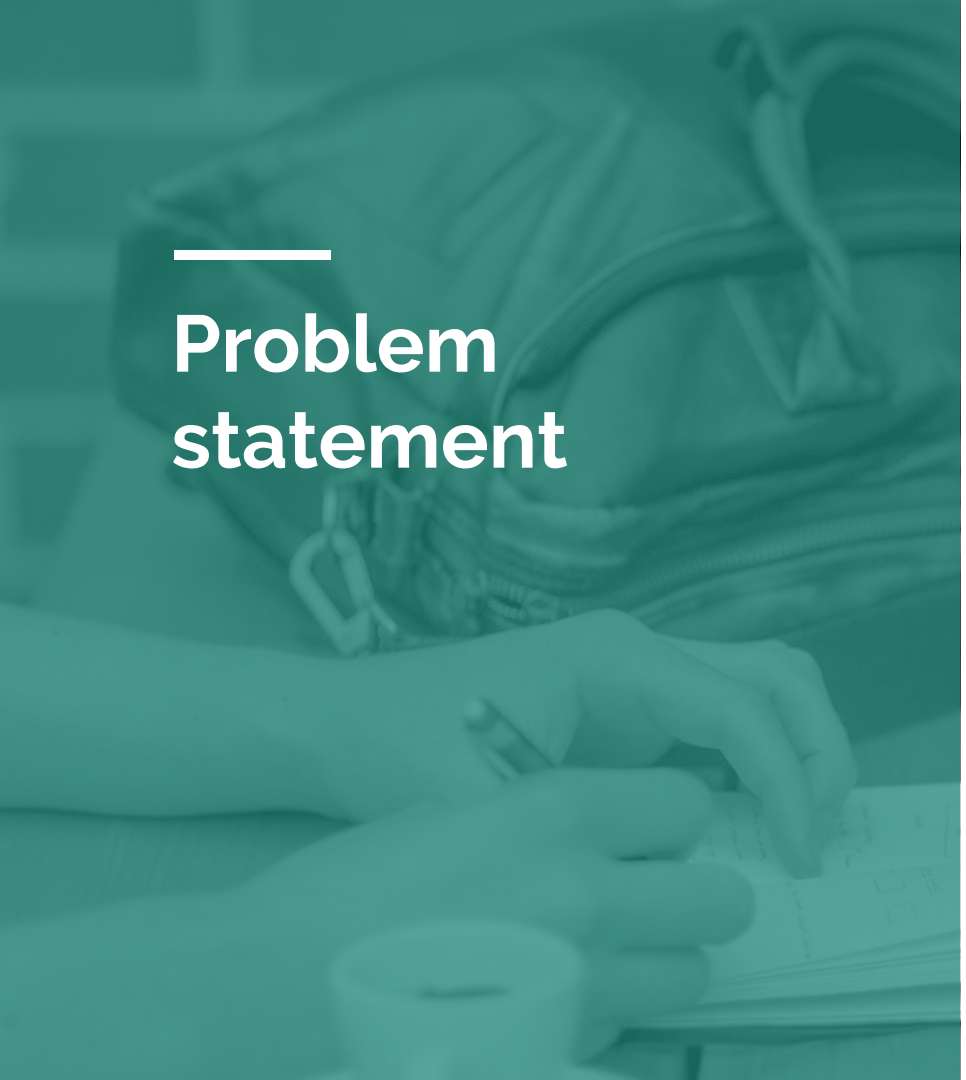
Credit Risk Analysis

Trupti Dhabu
Snehal Hawelikar
Anusha Ganesan

Contents



Problem statement

Business Problem :

To identify the customers, who would be eligible for loan in the future based on the past data.

Analytical goal :

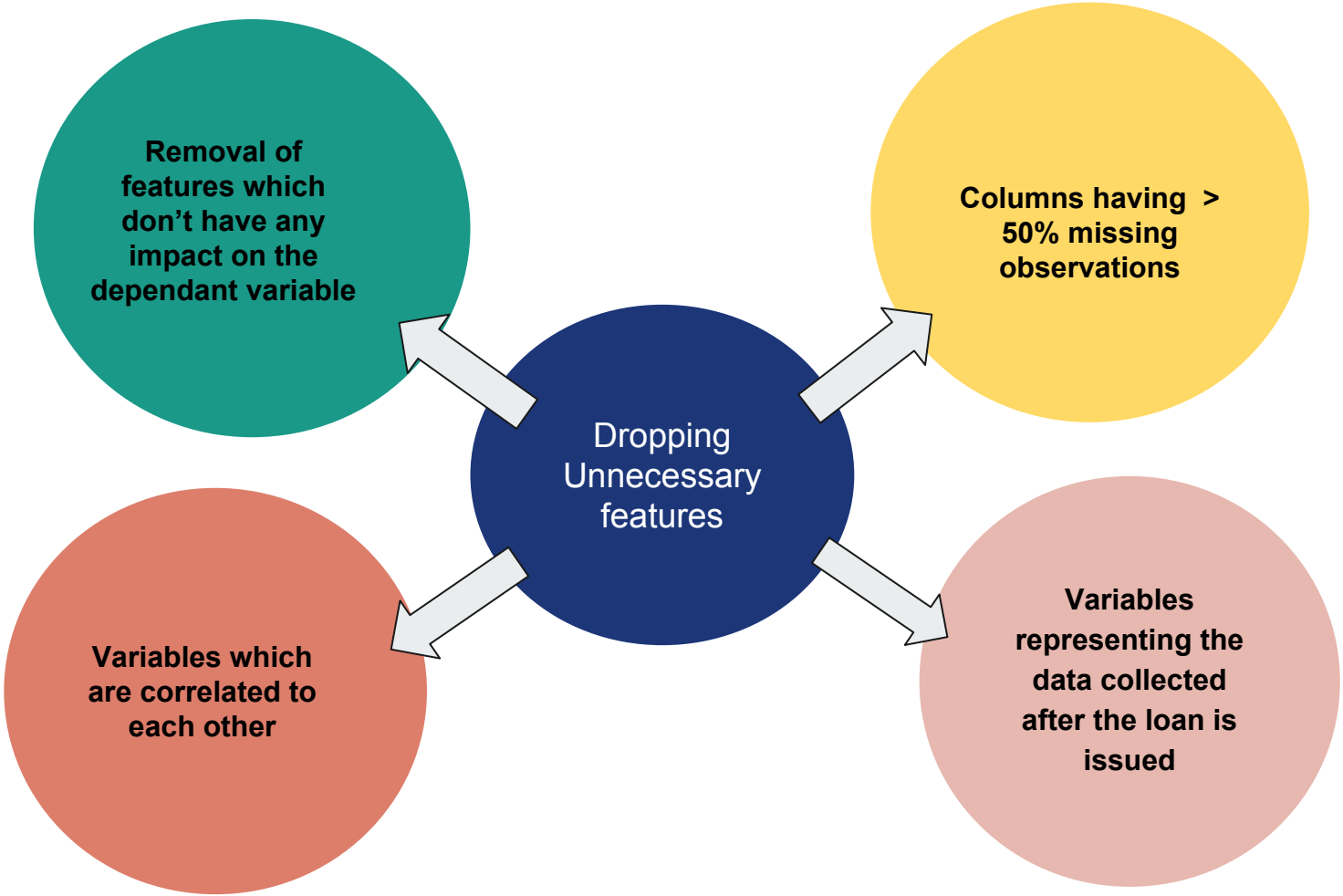
To build a classifier which will predict the defaulter and non defaulter on the basis of given past data more accurately

Introduction

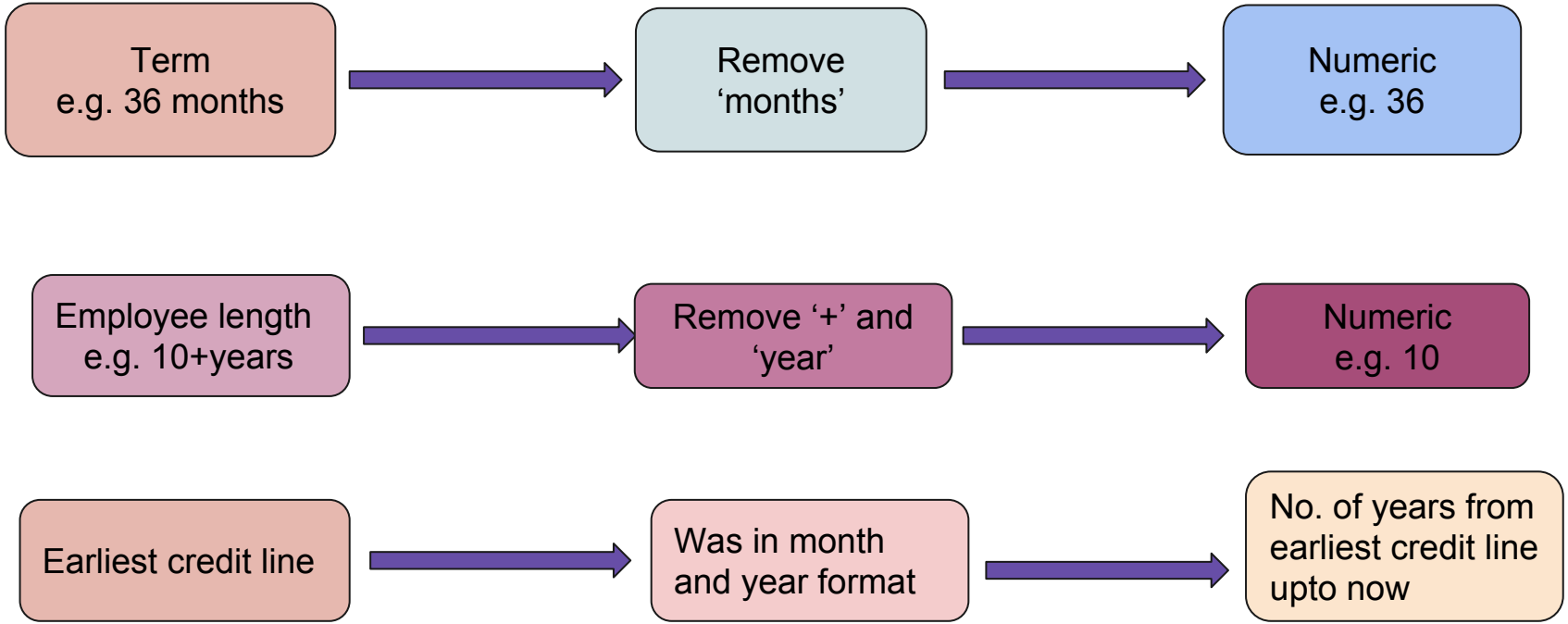
The dataset contains complete loan data for all loans issued by XYZ Corp. through 2007-2015 such as indicator of default, payment information, credit history, etc.

There are a total of 855969 rows and 73 columns.

Contains both categorical and numeric variables.

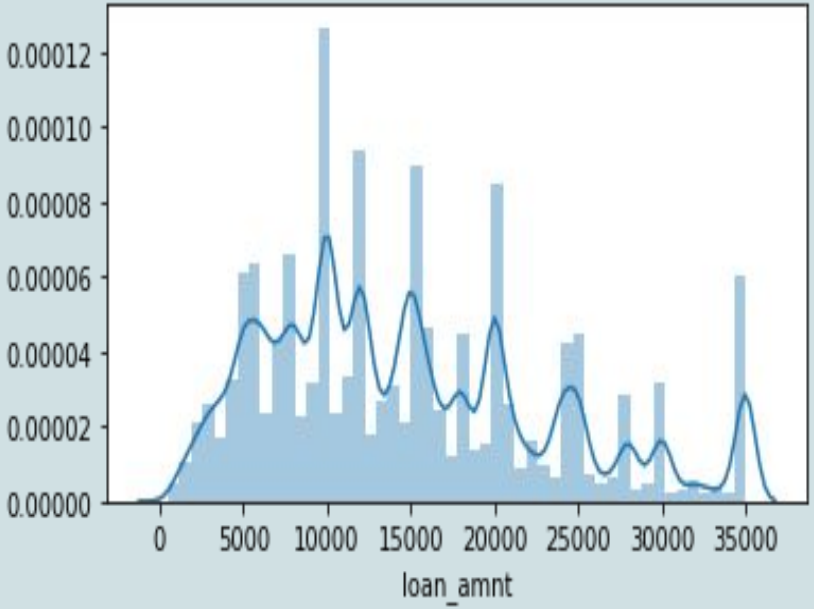

Removal of features which don't have any impact on the dependant variable
Columns having > 50% missing observations
Dropping Unnecessary features
Variables which are correlated to each other
Variables representing the data collected after the loan is issued

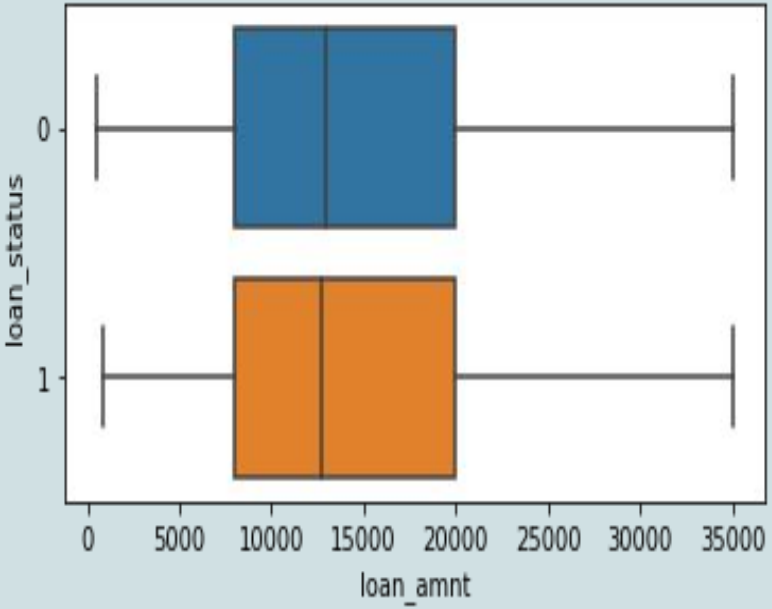
Variable Transformation



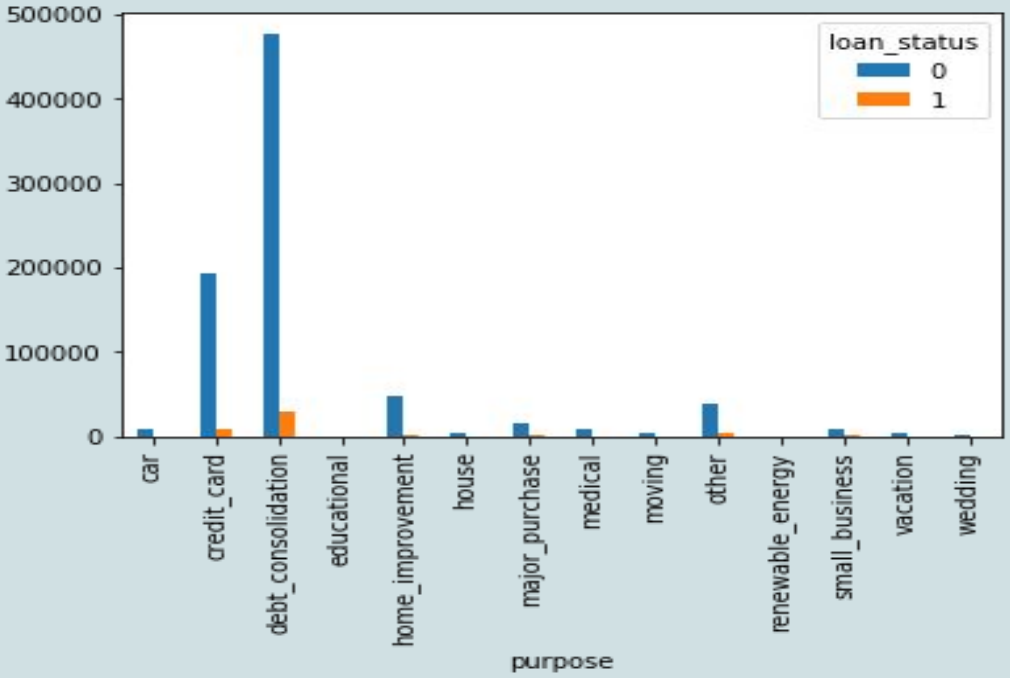
Exploratory Data Analysis (EDA)

How much loan the people are borrowing ?

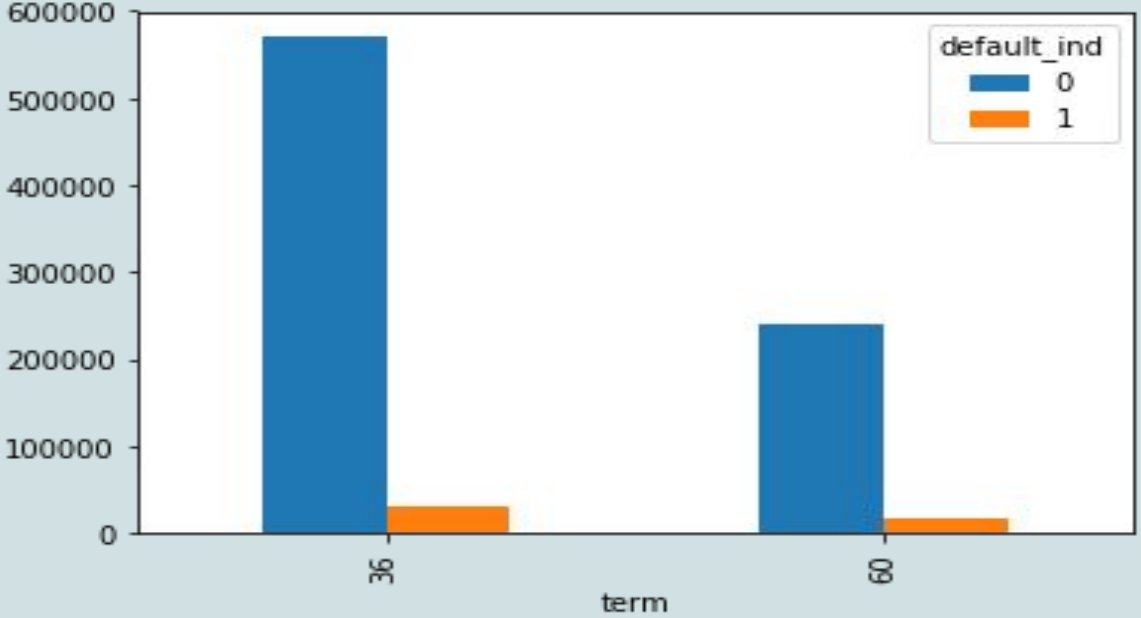




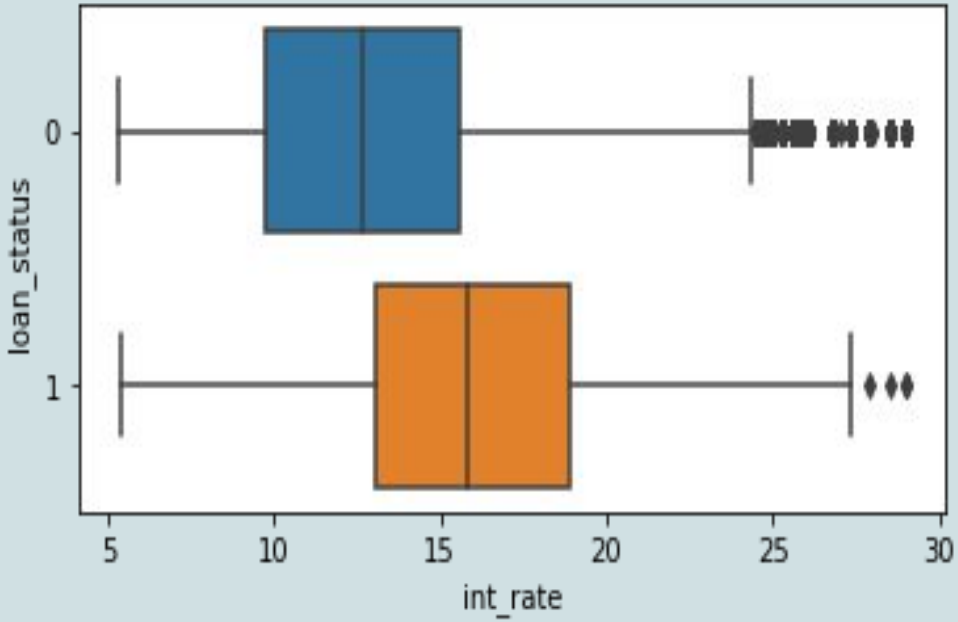
For which purpose most of the loan are borrowed by people?



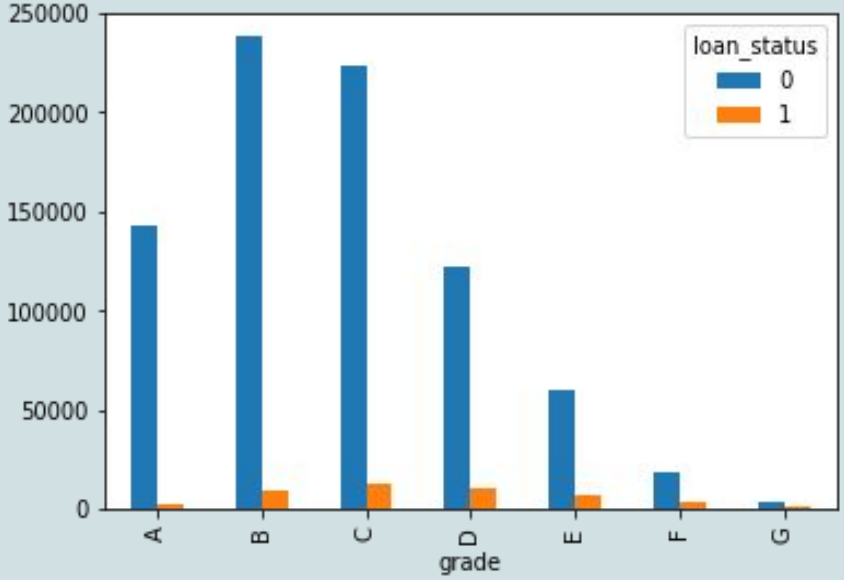
How long the loan terms are ?

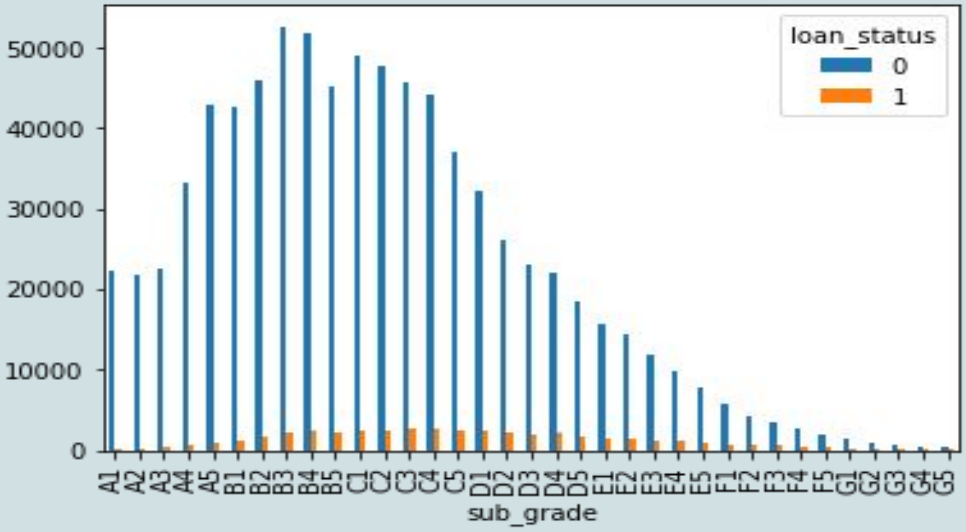


What kind of interest rate are borrowers paying ?

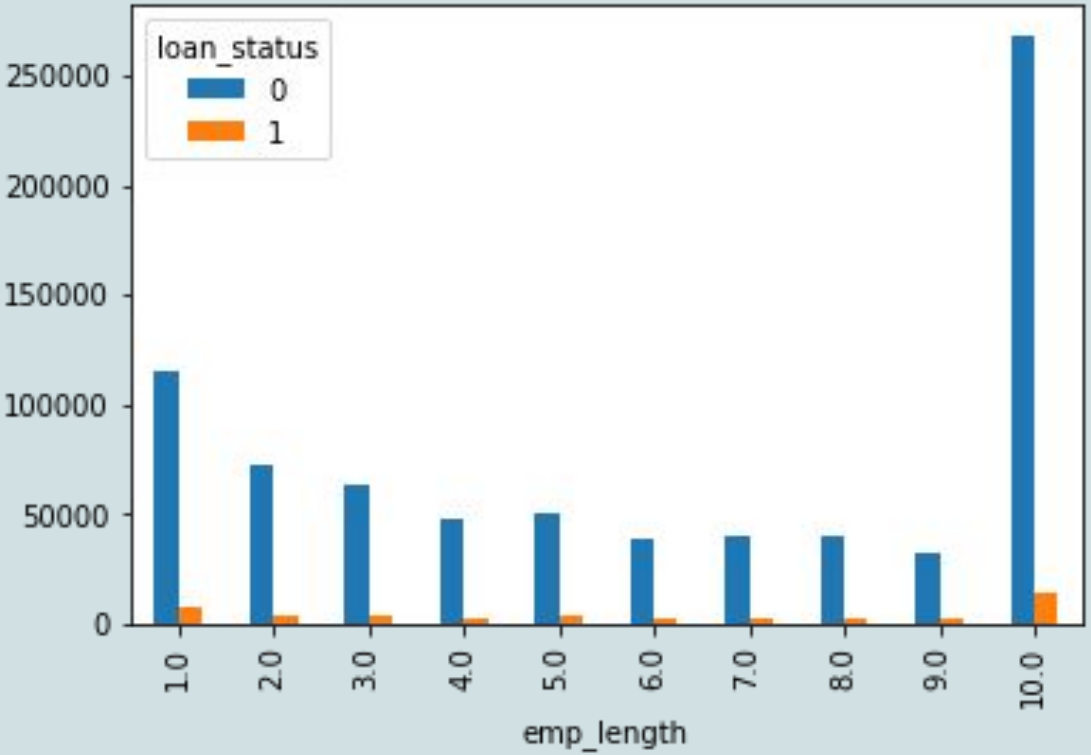


How Grades or sub-grades matters ?

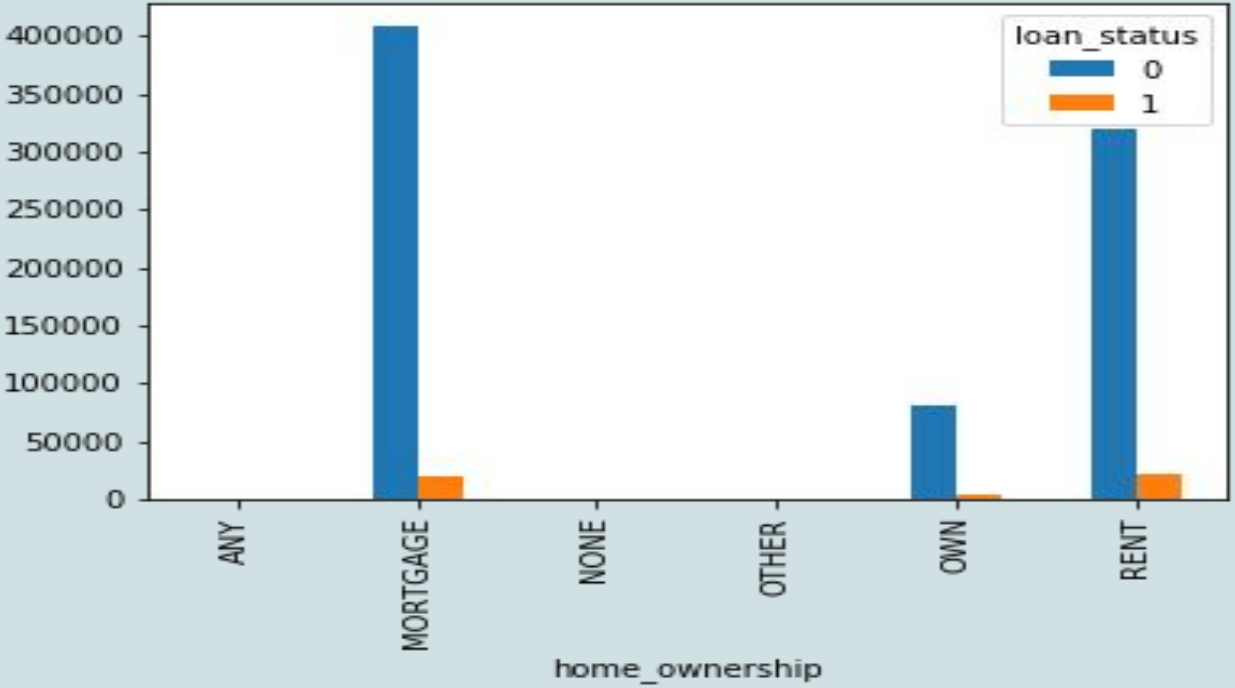




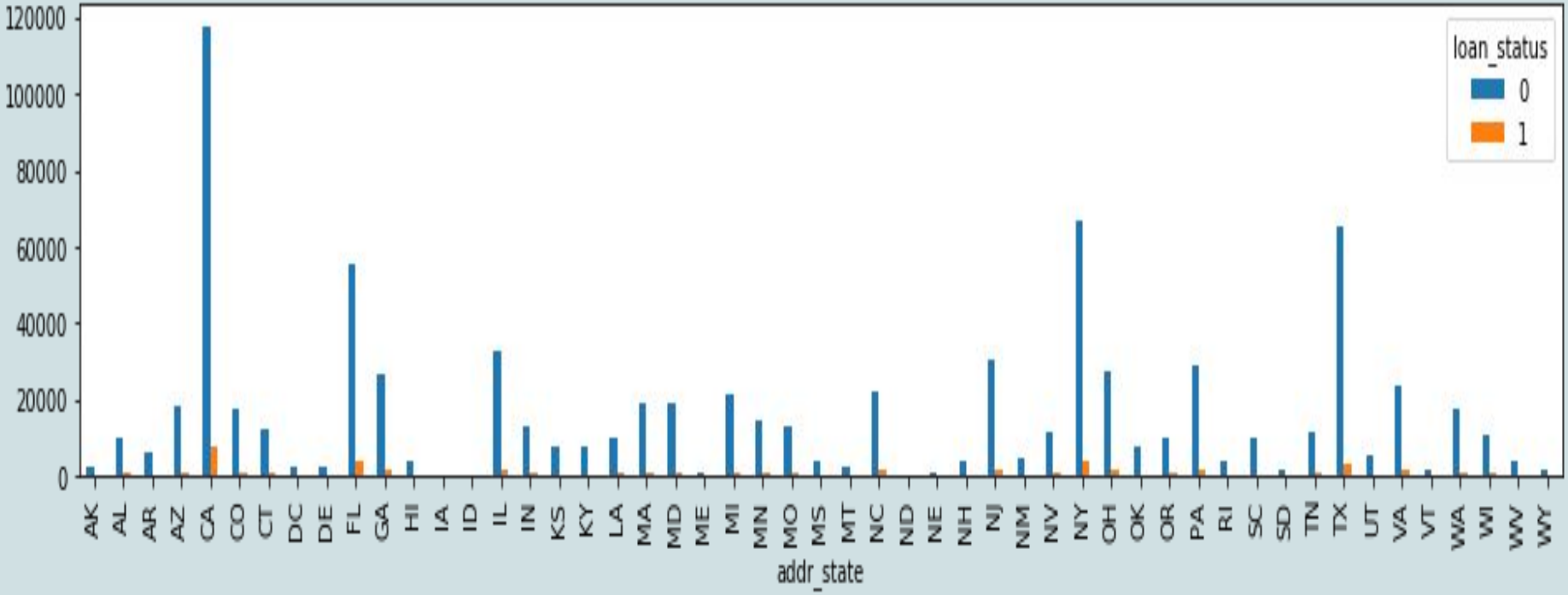
What is the professional experience of maximum number of non-defaulters?



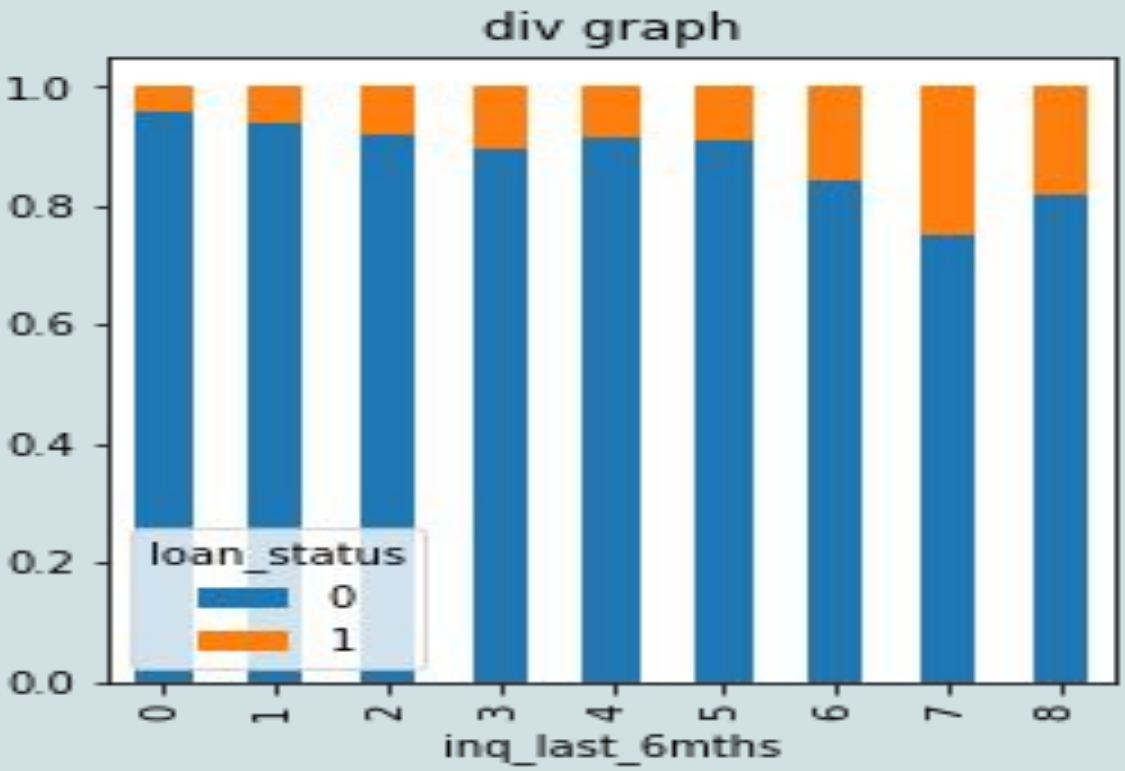
What about the home ownership ?



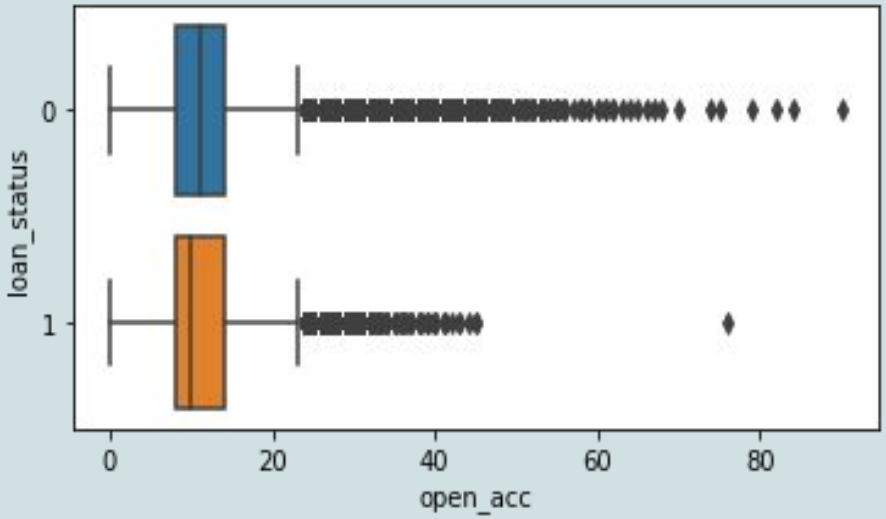
In which cities maximum loans are taken ?

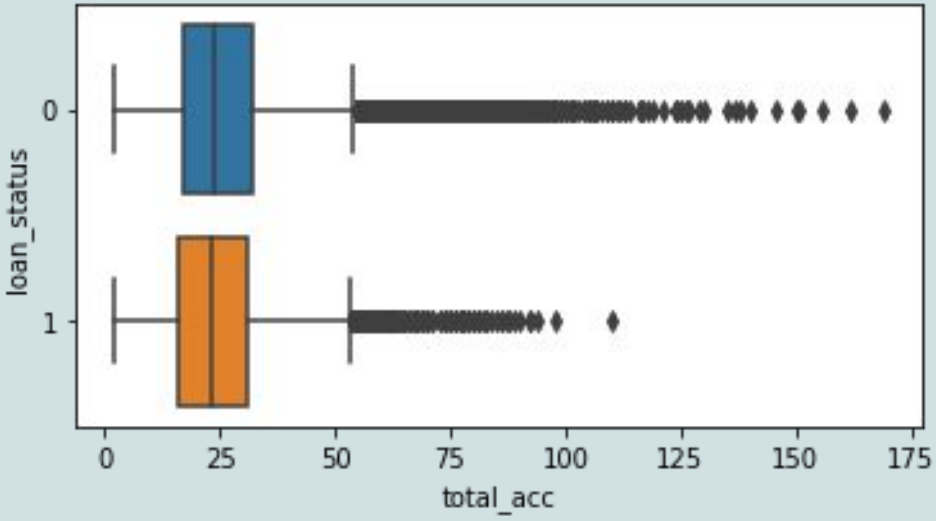


What about less or more inquiries ?

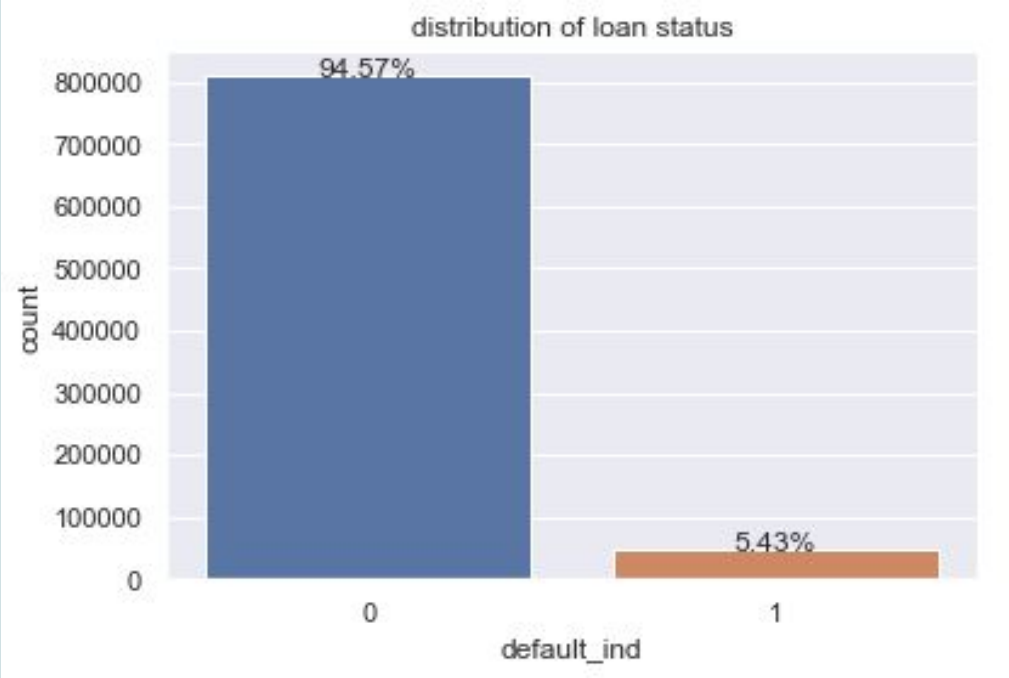


Total accounts and the loan status





Loan status



Maximum number of records fall under the non-defaulter category.

Missing value imputation

- The missing values in revolving line utilization rate and employee length were imputed with median.
- Records with high number of missing values were dropped.

Statistical significance

- **KS test** : For feature selection

  H0 : Two samples come from same distribution

  H1: Two samples come different distribution

- **chi-square tes**t : For feature selection

  H0 : there is NO association between both variables.

  H1: there is evidence to suggest there is an association between the two variables.

- **Correlation analysis :** The correlation between variables was checked so that only uncorrelated variables would be included in the model.

Correlation matrix

Data Preparation

- **Creating Dummies**
- **Splitting The Data:**

The data was divided into train and test sets using the variable ‘issue_d’.

Train data = June 2007 - May 2015

Out-of-time test data = June 2015 - Dec 2015

Modelling with original data

Logistic regression

```
log_score=accuracy_score(pred_log,y_test)
log_score
```

```
0.9987081259655007
```

```
# confusion matrix
confusion_matrix=confusion_matrix(y_test,pred_log)
confusion_matrix
```

```
array([[256659,     21],
       [   311,      0]], dtype=int64)
```

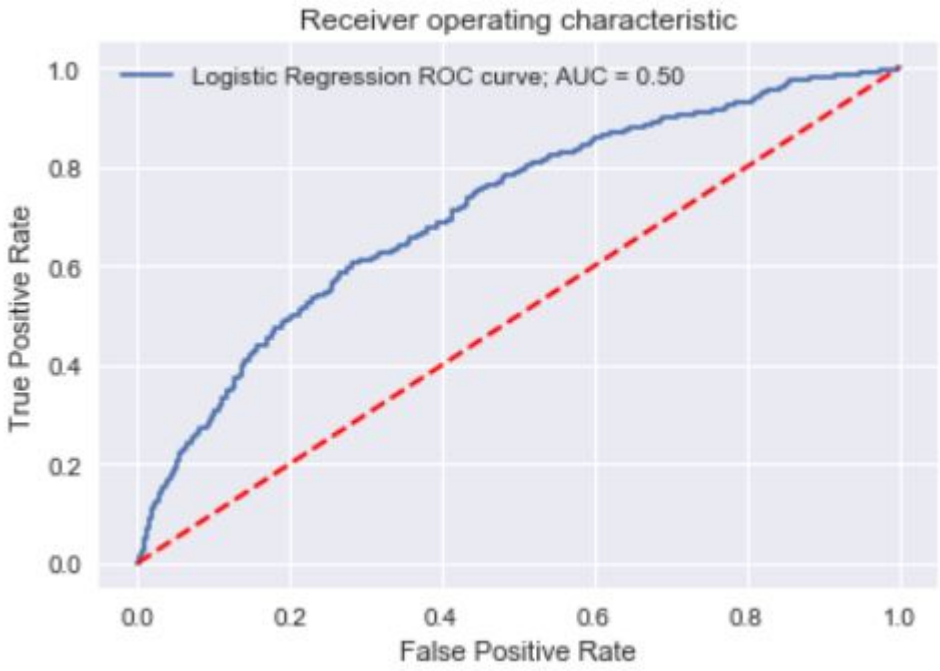
this is clear indication of imbalenced data our model is only learning to classify 0 and not 1

Techniques to Handle Imbalanced Data

- Under sampling
- Over sampling
- SMOTE

ROC of Original Data Model

Logistic Regression



Modelling with the scaled data

- Data scaling with standard scalar so as to bring all the data into the same scale
- Model building with scaled data
- Result : Behaving same as it was behaving with previous one

Modelling with scaled data

Logistic regression

```
0.9984707635675958
[[256598     82]
 [   311      0]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    256680
           1       0.00      0.00      0.00       311

   micro avg       1.00      1.00      1.00    256991
   macro avg       0.50      0.50      0.50    256991
weighted avg       1.00      1.00      1.00    256991
```

Modelling with oversample data

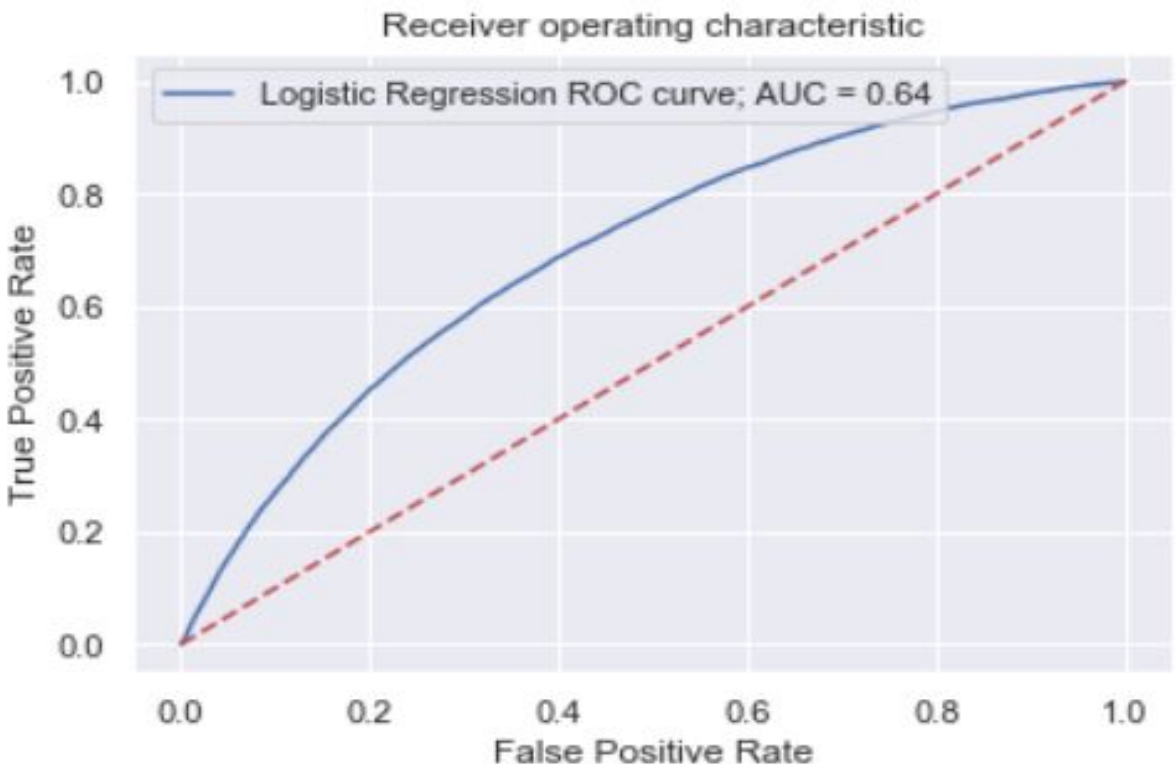
```
0.4672926289247483
[[119851 136829]
 [    72    239]]
             precision    recall  f1-score   support

          0       1.00      0.47      0.64    256680
          1       0.00      0.77      0.00       311

avg / total       1.00      0.47      0.64    256991
```

Model Building with Smote

Logistic regression result:

```
Accuracy of log model with smote: 0.6631982703930014

 confusion matrix for log model using smote:
  [[146952  74080]
 [   6615  11945]]

 classification_report for log model with smote:
              precision    recall  f1-score   support

          0       0.96      0.66      0.78    221032
          1       0.14      0.64      0.23     18560

avg / total       0.89      0.66      0.74    239592
```

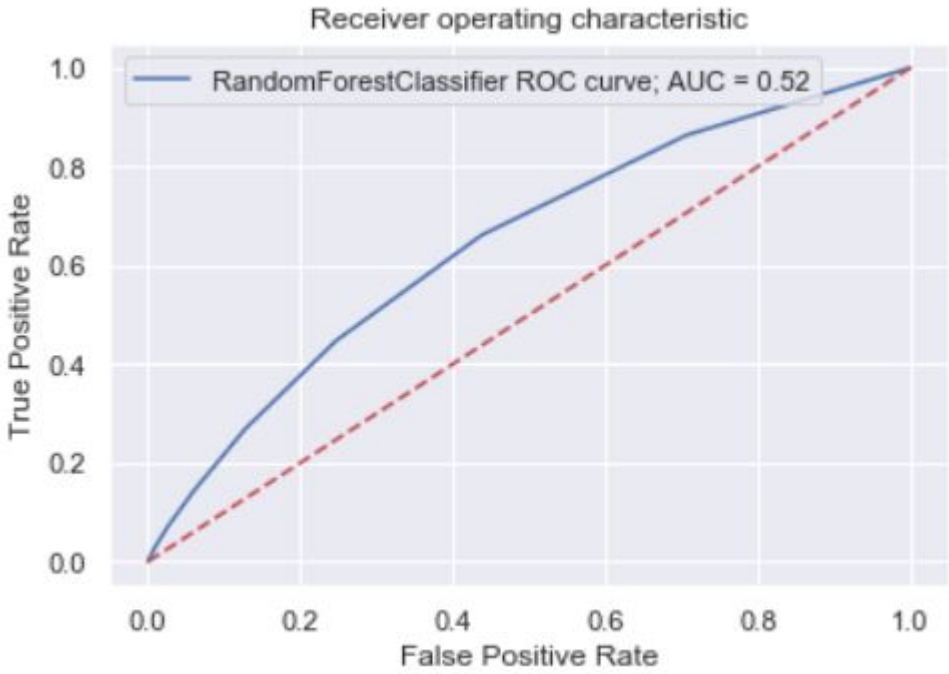
ROC for Logistic Regression with Smote



Random forest classifier with Smote

```
Accuracy of Randomforest model with smote : 0.9042789408661391

 confusion matrix for log model using smote:
 [[215360   5672]
 [ 17262   1298]]

 classification_report for log model with smote:
              precision    recall  f1-score   support

          0       0.93      0.97      0.95    221032
          1       0.19      0.07      0.10     18560

avg / total       0.87      0.90      0.88    239592
```

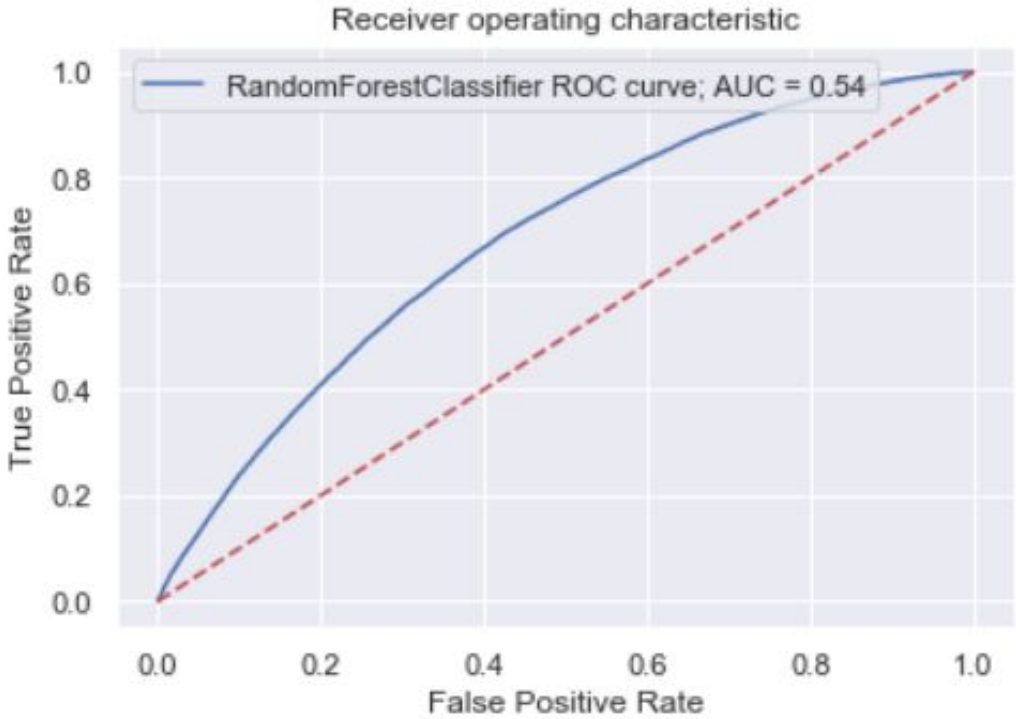
ROC for Random Forest with Smote



XGBoost classifier with Smote

```
Accuracy of XGBoost classifier model with smote : 0.8916950482486894

 confusion matrix for log model using smote:
 [[211544   9488]
 [ 16461   2099]]

 classification_report for log model with smote:
              precision    recall  f1-score   support

          0       0.93      0.96      0.94    221032
          1       0.18      0.11      0.14     18560

avg / total       0.87      0.89      0.88    239592
```

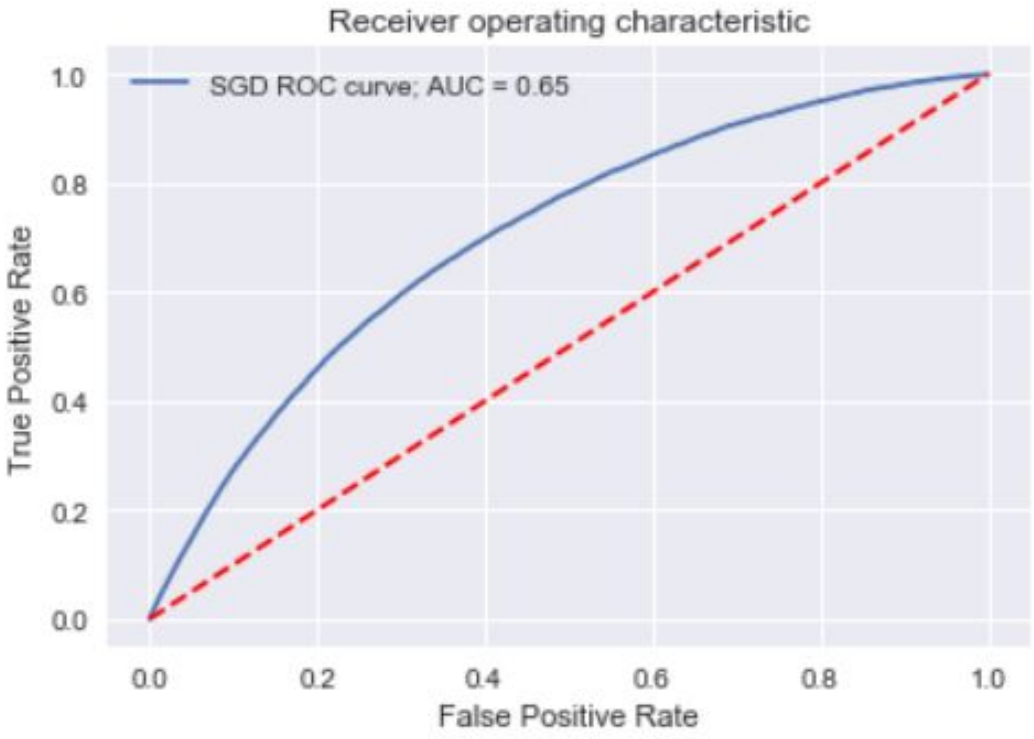
ROC for XGBoost



Stochastic Gradient Descent with logistic regression

```
Accuracy of SGD model with smote: 0.6503848208621323

 confusion matrix for SGD model using smote:
 [[143758  77274]
 [  6491  12069]]

 classification_report for SGD with smote:
               precision    recall  f1-score   support

           0       0.96      0.65      0.77    221032
           1       0.14      0.65      0.22     18560

    micro avg       0.65      0.65      0.65    239592
    macro avg       0.55      0.65      0.50    239592
 weighted avg       0.89      0.65      0.73    239592
```

ROC for Stochastic Gradient Descent with logistic regression



Making predictions on test data

Logistic Regression :

```
0.6596262125911024
[[169314  87366]
 [   107    204]]
              precision    recall  f1-score   support

           0       1.00      0.66      0.79    256680
           1       0.00      0.66      0.00       311

   micro avg       0.66      0.66      0.66    256991
   macro avg       0.50      0.66      0.40    256991
weighted avg       1.00      0.66      0.79    256991
```

Making prediction on test data

SGD classifier:

```
0.6478475899934238
[[166301  90379]
 [   121    190]]
              precision    recall  f1-score   support

           0       1.00      0.65      0.79    256680
           1       0.00      0.61      0.00       311

   micro avg       0.65      0.65      0.65    256991
   macro avg       0.50      0.63      0.40    256991
weighted avg       1.00      0.65      0.79    256991
```

Looking forward

- Feature selection
- outliers treatment
- More cost sensitive algorithm techniques and the hyperparameter tuning of parameters to optimize the accuracy of model.
- Data reduction kind of techniques like LDA, QDA
- The more efficient way to handle the imbalanced data so as to build an optimized classifier

Questions?


Thank
You