ML Project Presentation

Credit Score Classification

Group - 29

Aman Kumar -2020279

Karan Prasad Gupta - 2020439

Pritish Poswal - 2020321

Vibhu Jain - 2020151

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

Motivation

*"The world is one big data problem." ~**Andrew McAfee.***
Solve big problems using big data

We wanted to select something which had significance in Today's world.

Credit cards have become an integral part of our lives and a huge fraction of young and medium age people use it.
The usage of credit cards has increased over the years and with the emergence of companies like Cred there is more incentive for people to use credit cards.

Classification of credit score is important because credit score acts as a feedback to validate the users, thus a good credit score can be very beneficial as it helps the user to get more favourable loans, credit cards and more. In modern times even if we don't generate the exact credit score but instead just give a rough classification of the category then it would be pretty useful for the banks and lenders.

Image: Kotzias et al. from ZDnet

Introduction

Credit score classification is a complex problem as it depends upon many parameters and factors. Classifying credit score using traditional data analysis techniques or manual classification by a human being is a tedious and time consuming task. The process would be highly inefficient so hence we picked machine learning to solve this problem.

Income, No of loans, No of delayed payments,Payment history, debt-to-credit ratio, length of credit history, new credit, and the amount of credit are some of the factors which influence classification of credit score

Credit Score

Good   bad   standard

We decided not the use occupation as a parameter as we want to help people with undervalued occupations to get loans if they are eligible and we didn’t wanted occupation to impact classification of credit score in any way.

Literature Review

➢ **Paper1**

- Paper titled “Credit Risk Scoring Analysis” by iyue Qiu, Yuming Li, Pin Ni and Gangmin Li.
- reports out efforts in using feature engineering and machine learning models for credit Score modeling and reporting their AUC scores as false positive and false negative rates are a important issue for credit score classification.
- Used 4 Datasets : where first is original Dataset and other three were constructed datasets.The three constructed datasets were polynomial generated dataset, expert knowledge generated dataset and feature tools toolkit generated dataset/
- Pre-processing steps
  (1) Anomalies and contradiction detection (2) Missing Data Imputation (3) Nominal Data Pre-processing (4) Data Integration (5) Feature Selection (6) Feature Construction.
- Models used :
  (1) Logistic Regression
  (2) Random Forest
  (3) Light GBM [Light Gradient Boosting Machine]
- Best result on original Dataset : Light BGM - 72.1%
- The shortcoming here is that only limited models have been tried and generation of 3 new datasets have not been properly documented and explained.

Literature Review

➢ **Paper2**

- Paper titled "**Credit scoring using machine learning algorithms**" by Evander E.T. Nyoni1 , Ntandoyenkosi Matshisela2
- The paper talks about the problem of non performing loans in recent years for which improper classification of credit score is responsible
- used the AUROC approach to make analysis of machine learning methods of classification.
- They have Used 10 fold- Cross validation on the German Credit Data Set .
- The goal of this paper was to develop and evaluate the classification machine learning techniques.
- models used : (1)Random Forests, (2) Lasso regression , (3) Support Vector Machine, (4) Logistic regression.
- The result of this paper was that the Lasso Regression model was having an accuracy of 80.48% which implies that Regression is a good model in classifying the credit score in that dataset. In the end the conclusion was using machine learning techniques with such high accuracy millions of dollar of credit default can be avoided
- The limitations of this paper is first of all the dataset used has very less data also in the paper they have not given much details about preprocessing and feature selection done.

Dataset Description

➢ **Details:**

- Dataset has been taken from Kaggle.
- Data contains the 1,00,000 entries
- Total 28 features in the dataset shown in the image here
- Contains : Junk , Empty values
- Some Integers/Float fields columns were also of type String .
- **Our target variable column - > Credit_Score**

Good Standard Poor

Fig showing Distribution of Target Class

1	ID	string
2	Customer_ID	string
3	Month	string
4	Name	string
5	Age	string
6	SSN	string
7	Occupation	string
8	Annual_Income	string
9	Monthly_Inhand_Salary	float64
10	Num_Bank_Accounts	int64
11	Num_Credit_Card	int64
12	Interest_Rate int64	int64
13	Num_of_Loan	string
14	Type_of_Loan	string
15	Delay_from_due_date	string
16	Num_of_Delayed_Payment	string
17	Changed_Credit_Limit	string
18	Num_Credit_Inquiries	float64
19	Credit_Mix	string
20	Outstanding_Debt	string
21	Credit_Utilization_Ratio	float64
22	Credit_History_Age	string
23	Payment_of_Min_Amount	string
24	Total_EMI_per_month	float64
25	Amount_invested_monthly	string
26	Payment_Behaviour	string
27	Monthly_Balance	string
28	Credit_Score	string

Dataset Cleaning

- **Handling Null values :**

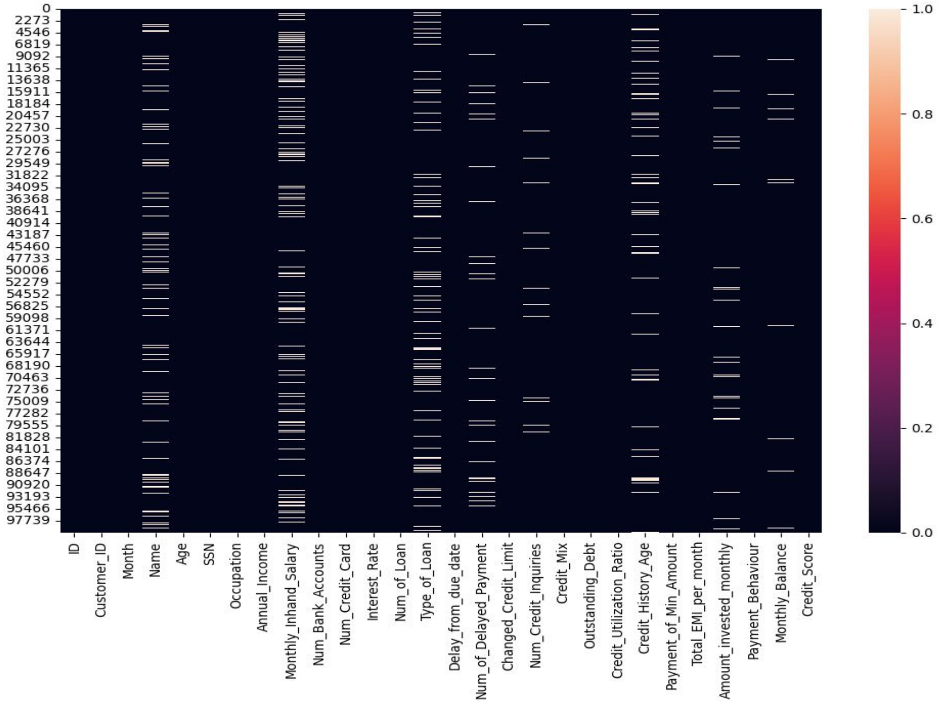


Fig Showing Null values in Each Columns

- **Monthly_InHand_Salary** : Used annual_Salary to calculate it .
- **Type_of_Loan** : Used 'Not_Specified'
- **Amount_invested_monthly, monthly balance :** Replaced by mean of the respective fields .
- **Num_of_delayed_payments , Num_credit_enquires :** Were replaced with 0 .
- **Credit_history_Age :** We removed the records with null values .

Dataset Cleaning

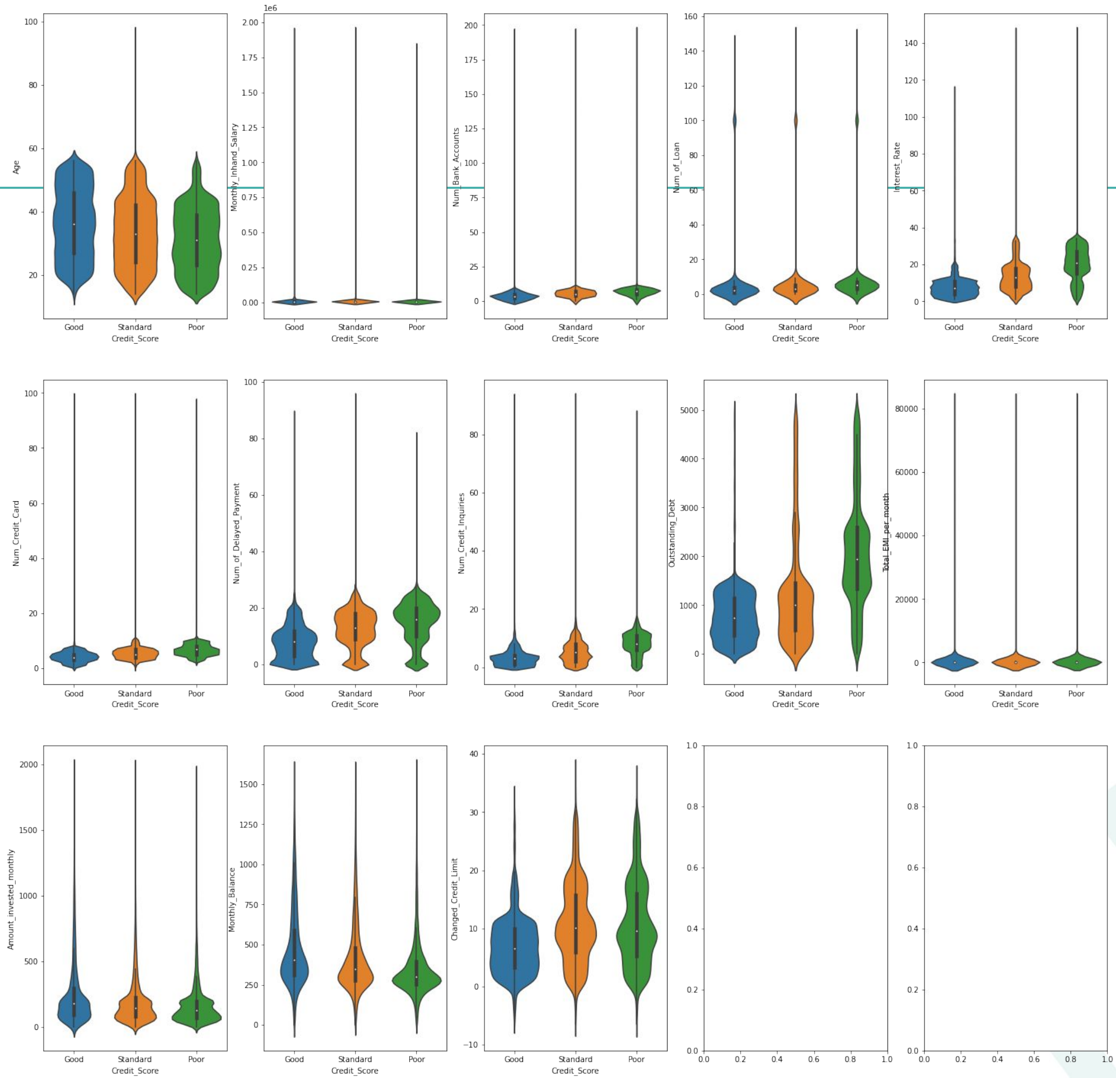
- Removing Redundant columns [Columns Dropped - **ID , Customer_ID, Name , SSN**]
- Handling Junk Values :
    - Some Entries like '_" or '__10000__', "__-333333333333333333333333333__"were present in dataset.
        - Removed some records where it seems to be irrelevant.
        - Changed some records which seems to be a mistake by just removing that extra Character .
        - Finally then type casting to Integer/Float .

- Handling Negative Records :
    - Removed the Irrelevant Junk values .
    - Took absolute values of records which needs to be in positive .

EDA

Violin plots of all
Non string parameters

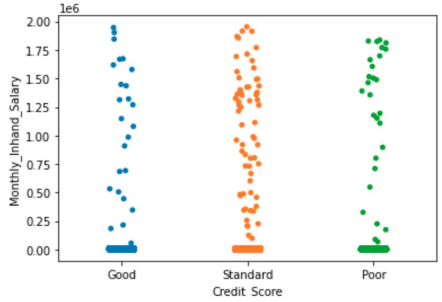
It helps us to visualize the density and distribution of of dimensional data

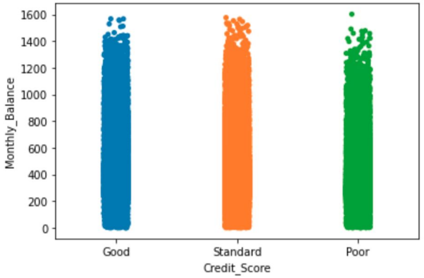


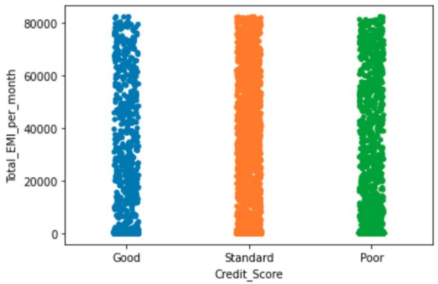
Age

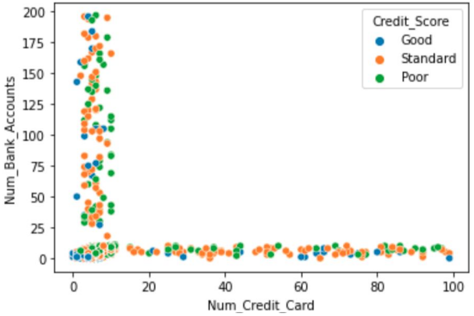
Outstanding debt

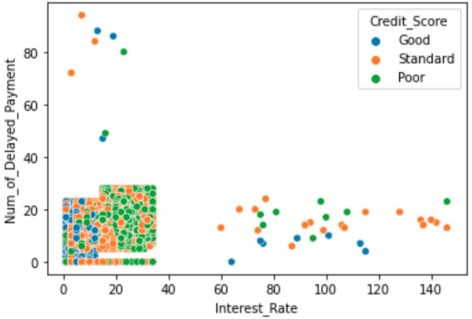
EDA





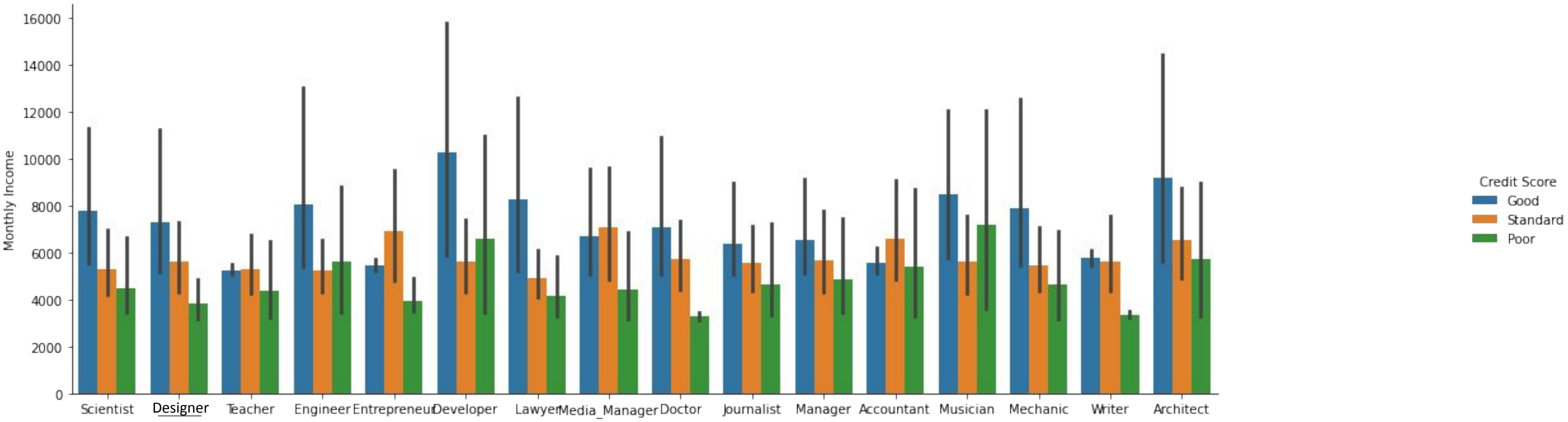






- **Stripplots helps us visualize distribution of one dimensional data**
- **Scatterplots help to visualize how one factor affects the other**

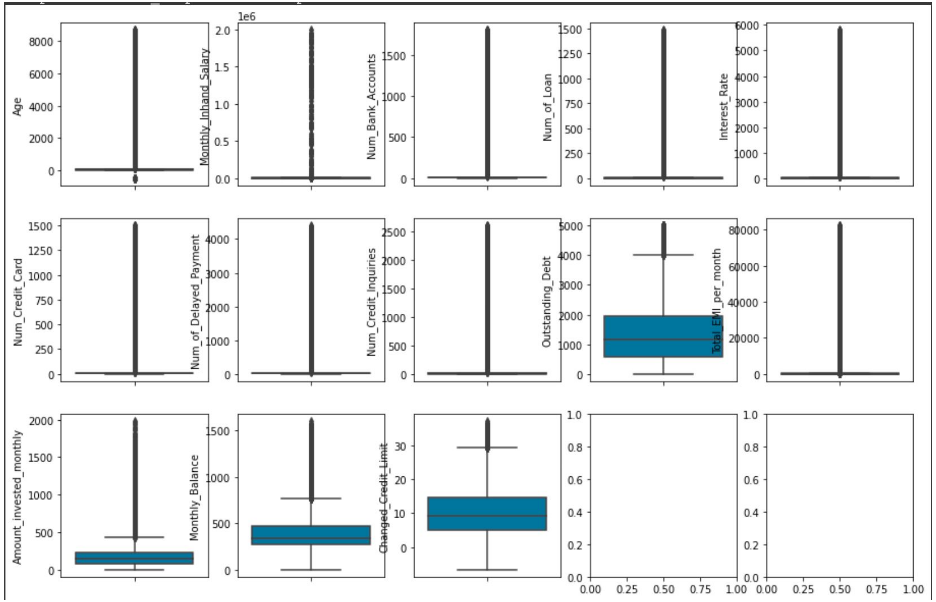
EDA

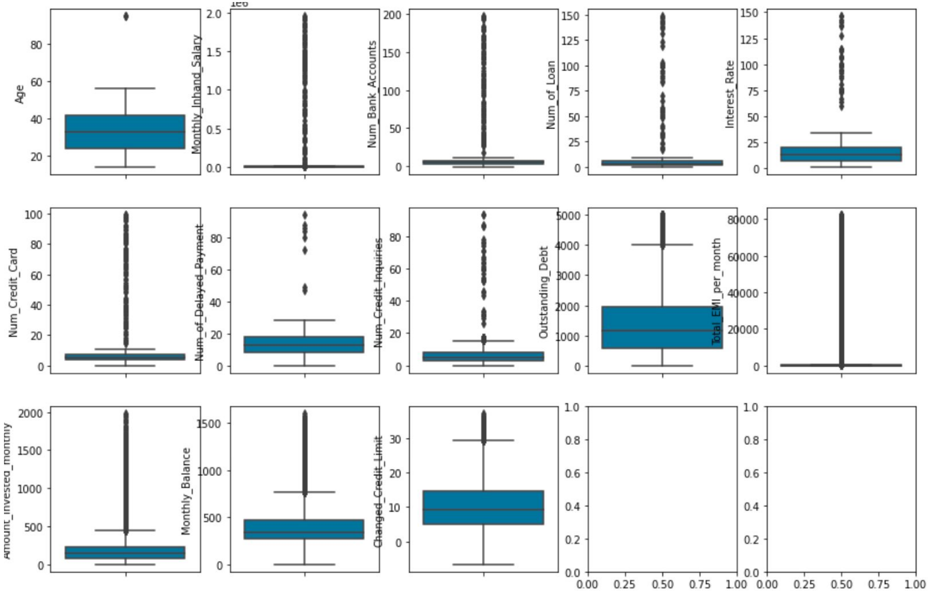


In Figure - Monthly Income, Occupation and Credit score relation using catplot

Developers have highest income among good credit score individuals
Doctors have lowest income among poor credit score individuals

Preprocessing – Removing Outliers





Removing outliers using Box plots to improve accuracy of various models which are trained.

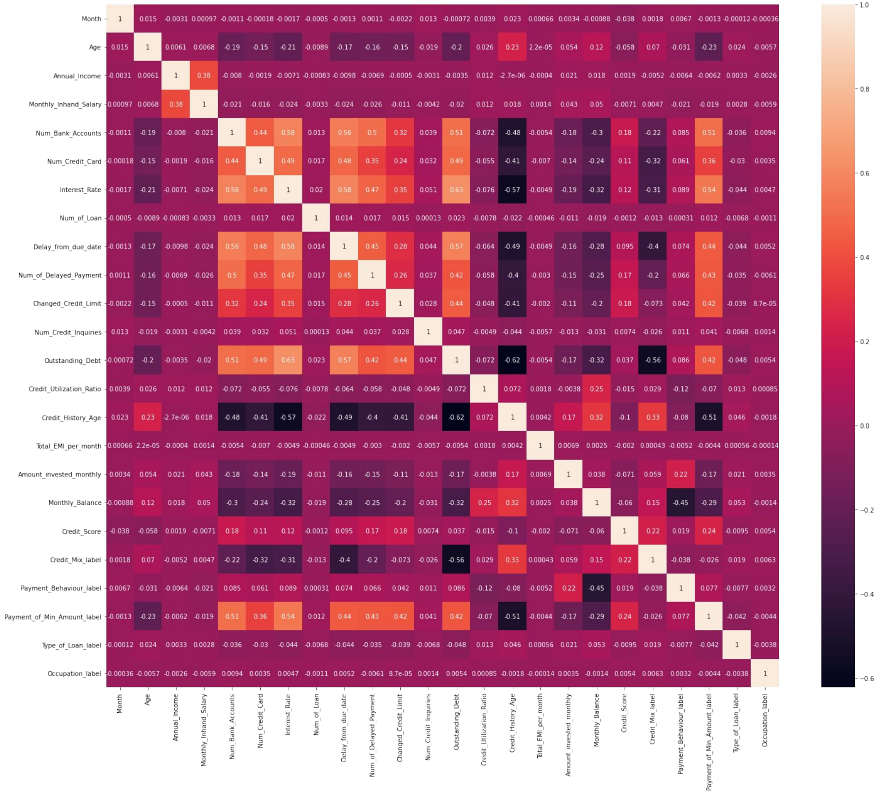
Preprocessing –Encoding string fields

N
Type_of_Loan
Auto Loan, Credit-Builder Loan, Personal Loan, and
Auto Loan, Credit-Builder Loan, Personal Loan, and
Auto Loan, Credit-Builder Loan, Personal Loan, and
Auto Loan, Credit-Builder Loan, Personal Loan, and
Auto Loan, Credit-Builder Loan, Personal Loan, and
Auto Loan, Credit-Builder Loan, Personal Loan, and
Auto Loan, Credit-Builder Loan, Personal Loan, and
Auto Loan, Credit-Builder Loan, Personal Loan, and
Credit-Builder Loan
Credit-Builder Loan
Credit-Builder Loan
Credit-Builder Loan
Credit-Builder Loan
Credit-Builder Loan
Credit-Builder Loan
Credit-Builder Loan
Auto Loan, Auto Loan, and Not Specified
Auto Loan, Auto Loan, and Not Specified
Auto Loan, Auto Loan, and Not Specified
Auto Loan, Auto Loan, and Not Specified
Auto Loan, Auto Loan, and Not Specified
Auto Loan, Auto Loan, and Not Specified
Auto Loan, Auto Loan, and Not Specified
Auto Loan, Auto Loan, and Not Specified
Not Specified
Not Specified
Not Specified

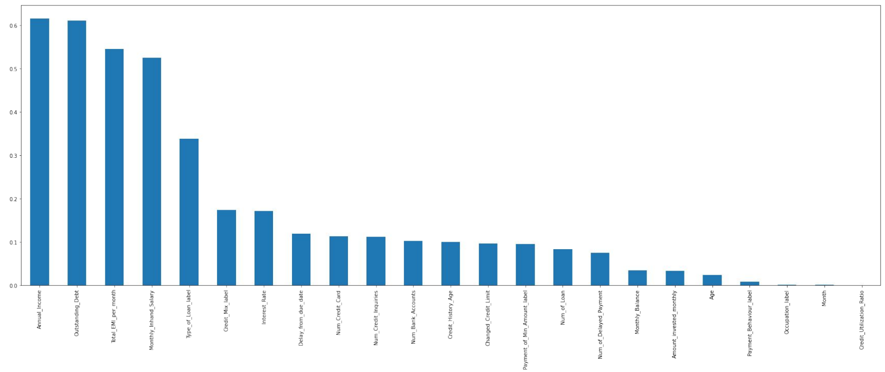
Type_of_Loan_label
128
128
128
128
128
684
684
684
684
684
684
684
684
63
63
63
63
63
63
3463
3463
3463

- We used LabelEncoder from sklearn to encode string fields
- We also used power transform for Normalizing the data which helped us get better results as it makes the data more Gaussian like.

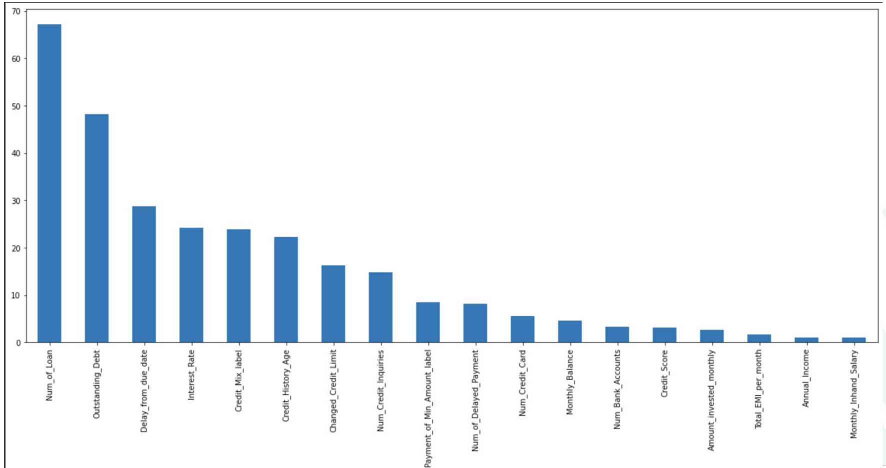
Feature Selection



Heatmap : helps us to find the correlated features



Information Gain and Fischer Scores were used to remove irrelevant features



Details of the Preprocessed Data used for Model training

- After removing outliers, handling null values, removing some features. We obtained the data with following specifications

- Data used finally has 71876 rows and total of 18 columns

- We removed the fields -

  ID, Name, SSN, Customer ID, Credit_Utilization_ratio, Occupation, Month, Payment_Behaviour, Age

```
RangeIndex: 71876 entries, 0 to 71875
Data columns (total 19 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   Annual_Income               71876 non-null  float64
 1   Monthly_Inhand_Salary       71876 non-null  float64
 2   Num_Bank_Accounts           71876 non-null  int64  
 3   Num_Credit_Card             71876 non-null  int64  
 4   Interest_Rate               71876 non-null  int64  
 5   Num_of_Loan                 71876 non-null  float64
 6   Delay_from_due_date         71876 non-null  int64  
 7   Num_of_Delayed_Payment      71876 non-null  float64
 8   Changed_Credit_Limit        71876 non-null  float64
 9   Num_Credit_Inquiries        71876 non-null  float64
 10  Outstanding_Debt            71876 non-null  float64
 11  Credit_History_Age          71876 non-null  float64
 12  Total_EMI_per_month         71876 non-null  float64
 13  Amount_invested_monthly     71876 non-null  float64
 14  Monthly_Balance             71876 non-null  float64
 15  Credit_Score                71876 non-null  int64  
 16  Credit_Mix_label            71876 non-null  int64  
 17  Payment_of_Min_Amount_label 71876 non-null  int64  
 18  Type_of_Loan_label          71876 non-null  int64  
dtypes: float64(11), int64(8)
```

Methodology

We Applied various models like

1)Logistic Regression,
2)Gaussian Naive Bayes,
3)Decision tree with gini index criterions
4)Decision tree with gini index entropy criterions
5)Random forest classifier
6)KNN
7)MLP
8)MLP Bagging
9)XGB
10)Extra Tree Classifier
11)SVM
12)Stack (RF+KNN)
13)Stack (RF+KNN+XGB)

After which we measured the accuracy, Precision, Recall and F1 score for each. Along with it with we also plotted ROC-AUC curves for each model to better understand and analyze the results.

For each model we tried various combinations of hyperparameters and also used methods like GridSearch to find the optimal parameters to find the best accuracy for our project.

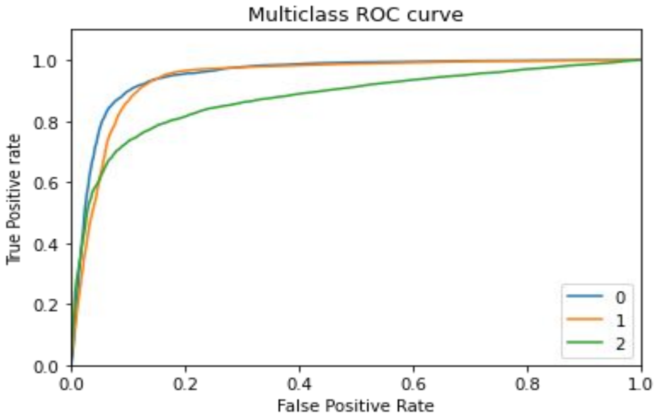
Random Forest

Accuracy Achieved : 81.14%

Recall : 0.80

Precision : 0.80

F1 Score : 0.80



0 - Good
1- Poor
2- Standard

Parameters: max_depth:None, n_iterations:200

, criterion: ‘gini’

We Used multiple Grid search to fine

tune the hyperparameters

```
param_grid = {
        'n_estimators' :[80,100,110,130],
        'max_depth':[10,15,20,None],
        'criterion':['gini','entropy']

  }
```

N_estimators : 130 max_depth: None, criterion:gini

```
param_grid = {
        'n_estimators' :[130,160,200,230],
        'max_depth':[None],
        'criterion':['gini','entropy']

}
```

N_estimators: 200, criterion:’gini’

XGBoost

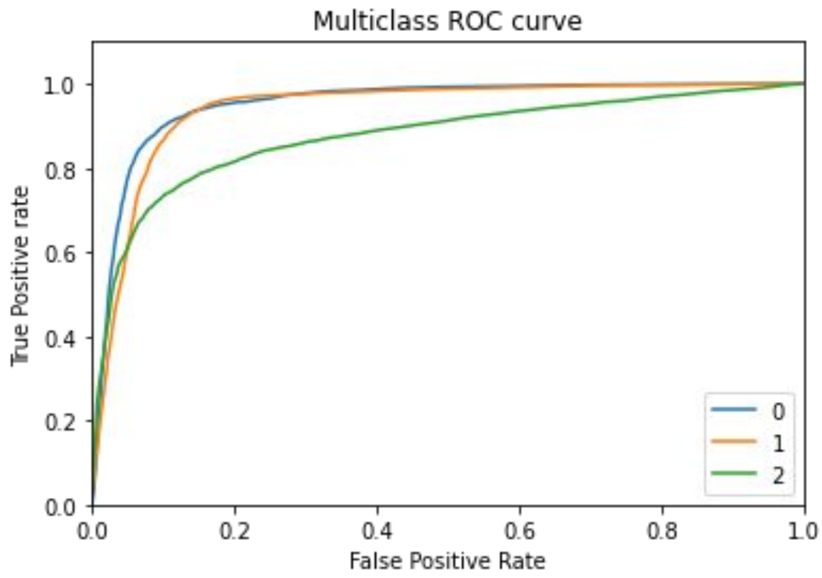
- We tried using Extreme Gradient Boosting Algorithm to see if we could get an alternative to random forest and get a lower cost model.

- GridSearch was performed for improving the model giving n_estimators=500

- Results for this model had high accuracy of 73% but it was not as good as random forest

Stack Classifiers

- Used several models together to build new model with better Performance
- Like combining Random forest , KNN, XGBoost together gave us a high production accuracy of more than 80%
- We also tried stacking other model combinations together like logistic regression and naive bayes and the performance was better than individual models.
- Models used in stacking were first individually optimized by fine-tuning hyperparameters

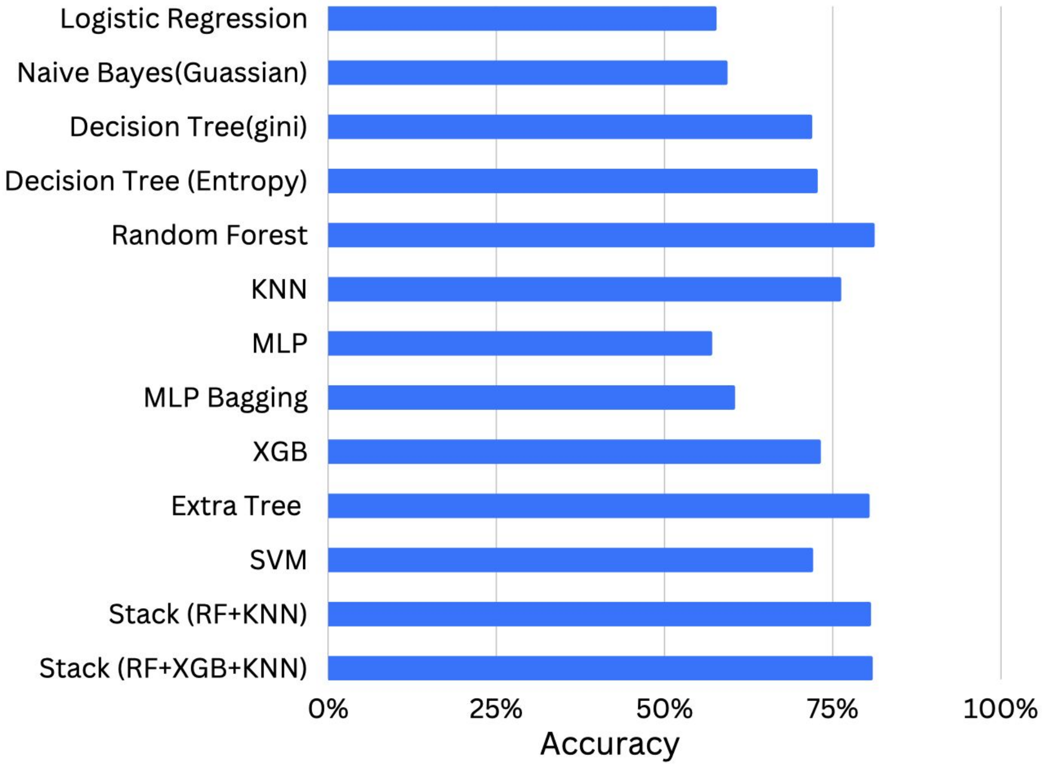


0 - Good
1- Poor
2- Standard

Results and Analysis

- Random forest, KNN, SVM, Decision trees, XGBoost are the best performing models Accuracy b/w 71% to 81%
- Overall performance, Random Forest gives the best accuracy of 81% on the test data.
- Stacking models like KNN, Random forest also gave us highly accurate results

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	57.65 %	0.49	0.43	0.41
Gaussian Naïve Bayes	59.28 %	0.54	0.51	0.52
Decision Tree with Gini	71.86 %	0.69	0.70	0.69
Decision Tree with Entropy	72.68 %	0.70	0.72	0.71
Random Forest	81.14 %	0.80	0.80	0.80
KNN	76.17 %	0.75	0.75	0.75
MLP	57.00 %	0.60	0.38	0.33
MLP bagging	60.40%	0.61	0.62	0.61
XGB (Extreme Gradient Boosting)	73.14 %	0.71	0.72	0.71
Extra Tree Classifier	80.39 %	0.79	0.79	0.79
SVM	72.00 %	0.70	0.71	0.70
StackClassifier (KNN + Random Forest)	80.60 %	0.80	0.79	0.80
StackClassifier (XGB+ KNN + Random Forest)	80.85 %	0.80	0.80	0.80

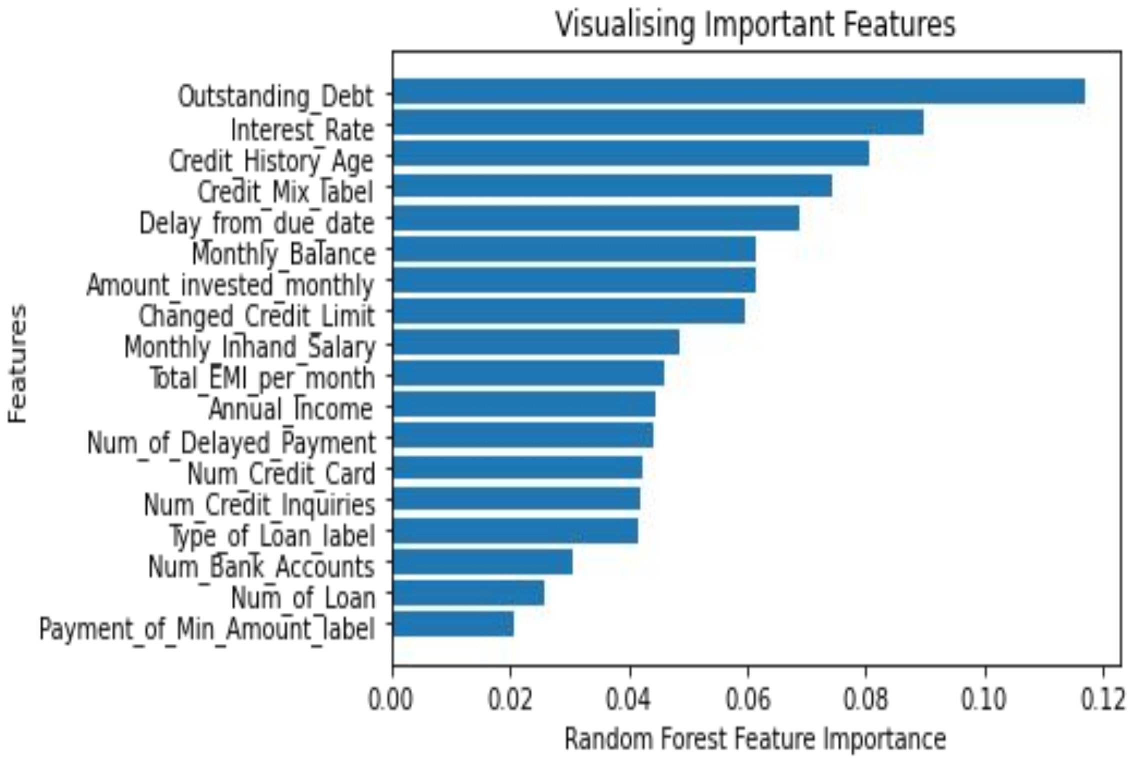


Accuracies of All models

SVM was performed on small set of data still giving good accuracy of 72%

ExtraTreesClassifier and Random forest are similar models
Random forest gives slightly higher accuracy means calculating optimum split point is better than choosing randomly

Results and Analysis



Plot for Features vs importance in RF classification

- Best Model: **Random Forest**
- We found the most important features for Random Forest
- Outstanding debt is the most important feature in classification

Conclusion

Learnings:

The project helped us to get essence of working with large amount of data at once where lot of preprocessing is required. We also learnt how to do EDA, feature selection and finding the optimal hyperparameters for the ML model. We learnt many things which were not part of the course

Future task:

Future scope of the project can include finding and experimenting some unexplored models, working together with small scale financial institutions for helping them to solve problem of lending and we can even use knowledge to work on more challenging problem of predicting the exact credit score ie the CIBIL score.

Timeline and New work done

We were able to follow the proposed timeline given in the project proposal. After the mid sem presentation we did a significant amount of work in improving our model. The work done was more than what was written in future plan.

- We again did data cleaning, feature selection and preprocessing for further improvements in our accuracy
- We applied a large amount of new models like KNN, MLP, MLP bagging, SVM, XGB, Extra tree classifier.
- As mentioned in future plan we combined multiple models together by using stacking
- We fine tuned our hyperparameters to get optimal results

Individual Contributions

Aman Kumar: Preprocessing, Exploratory Data Analysis, Hyperparameter tuning, Data Visualization, Report Writing, Analysis of the performance of the models.

Karan Prasad Gupta: Literature Review, Data Visualisation, Model Training specially stacking of the models, Model Testing, Report Writing

Pritish Poswal: Data Cleaning, Data Preprocessing , Applying ML Models to data, Model Selection, Model Testing, Hyperparameters tuning, Making presentation

Vibhu Jain: Data Preprocessing, Data Visualisation, Exploratory Data Analysis, Literature Review, Feature Selection, Report Writing and making presentation

Thank you !!!