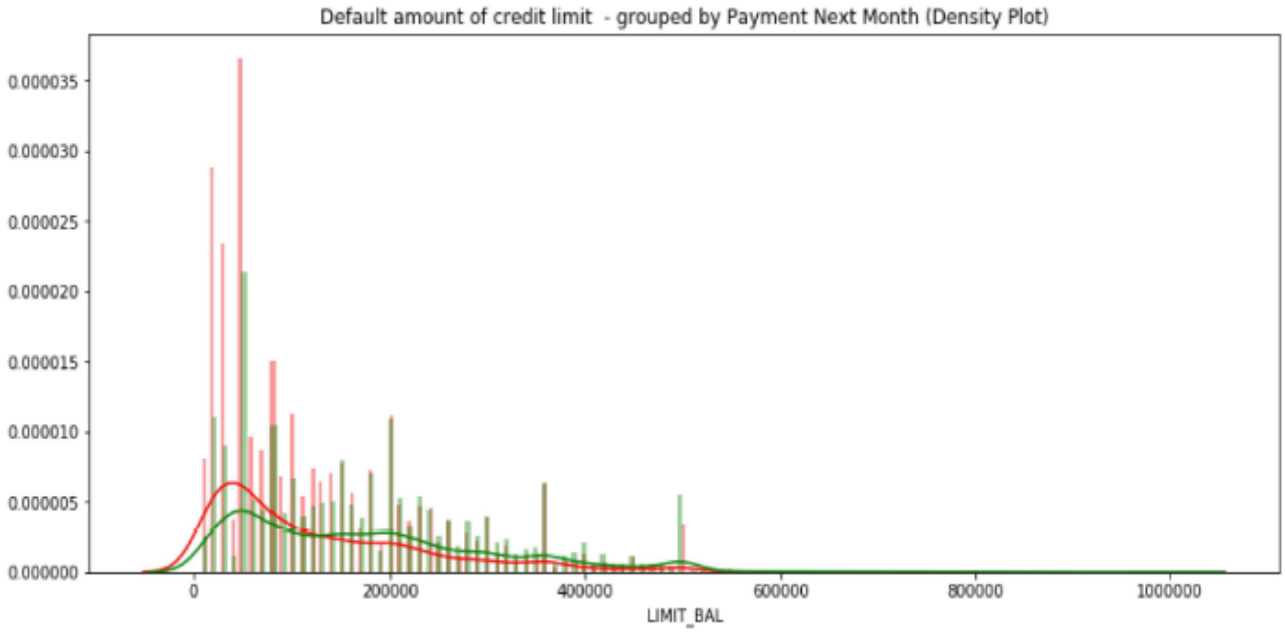


I obtained the description of the data, which included mean, median, standard deviation and other statistical measures to get a summary of all the attributes. I checked the distribution of the target variable and found out that 77 percent samples belonged to the majority class (not default) and 22 percent samples belonged to the minority class (default). This showed that the data was highly imbalanced, so I had to deal with this later.

On further analysis, I conducted a T test and found that all the numerical attributes had high importance, and all, with the exception of 'BILL_AMT4', 'BILL_AMT5', 'BILL_AMT6', showed statistical significance with respect to the 'default' variable. This meant that all of them held important information in regards to predicting a default client. For categorical features, I

used a Chi-Square test, which showed that all of them had statistical significance. Thus, no features were dropped till this point. Next, I held a correlation analysis and dropped all features with correlation greater than 0.95. Thus, the attribute 'BILL_AMT2' was removed.

Then, I checked the distribution of various variables with the 'default' variable. I found that the female population was in majority and was more likely to default, which could be explained by their high percentage in the dataset. Next, I found that University education level was highest in the dataset, and was most likely to default, which again could be explained by their high percentage in the dataset. Though, those least likely to default were unknown labels [0,4,5,6]. Logically speaking, they needed to be above graduate level at least as education level and likeliness of default has an inverse relationship. I tried removing these unknowns thinking they were errors, but this reduced the accuracy of the models. Thus, I believed that they carried important information and treated them as separate variables. I also found that single people were highest in number and were most likely to default, which made sense. The '0' value in this column was unknown though. I tried removing these '0' values from the 'Marriage' variable but this reduced model accuracy. Thus, I inferred that they too carried important information and treated them as separate variables. Furthermore, I plotted a density plot for the credit limit variable, and found that it was right skewed so higher the credit limit, lower was the chance of default. This was sensible as richer people tend to have higher credit limits and are less likely to default on loans. The highest defaulters were for credit limit 0 to 100,000, with the highest being for credit limit 50,000, and the density for this interval was larger for defaulters than for non-defaulters, as shown in the figure below. Last, I checked the 'PAY_0' feature. There was a '-2' and '0' value in all 'PAY' features which was not quoted in the data description. I was not able to explain it on inspection of the dataset too so I didn't remove it. This dataset seemed to have many errors.



Default	0	1
EDUCATION		
0	14	0
1	8549	2036
2	10700	3330
3	3680	1237
4	116	7
5	262	18
6	43	8

Default	0	1
MARRIAGE		
0	49	5
1	10453	3206
2	12623	3341
3	239	84

Default	0	1	perc
PAY_0			
-2	2394	365	0.132294
-1	4732	954	0.167781
0	12849	1888	0.128113
1	2436	1252	0.339479
2	823	1844	0.691414
3	78	244	0.757764
4	24	52	0.684211
5	13	13	0.500000
6	5	6	0.545455
7	2	7	0.777778
8	8	11	0.578947

During Pre-processing, I separated the data into features and target variable, and split the data in 7:3 ratio. I scaled the data using Robust Scaler to bring the data on the same scale and remove any chance of biasness. Only the data containing features was scaled, and the training and testing sets were scaled separately to avoid data leakage. To fix the data imbalance, I made 2 more datasets. As it was a large dataset, I applied Synthetic Minority Oversampling Technique (SMOTE) and Downsampling using 'resample'. I used SMOTE because it increased the minority class, while keeping the majority class same so no information was lost. Downsampling reduces the majority class to equal the size of the minority class. Important information may be lost this way but I computed it for the sake of comparison as the dataset was huge. Now I had 3 training datasets to test the models on: Original data, SMOTE data, and Downsampled data.

More or less, the same procedure was followed as that on the previous dataset. First, I applied Logistic Regression. I again used GridSearchCV to find the best model parameters and fit them on the training data. For the Decision Trees, I first tested the model with various depths. The training score increased with depth, but the testing score increased till a certain point, and then began decreasing. I fit the classifier at depth 4 because the testing score was the highest there, and the training score was good too. Then I applied Random Forest Classifier with n_estimators=100. This gave the best results as any number of estimators more than this made the model extremely slow due to the large dataset. After this, I applied SVM with Radial Basis Function kernel. The rbf function with a low gamma of 0.01 and C=1 gave the best results. Any penalty parameter (c) larger than this made the model become extremely slow. Next, I fitted a simple Naïve Bayes model with default parameters as they gave the best results. Last I applied the XGBoost Classifier with Binary Logistic objective.

Out of the 3 training datasets, all the models performed best on the original data.

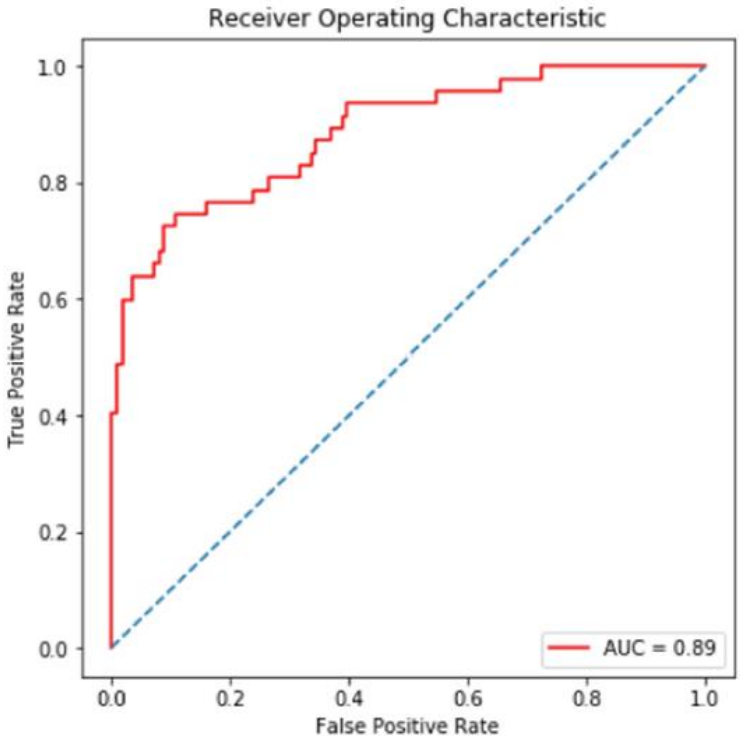
Results

For both datasets, results differed greatly. I used various testing parameters to test the data on: Model Accuracy, Confusion Matrix, Precision, Recall, F1 score, ROC curve and the AUC score. (F1 score measure ranges from 0-1. The closer the score is to 1, the better the precision and recall are.)

Loan Data Set

Logistic regression got an accuracy of 0.82 on original data and 0.81 on SMOTE data. The f1 score for both majority (0.88) and minority (0.70) class was better for the original data, and the AUC received on this was 0.88. Decision Tree got the same accuracy on both original and SMOTE data. The accuracy was 0.85, and the f1 score for both classes (0.90, 0.69) were better than logistic regression. The AUC score was 0.82. Random Forest Classifier also got better results on original data. An accuracy of 0.87 was reached, and the f1 score was a lot better than that received on SMOTE data for both classes (0.91, 0.75). The AUC score was 0.87. SVM gave slightly better accuracy on original data (0.86) than SMOTE data (0.83). The f1 score for original data was better (0.91, 0.71). The AUC score, however, reduced to 0.78 compared to previous models. Naïve Bayes performed best on original data with an accuracy of 0.86, and an f1 score of 0.91 on majority class and 0.71 on minority class. The AUC score was the highest, 0.88. Lastly, XGBoost Classifier performed slightly better on SMOTE data compared to original data in all aspects. Accuracy on SMOTE data was 0.87, whereas that on original was 0.86. The f1 score was also a lot better on SMOTE data with 0.91 on majority class and 0.74 on minority class (f1 score on original data's minority class was 0.71). The AUC score differed between the two by 0.01, and was better on SMOTE data, i.e. 0.89.

In summary, models' performance in terms of **accuracy (highest to lowest):** XGBoost, Random Forest/SVM/Naïve Bayes, Decision Trees, Logistic Regression; **AUC score (highest to lowest):** XGBoost, Logistic Regression/Naïve Bayes, Random Forest, Decision Trees, SVM. Moreover, XGBoost had the highest f1 score on majority class (0.91) and a high f1 score on minority class (0.74). Even though, Random Forest Classifier had a slightly higher f1 score on minority class of 0.75, but we need to consider that XGBoost outperformed it in all other ways. Moreover, XGBoost successfully addressed the class imbalance problem by giving better results on SMOTE data. Thus, XGBoost Classifier outperformed all the models in every aspect and is the most suitable model for loan default prediction in this case. The ROC curve for this is shown below.

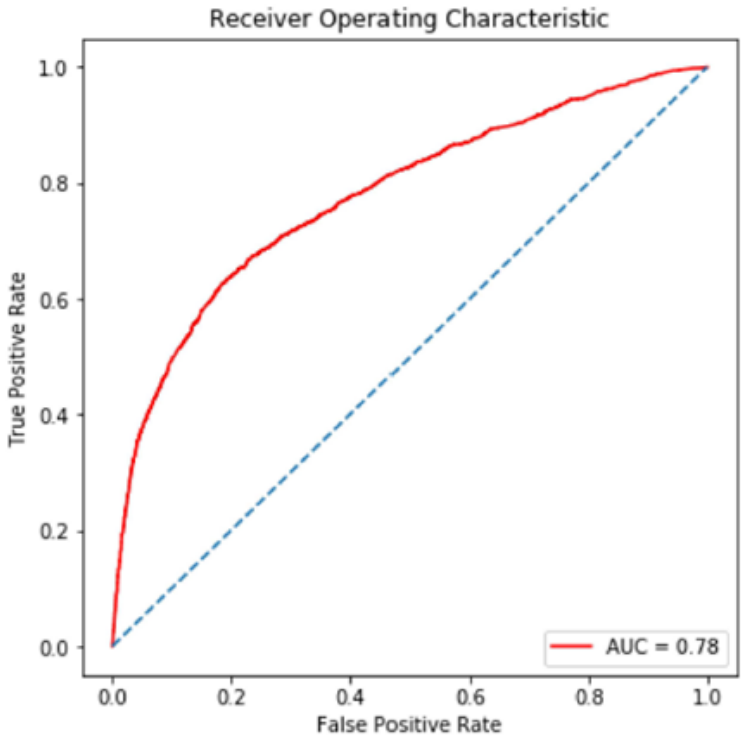


Default of Credit Card Clients

The models performed had similar performances on original and SMOTE data, but gave poor results on downsampled data. Logistic regression got an accuracy of 0.69 on original data, 0.68 on SMOTE data, and 0.53 on downsampled data. The f1 score for both majority (0.78) and minority (0.47) class was same for original and SMOTE data, and the AUC observed on this was 0.73. Decision Tree got a higher accuracy on original data. The accuracy was 0.83, and the f1 score for both classes (0.90, 0.47) were better than that of logistic regression. It performed really poorly on downsampled data with only 0.23 accuracy. The AUC score was 0.75. Random Forest Classifier also got better accuracy on original data, i.e. 0.83. However, the f1 score for minority class was better with SMOTE data (0.88, 0.52) than with original data (0.90, 0.48). The AUC score was 0.77. SVM also gave better accuracy on original data (0.82). The f1 score for minority class was, however, better with SMOTE data. The AUC score, though, reduced to 0.64 compared to previous models. Naïve Bayes performed the worst out of all models with only 0.58 accuracy. The f1 score of 0.66 on majority class and 0.44 on minority class with original data was a lot better than that on SMOTE and downsampled data. The AUC score was 0.74. Last is XGBoost Classifier which performed better on original data compared to other datasets. The accuracy obtained was 0.83. The f1 score for majority class was better on original data, but that for minority class was better with SMOTE data. The AUC score was the highest of all the models, i.e. 0.78.

A trend we can see in these results is that SMOTE data had a relatively higher f1 score on minority class but a lower one on majority class, compared to the original data. This shows that there is a minority-majority class prediction tradeoff with SMOTE data. However, as SMOTE data always gave lower accuracies, we will give results on original data more weightage. It was also observed that all models performed very poorly on the downsmpled data. This means that downsampling led to crucial information being lost.

In summary, models' performance in terms of **accuracy (highest to lowest):** XGBoost/Random Forest/Decision Trees, SVM, Logistic Regression, Naïve Bayes; **AUC score (highest to lowest):** XGBoost, Random Forest, Decision Trees, Naïve Bayes, Logistic Regression, SVM. Moreover, XGBoost had a high f1 score on both majority class (0.90) and minority class (0.47). Even though Random Forest Classifier had a slightly higher f1 score on minority class of 0.48, but XGBoost outperformed it in all other ways. Thus, XGBoost Classifier outperformed all the models in every aspect, and is the most suitable model for loan default prediction in this case too. The ROC curve for this is shown below.



We can clearly see that the results achieved on 'Default of Credit Card Clients' dataset are comparatively lower to those achieved on the 'Loan Data Set', even though the models applied on both were the same.

Conclusion

This study was conducted to find an algorithm which would predict loan default accurately, and save financial institutions from loaning to defaulting customers and incurring losses. Two datasets were used, one large and one small, so an all-rounder model could be achieved. Six models were tested on both the datasets: Logistic Regression, Decision Trees, Random Forest Classifier, Support Vector Machine (SVM), Naïve Bayes, and XGBoost Classifier. Although Random Forrest Classifier came in a close second, the new classifier which has made for itself a reputation in winning Kaggle competitions has proved itself once again. The XGBoost Classifier proved itself to be the most optimal classifier in predicting if a loan would default or not on both large and small datasets. Its results surpassed the other classifiers in all aspects, be it accuracy or AUC score. Moreover, it is seen that results achieved on the smaller data were far better than that reached on the larger dataset. For example, the highest AUC score on 'loan data set' was 0.89, whereas that on 'default of credit card clients' was only 0.78. This can be due to the fact that there were many discrepancies in the 'default of credit card clients' dataset, as I quoted in the methodology section. It had errors, values that were not quoted in the data description, and trends which could not be explained logically. We could not remove these values either as they carried important information, and on their removal the model accuracies reduced. On the other hand, the 'loan data set' provided better results even though it contained missing values. This proves the fact that a model is only as good as the database.

Future Work

Work on loan default prediction is not complete yet. Future researchers should strive to build a model which would achieve an accuracy of at least 95 percent, with an f1 score greater than 0.95 for both majority and minority classes. They can also build models that will not only predict if a client will default or not, but will also calculate the probability of default. Only models with such results can accurately predict if a person will default on a loan. Financial institutions will then be fully able to depend on these systems, and rid themselves completely of any future loss.