PREDICTING WHETHER A CUSTOMER WILL DEFAULT ON A LOAN

Group-1

Akhil Shukla - 180057

Aman Mishra - 180071

Ashutosh Verma - 180155

Avi Alok - 180164

Ayush Mishra - 180176

Problem Description

With the increase in online and offline transactions, more and more customers are coming to banks for loans. A significant number of those customers fail to pay back the loan, leading to losses for the bank. So, if the bank already knew that a customer is likely to default on a loan, it will reject the customer's loan application and save itself from future losses.

Our task is to build a Machine Learning model, which can help banks predict these customers who are likely to default. We will be given 12 attributes related to the customer, and we have to predict whether the customer will default or not. We are also given historical data related to the customers along with the fact whether they defaulted or not. Using the historical dataset, we need to create Machine Learning models and choose the best one to solve the defaulting problem for the bank.

Data Understanding

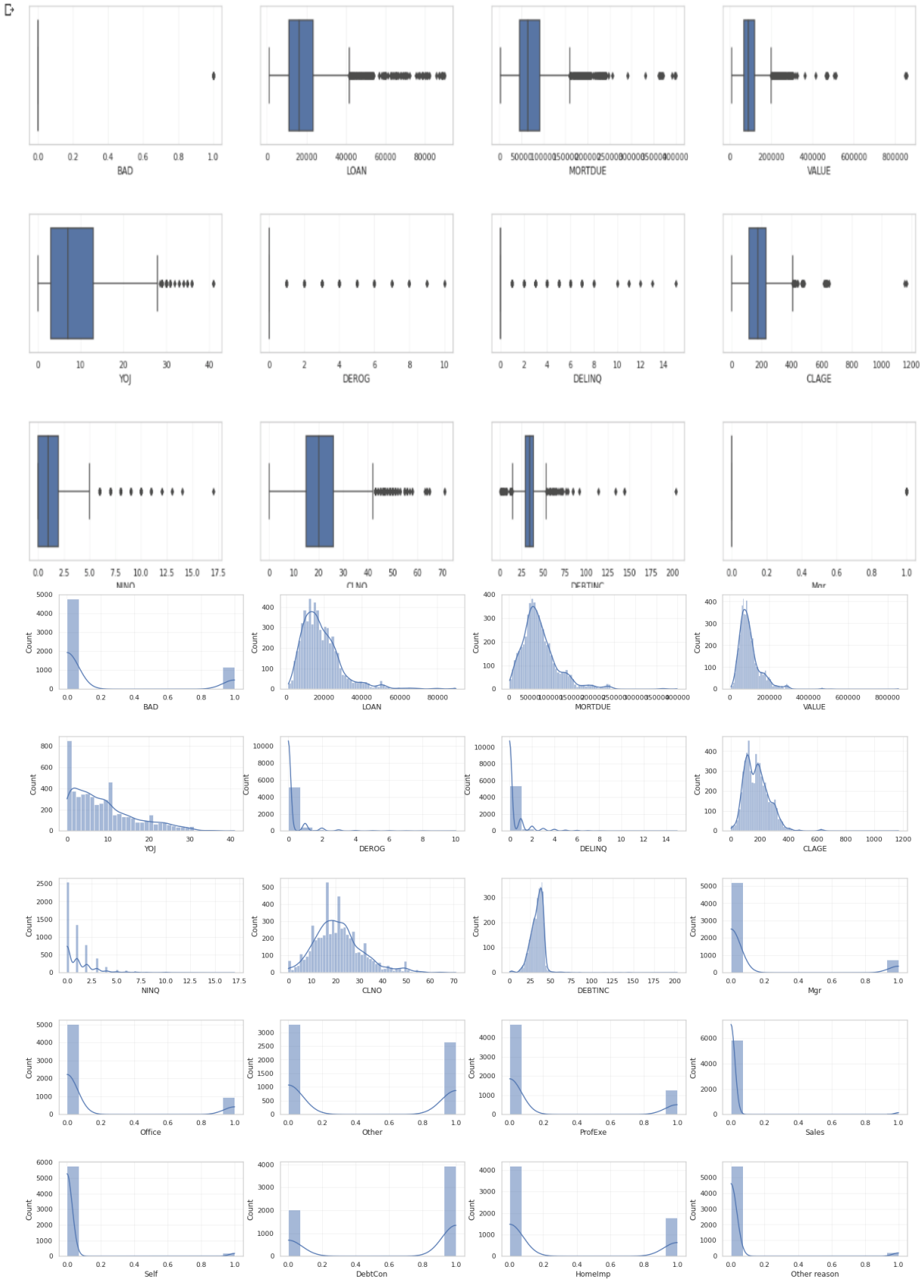
The given historical dataset has the following for every 5960 customers :

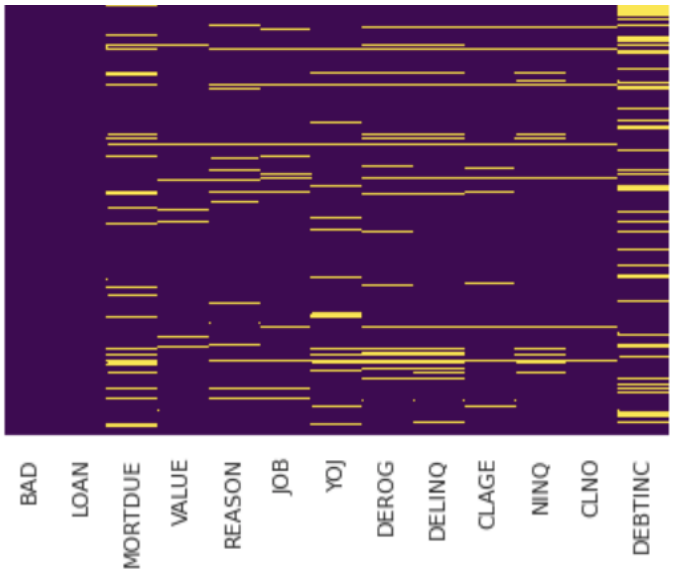
- BAD: 1 = applicant defaulted on loan or seriously delinquent; 0= applicant paid loan (Target Variable){Assymetric Binnary Nominal}
- LOAN: Amount of the loan request (Ratio-Scaled Numeric)
- MORTDUE: Amount due on existing mortgage (Ratio-Scaled Numeric)
- VALUE: Value of current property (Ratio-Scaled Numeric)
- REASON: DebtCon = debt consolidation; Homelmp home improvement (Nominal)
- JOB: Occupational categories (Nominal)
- YOJ: Years at present job (numeric discrete)
- DEROG: Number of major derogatory reports (numeric discrete)
- DELINQ: Number of delinquent credit lines (numeric discrete)
- CLAGE: Age of oldest credit line in months (numeric discrete)
- NINQ: Number of recent credit inquiries (numeric discrete)
- CLNO: Number of credit lines (numeric discrete)
- DEBTINC: Debt-to-income ratio (Ratio-Scaled Numeric)

The dataset has three categorical attributes, *REASON*, *JOB* and *BAD*.

Data Visualization:

- Plotted histogram and boxplot, heatmaps, pair plots with hue to understand the distribution and number of outliers, missing values and a general correlation between the variables so we don't miss out on any obvious details about the data


BAD
LOAN
MORTDUE
VALUE
YOJ
DEROG
DELINQ
CLAGE
NINQ
CLNO
DEBTINC
Mgr
Count
Office
Other
ProfExe
Sales
Self
DebtCon
HomeImp
Other reason



There are no anomalies present in the data and the distributions are fine. The feature YOJ is highly skewed and may be modified to decrease skewness. JOB and REASON must be one hot encoded such that we can use them for logistic regression model. DELINQ, DEROG may be divided into 2 classes to create new binary variables.

Observations from the above plots:

- The **scale of each attribute is different**. We need to normalize all the features.
- Some attributes have **skewed distribution**
- Some attributes have a lot of **outliers** (DEBTINC, LOAN, MORTDUE, VALUE)

Attributes like LOAN, MORTDUE, and VALUE have many outliers (from boxplot), so we will also try Z-Score Normalization (Standard Scaler).
For fixing the skewness, we need to transform the attributes.

- We also plotted the Quantile plot to compare their distribution with the normal distribution, and the continuous attributes were not normally distributed. We concluded that data transformation needs to be applied to make them normally distributed.
- Plotted the heatmap of the correlation matrix to understand the type of linear relation between attributes. Features/attributes which were highly correlated could be dropped to make the ML model simpler, as they essentially provide almost similar information to the model. We used a threshold of 0.5 to decide if we wanted to remove the variables or not.
- Plotted the heatmap of Predictive Power Score to determine non-linear relationships between attributes. From the heatmap, we inferred potentially important features.
- Plotted heatmap of missing value to visualize missing values in each attribute as well as in the tuples/rows.
- Performed Chi-square analysis to identify potentially important feature, which might be useful in the further Data Cleaning and Transformation

Data Preprocessing

Following steps were performed before the visualization step to ease data understanding

- Replacing missing value of REASON column with *Other Reason* and the JOB, DEROG and DELINQ columns with the *mode()*
- One-hot-encoding (get_dummies) the REASON and JOB, as these are nominal attributes and the visualization libraries and ML models require numerical input.

The following were performed after understanding the dataset:

- **Filling Missing Value** using inbuilt interpolate function in pandas if some missing values are still left behind, then filling them with the mode of that attribute. Interpolate fills the missing values with the average of the numbers just above and below the missing value.
- **Numerosity Reduction**: Many tuple/observations had many missing values in their attributes. We can consider dropping them to improve the data quality. We decided on four as the threshold value so that the overall data quality improves and much data is not lost. DEBTINC had the most missing values in the data and at first it seems it's better to remove the whole column but on later inspection it turns out it's a very important feature.
- **Feature Reduction** - Dropping columns with the same value for most observations (*DELINQ* and *DEROG*) after considering their Correlation and Predictive Power Score (values from *REASON* and *JOB*). We applied PCA for numerical continuous variables. The explained variances were fairly distributed hence the application of PCA in feature reduction was not much useful as the original features are good enough to make machine learning models. Along with this chi-square test was used to reduce the number of nominal features, but again, the best features according to the chi-square test only improved the efficiency very minutely.
- **Feature Modelling** – Added new feature MORTDUE/DEBTINC.
- **Feature Transformation** – after observing the Quantile plot from the Visualisation section, we tried different transformations like taking the log, power transformations (simply raising power, yeo-johnson and box-cox) to make the current distribution close to a normal distribution as possible. Following transformation were finalised as the best:
    - yeo-johnson for LOAN
    - raising to the power 0.125 for MORTDUE
    - taking log(value + 10) for YOJ, VALUE, CLNO
- As the features have different scales, we needed to use scaling. After comparing the result with Min-Max Scaler, Standard Scaler and Robust Scaler, Robust was performing better
- Data was skewed in many variables. In skewed data, the tail region may act as an outlier for the statistical model, and we know that outliers adversely affect the model's performance especially regression-based models. There are the statistical model that is robust to outliers like a Tree-based models, but it will limit the possibility to try other models. So there is a necessity to transform the skewed data to close enough to a Gaussian distribution or Normal distribution. This will allow us to try more statistical models.
- Removal of outliers was also done using the IQR range, but the original data was skewed in many features, and hence just removal of them would reduce the accuracy of models. Feature transformation using the log to change the distribution was a better way to treat these outliers rather than just blatantly removing them.
- In preprocessing there were many steps that can be taken differently so we created different datasets. For example while filling missing values several different techniques were applicable like filling the data with mode ,mean etc.

Model Building and Analysis

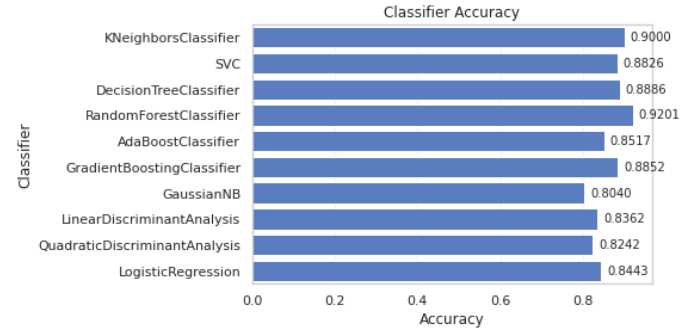
We tested the dataset on several machine learning models. They are listed below. To accommodate this, we did several different kinds of transformation and preprocessing.

1. Logistic Regression
2. Decision Tree
3. Random Forest Classifier
4. Support Vector Machine Classifier
5. Gradient Boosting Classifier
6. Extra Trees Classifier
7. Adaptive Boosting Classifier
8. Linear Discrimination Analysis
9. Quadratic Discrimination Analysis
10. K-Neighbors Classifier
11. Gaussian NB

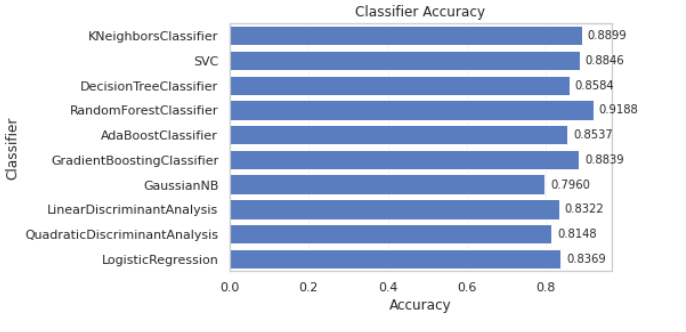
Our objective is to minimize company loss, predicting the risk of client default. A good recall rate is desirable because we want to identify the maximum number of clients that are indeed prone to stop paying their debts. Thus, we are pursuing a small number of False Negatives. We created different datasets with different types of transformations and preprocessing techniques. The preprocessing techniques deferred in the way the outliers were replaced and the scaling done for various features.

For various datasets, we created a summary of the accuracy of different ML models. This was done so we could identify the best preprocessing and transformation. After this, an in-depth analysis was pursued of the models with the best scores. As evident from the graph in the Decision tree classifier and other classifiers, the accuracy does not approve by much by doing preprocessing as these models themselves do some feature selection. However, as evident from the graph, the preprocessing and transformation have improved the accuracy in every case.

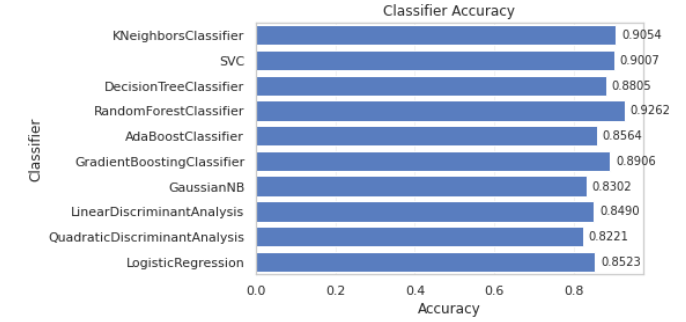
For dataset df: original with one added feature (PROBINC)



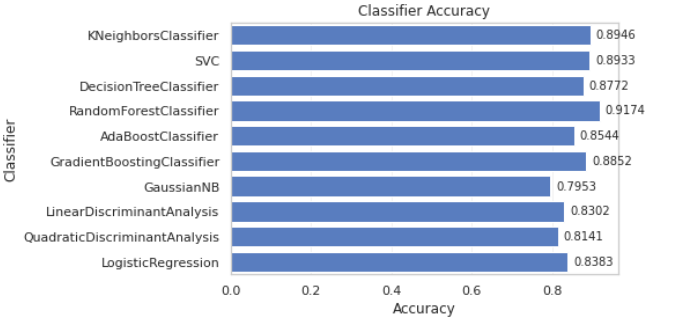
For dataset df_1: transformed just LOAN feature



For dataset df_2: transformed LOAN and other features too, and dropped other features (MORTDUE, YOJ), which seemed unimportant from the visualisation section



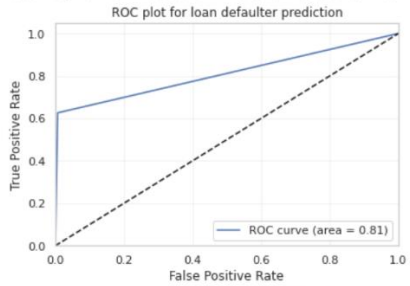
For dataset df_3: contains transformed features, LOAN and others (MORTDUE, YOJ, etc.)

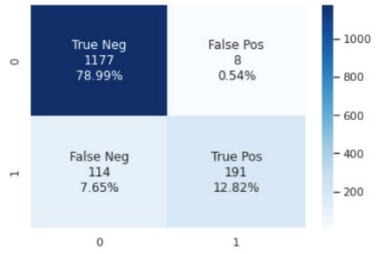


We inferred that df_2 (with all the transformations) is slightly better than other datasets from the above plots. The problem converged to finding out the best models for this dataset.

A summary of models trained on df_2 :

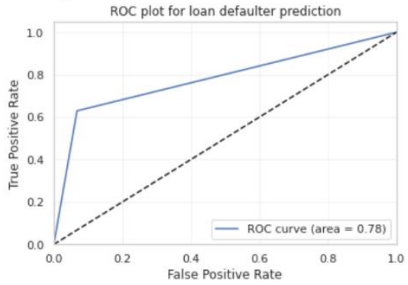
RandomForestClassifier
Accuracy on Training Dataset : 1.0
Accuracy on CV Dataset : 0.9181208053691275

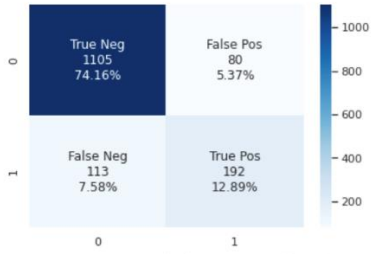




	precision	recall	f1-score	support
0	0.81	0.87	0.84	1212
1	0.18	0.13	0.15	278
accuracy			0.73	1490
macro avg	0.50	0.50	0.49	1490
weighted avg	0.69	0.73	0.71	1490

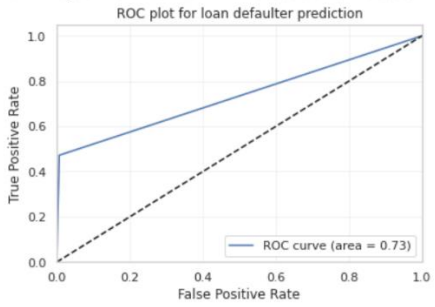
DecisionTreeClassifier
Accuracy on Training Dataset : 1.0
Accuracy on CV Dataset : 0.8704697986577181

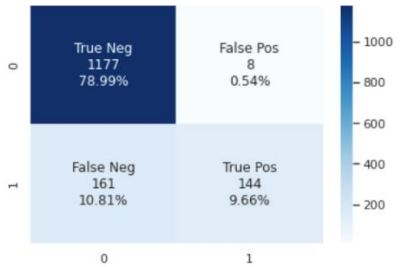




	precision	recall	f1-score	support
0	0.81	0.81	0.81	1212
1	0.16	0.16	0.16	278
accuracy			0.69	1490
macro avg	0.48	0.49	0.48	1490
weighted avg	0.69	0.69	0.69	1490

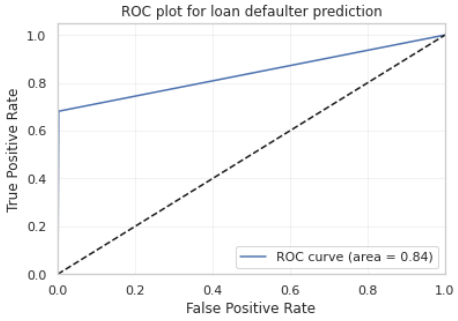
SVC
Accuracy on Training Dataset : 0.9029082774049217
Accuracy on CV Dataset : 0.8865771812080537

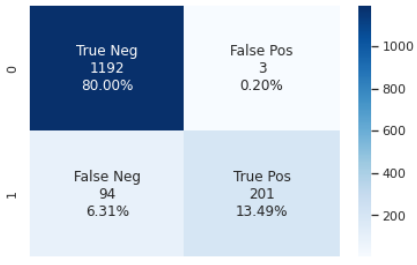




	precision	recall	f1-score	support
0	0.81	0.90	0.85	1212
1	0.17	0.09	0.12	278
accuracy			0.75	1490
macro avg	0.49	0.49	0.49	1490
weighted avg	0.69	0.75	0.72	1490

ExtraTreesClassifier
Accuracy on Training Dataset : 1.0
Accuracy on CV Dataset : 0.9348993288590604





	precision	recall	f1-score	support
0	0.80	0.87	0.83	1191
1	0.22	0.15	0.18	299
accuracy			0.72	1490
macro avg	0.51	0.51	0.51	1490
weighted avg	0.69	0.72	0.70	1490

Result Interpretation

SVC has the best recall rate from the detailed analysis, but Extra Trees have the least count of False Negative. Random Forest and Extra Trees have the highest accuracy. Though SVC has good recall, its count of False Negative is large relative to the others, and its area under the ROC curve is close to 0.5 (relative to others).

Rest three have the almost same count of False Negatives. Decision Trees have low recall and accuracy compare to the Random Forest and Extra Trees.

Thus, Random Forest and Extra Trees Classifier are the best choices for the problem. They have high accuracy, good recall, and fewer False Negatives than the other models.