Ksolves India Limited
Suite No- 213, H-Block,
Building No. 221, Infinity Space,
Sector 63, Noida, 201301
+91 896 266 9996
sales@ksolves.com

LCF Project

21th June 2022

OVERVIEW

Our approach is to build the model using Logistic Regression and Xgboost Classifier. Firstly we are gathering the data from the LCF database using SQL queries. After that we performed EDA **(Exploratory Data Analysis)** and Feature Engineering. We preprocessed the data and applied our model Logistic Regression and Xgboost Classifier. We will discuss all the steps further in detail.

Gathering the Data

➢ All data directly comes to LCF's own database.
➢ We are getting the data using SQL queries from the LCF database.
➢ All data we are getting is in the CSV **(Comma Separated File)** form.

Performed EDA

➢ After getting the data we first performed Exploratory data analysis. We found both Categorical and Numerical data in our dataset.

➢ Firstly we checked how many missing values and duplicate values in our dataset, we found a lot of missing data in our dataset.

➢ Then we checked the distribution in our data by plotting Distplot whether our distribution is **Gaussian (Normal) distribution or Asymmetric distribution.** Distplot depicts the variation in the data distribution and represents the overall distribution of continuous data variables.

=================*=============================*========================

➢ After that we detected the outliers in our data by plotting the Boxplot and using **IQR (Interquartile Range).**

➢ Now we ascertained the correlation between multiple features. There are some features who are strongly correlated. Means features who are highly positive correlated with other features providing the same information.

➢ There are a lot of things we observed in the data and we worked on that.

Data Cleaning

Handling Missing Values

➢ There are lots of Nan values in data. For example: Some features have **50% - 74% missing values** which are not able to perform well within model training.
➢ We handled the features having less than **1 % missing values.** We imputed all the missing values with the mean() and median() according to data distribution.
➢ If the distribution of data is Normal or Gaussian distribution then we just imputed all the missing values by its mean. In normal distribution the mean and median value are almost closer.
➢ If the distribution of data is Asymmetric distribution or our distribution is skewed then we imputed mostly missing data with the median.

Handling Outliers

➢ We found some outliers in our data but some of the data points could represent some information so we didn't remove these outliers.

Handling Categorical features

➢ There are some categorical features in our dataset, some features represent Loan ID, Business_Name, Date and time for every renewal deal etc.
➢ These kinds of categorical features are not predictable, so we dropped these categorical features.

=================*=============================*=========================

Feature Selection

➢ Feature selection is basically used for filtering irrelevant or redundant features from your dataset.

➢ We checked correlation for all the features with respect to target variable "WriteOff_YN". Some features are strongly correlated with it and some are weak.

➢ We selected all those features that are highly correlated with 'Writeoff_YN' and the fluctuation in these features' values directly affects the target variable "WriteOff_YN".

Model Building

We have two approaches for building the model -

★ Logistic Regression

★ Xgboost (Extreme Gradient Boosting) Classifier

1st Approach - Model using Logistic Regression

Logistic regression is a statistical analysis method to predict **a binary outcome, such as yes or no,** based on prior observations of a data set. A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables.

Checking Skewness in data

```
# Skewness in Writeoff_YN -

print(y_train.value_counts(normalize=True).round(2))
print(' ')
print(y_test.value_counts(normalize=True).round(2))
```

```
0    0.74
1    0.26
Name: WriteOff_YN, dtype: float64

0    0.75
1    0.25
Name: WriteOff_YN, dtype: float64

74 % for 0 and 26% for 1  (On train data)
75% for 0 and 25% for 1   (On test data)
```

Selected Features for Logistic Model Training

```
Renewal_PreviousPace
Renewal_PreviousPaidPercentageOfRTR
Renewal_PreviousCreditScore
Renewal_CountPreviousDeals
Renewal_PreviousCountBounces
Renewal_PreviousPosition
Renewal_PreviousHBwLCF
Renewal_PreviousAvgDailyBalance
Renewal_TwoWeekBounces
NumberOfBankStatements
BankStatement_MA_OverdraftDays
BankStatement_MA_NumReturnItems
Credit_Score__c
Time_in_Business__c
Is_this_Business_home_based__c
HB_with_LCF_Payment__c
Contract_HoldBack__c
AvgMonthRevenue
WriteOff_YN
```

=================*=============================*========================

Applied Logistic Regression and Fit to model

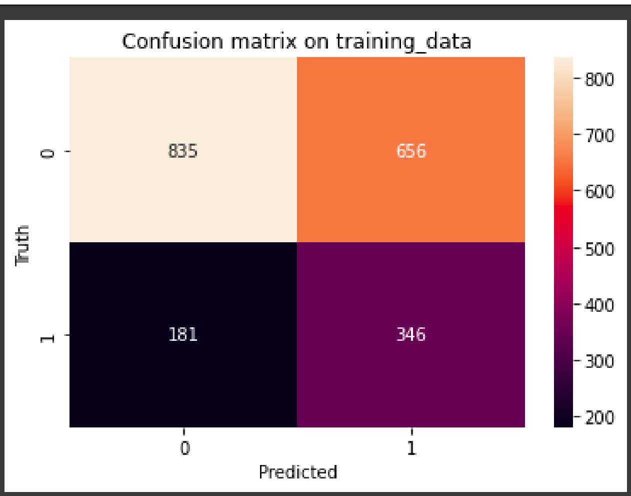
```
lg = LogisticRegression(C=0.001, class_weight='balanced',
solver='newton-cg')
lg.fit(X_train, y_train)

Output: LogisticRegression(C=0.001, class_weight='balanced',
solver='newton-cg')
```

Confusion Matrix for Training Data

★ We got **58% accuracy** on the training set and **F1-score is 0.4** which is quite bad.

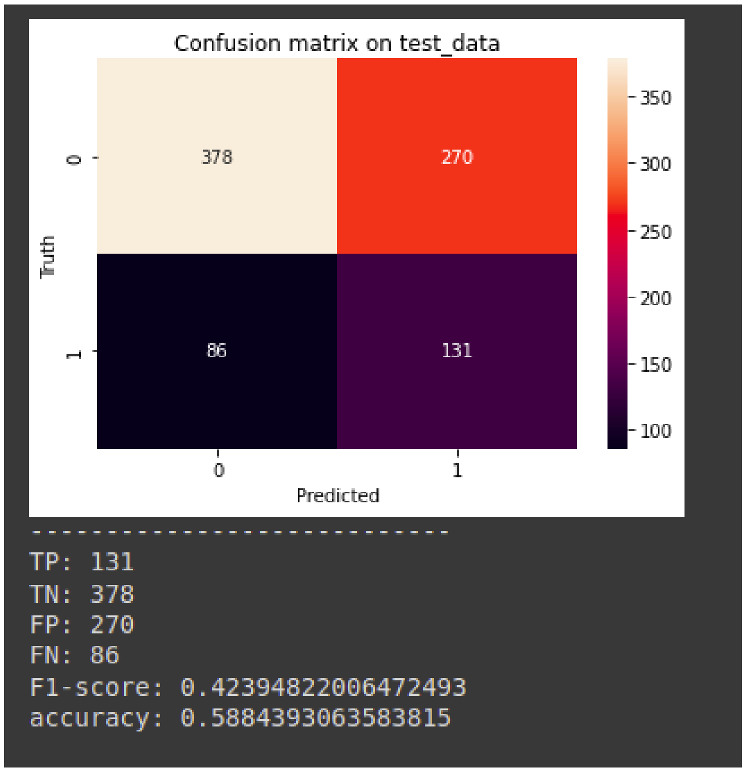
★ Our model is not performing well due to the imbalance and quality of data.



```
----------------------------
TP:  346
TN:  835
FP:  656
FN:  181
F1-score: 0.45258338783518637
accuracy: 0.5852329038652131
```

=====================*===*==============

Confusion Matrix on Test Data



```
----------------------------
TP:  131
TN:  378
FP:  270
FN:  86
F1-score:  0.42394822006472493
accuracy:  0.5884393063583815
```

★ We also got **58% accuracy** on the test set and **F1-score is 0.4** which is not good.

★ We tuned all the hyperparameters for getting the best weights and to improve the model accuracy.

★ But we achieved this accuracy with the best model parameters. Need to improve data for improving model performance.

======================*==*===============

2nd Approach - Model using Xgboost Classifier

Unlike many other algorithms, **XGBoost** is an ensemble learning algorithm meaning that it combines the results of many models, called base learners, to make a prediction.

Selected Features for Xgboost Model Training

```
Renewal_PreviousPace
Renewal_PreviousPaidPercentageOfRTR
Renewal_PreviousCreditScore
Renewal_CountPreviousDeals
Renewal_PreviousCountBounces
Renewal_PreviousPosition
Renewal_PreviousHBwLCF
Renewal_PreviousHBwoLCF
Renewal_PreviousAvgMonthRevenue
Renewal_PreviousAvgDailyBalance
Renewal_TwoWeekBounces
NumberOfBankStatements
BankStatement_MA_OverdraftDays
BankStatement_MA_NumReturnItems
Credit_Score__c
Time_in_Business__c
Is_this_Business_home_based__c
HB_with_LCF_Payment__c
Contract_HoldBack__c
AvgMonthRevenue
WriteOff_YN
```

==================*==============================*=========================

Applied Xgboost Classifier and Fit to model

```
#initializing Xgboost model
x_cfl = XGBClassifier()
clf = GridSearchCV(estimator=x_cfl, param_grid=parameters,
scoring='roc_auc', n_jobs=-1, cv=5, return_train_score=True)
clf.fit(X_train,y_train)

#best parameters after training
Clf.best_estimator_

Output: XGBClassifier(eta=0.01, eval_metric='aucpr', gamma=0.25,
lambda=1, n_estimators=40, scale_pos_weight=4, subsample=0.75,
use_label_encoder=False)
```

XGBoost Parameters Explanation

★ **eta -**
Learning rate - Prevents overfitting - Default = 0.3 - Range = [0,1]

★ **gamma -**
min_split_loss - Min loss reduction for a further partition - default = 0 - range = [0,infinity] (Higher gamma value means smaller loss reduction which means more splits. As a result, a more conservative model,chances of improvement in accuracy.Higher values can also cause overfitting).

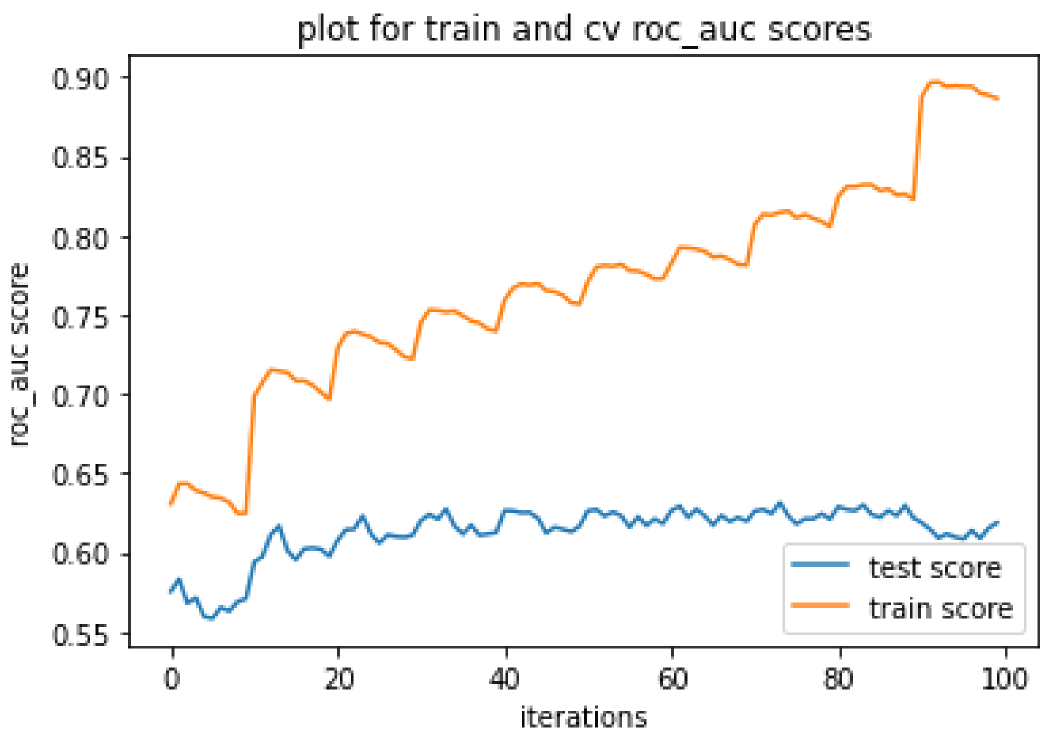
★ **lambda -**
L2 regularization - default=1 - Higher value means more conservative model, likely to overfit.

==================*==============================*========================

★ **alpha -**
L1 regularization - default=0 - Higher values lead to a more conservative model, likely to overfit.

★ **scale_pos_weight -**
default=1 - Control the balance of positive and negative weights, useful for unbalanced classes. A typical value to consider: sum(negative instances) / sum(positive instances).

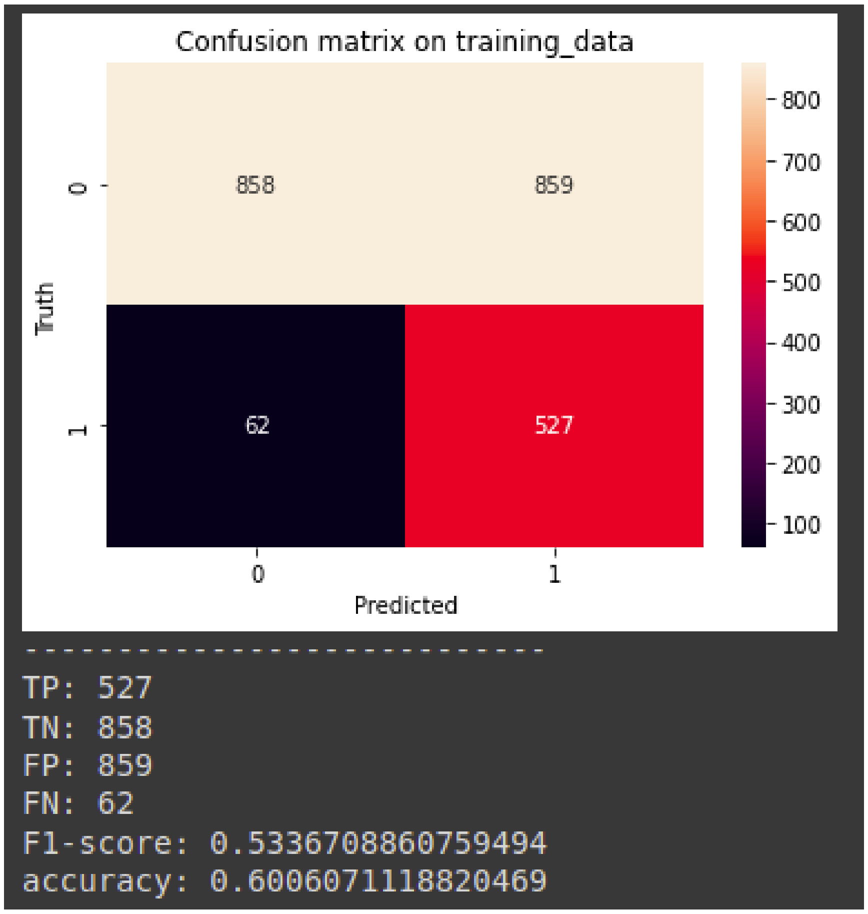
ROC_AUC Score



★ Our **AUC score** for training data is between **(0.80 - 0.90)** which is good but for test data it's not performing well.
★ Classifier yields a **score less than 0.65** for test data, it simply means that the model is performing worse than a random classifier.

=================*=============================*========================

Confusion Matrix for Training Data

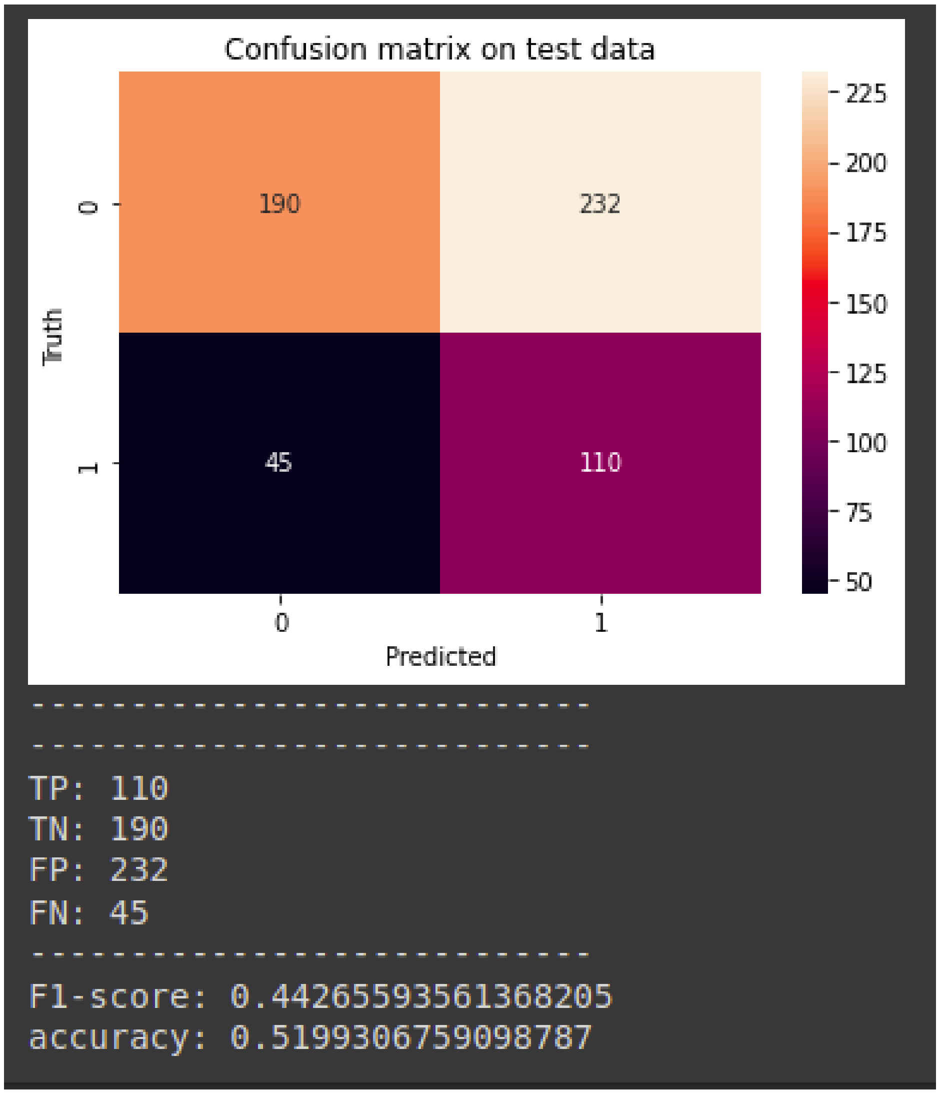


```
----------------------------
TP: 527
TN: 858
FP: 859
FN: 62
F1-score: 0.5336708860759494
accuracy: 0.6006071118820469
```

- ★ We got **60% accuracy** on the training set and **F1-score is 0.5** which is also quite bad.
- ★ We have to work on False Positive values to reduce these wrong predictions and to improve F1 - score and accuracy.
- ★ We achieved this accuracy with the best model parameters. We need to improve data for improving model performance.

==================*==============================*=========================

Confusion Matrix on Test Data



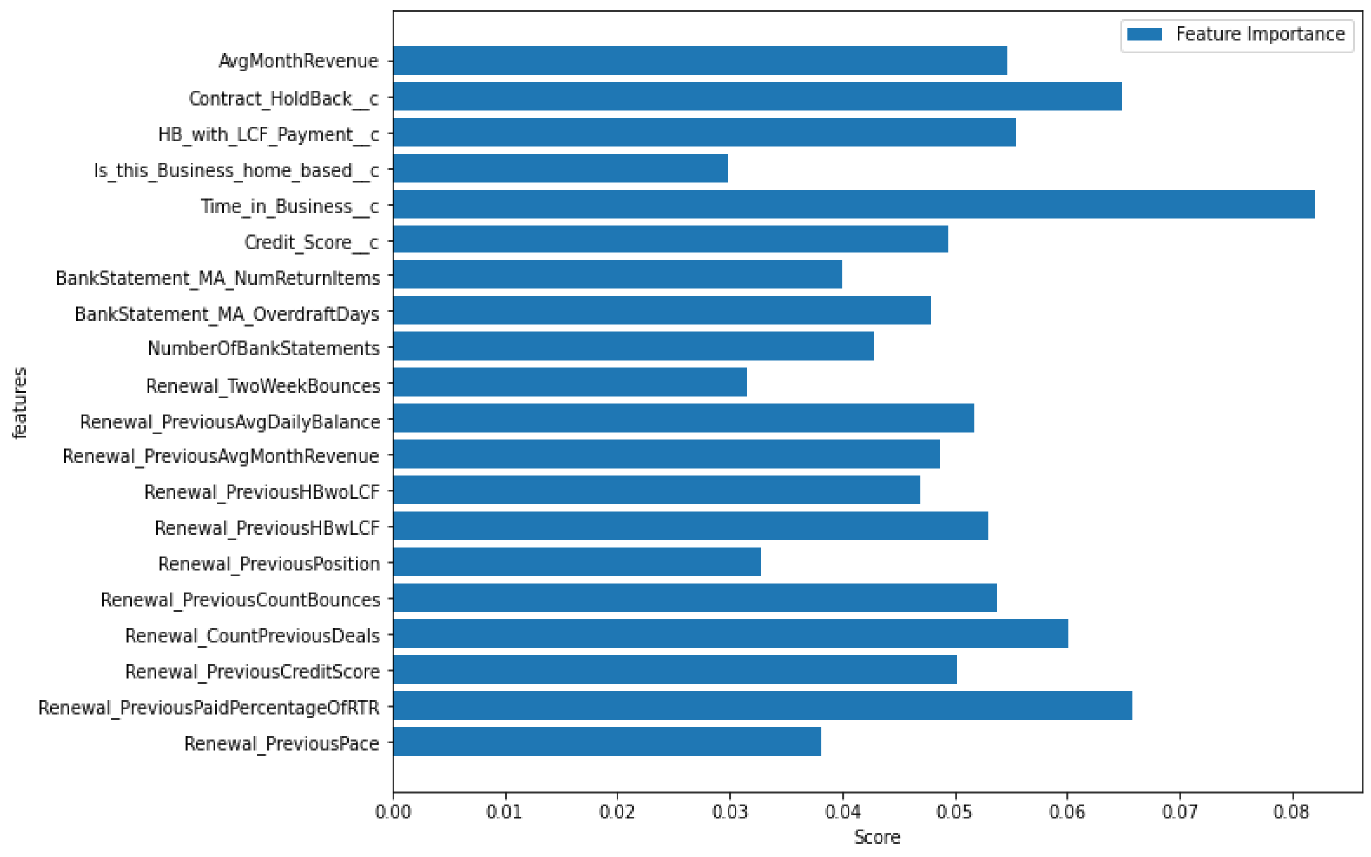
```
-----------------------------
-----------------------------
TP:  110
TN:  190
FP:  232
FN:  45
-----------------------------
F1-score: 0.44265593561368205
accuracy: 0.5199306759098787
```

★ We got **51% accuracy** on the test set and **F1-score is 0.4** means our Xgboost model is not performing well with this data.

★ We tuned all the hyperparameters for getting the best weights and to improve the model accuracy but still we got these results due to the bad data.

=================*=============================*========================

Feature Importance with respect to target variable (WriteOff_YN)


Feature Importance
features
AvgMonthRevenue
Contract_HoldBack__c
HB_with_LCF_Payment__c
Is_this_Business_home_based__c
Time_in_Business__c
Credit_Score__c
BankStatement_MA_NumReturnItems
BankStatement_MA_OverdraftDays
NumberOfBankStatements
Renewal_TwoWeekBounces
Renewal_PreviousAvgDailyBalance
Renewal_PreviousAvgMonthRevenue
Renewal_PreviousHBwoLCF
Renewal_PreviousHBwLCF
Renewal_PreviousPosition
Renewal_PreviousCountBounces
Renewal_CountPreviousDeals
Renewal_PreviousCreditScore
Renewal_PreviousPaidPercentageOfRTR
Renewal_PreviousPace
Score