CREDIT CARD LEAD PREDICTION

Submitted by: Priyal Agarwal

Project Introduction:

- Happy Customer Bank is a mid-sized private bank that deals in all kinds of banking products, like Savings accounts, Current accounts, investment products, credit products, among other offerings.

- The bank also cross-sells products to its existing customers and to do so they use different kinds of communication like tele-calling, e-mails, recommendations on net banking, mobile banking, etc.

- In this case, the Happy Customer Bank wants to cross sell its credit cards to its existing customers. The bank has identified a set of customers that are eligible for taking these credit cards.
- Bank wants to identify customers that could show higher intent towards a recommended credit card, given:

a) Customer details (gender, age, region etc.)

b) Details of his/her relationship with the bank (Channel_Code,Vintage, 'Avg_Asset_Value etc.)

- Building a model that's capable of identifying customers who are interested for the credit card .

Mathematical/ Analytical Modelling of the Problem/Exploratory Data Analysis Steps

DATA SOURCE

The data is provided from Analytics Vidhya platform contest named JOB-A-THON.

DATA FEATURES CHECK

- □ Extract feature information about dataset such as number of rows ,columns and data types of the different features.
- □ In this dataset, we have 351037 rows with 12 features.

.

NULL VALUE CHECK

- □ Check for the null values present in our dataset.
- □ Null values are present in our dataset in 'Credit_Product' feature.

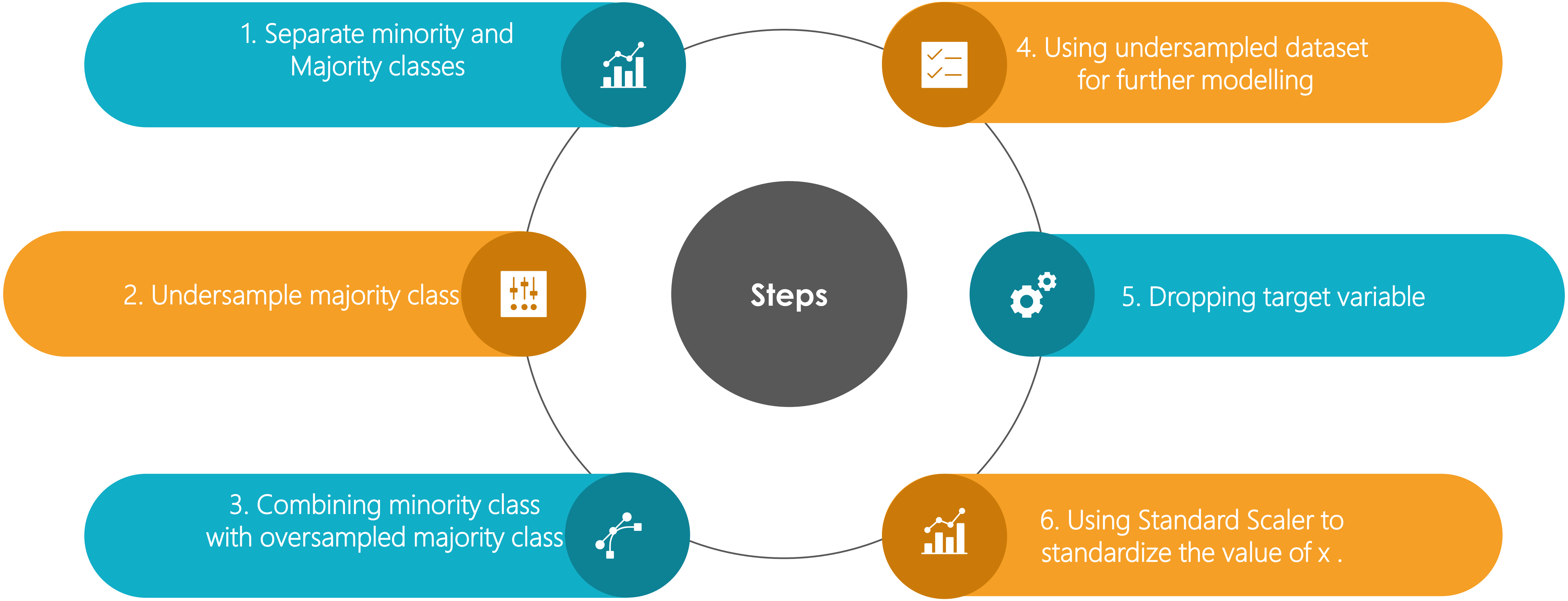
STATISTICS CHECK

This part tells about the statistics i.e. mean, median, max value ,min values ,75% and it also gives some sort of outliers' analysis

CHECK DATATYPES

- □ Check for the datatypes of features present in our dataset.
- □ There are 6 categorical features that needs to be converted in numerical datatype by using Label Encoder.

Data Pre-processing Steps



Steps

1. Separate minority and Majority classes

2. Undersample majority class

3. Combining minority class with oversampled majority class

4. Using undersampled dataset for further modelling

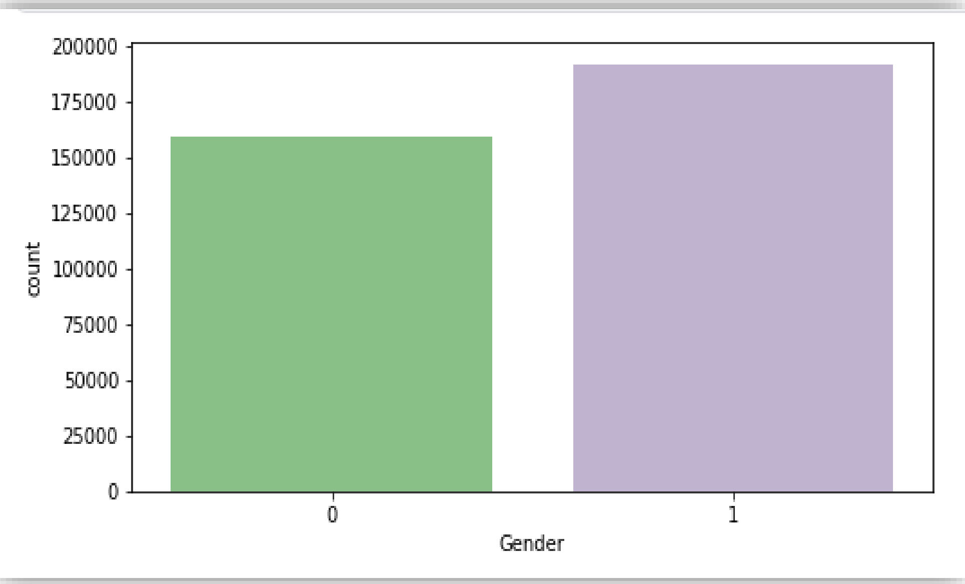
5. Dropping target variable

6. Using Standard Scaler to standardize the value of x .

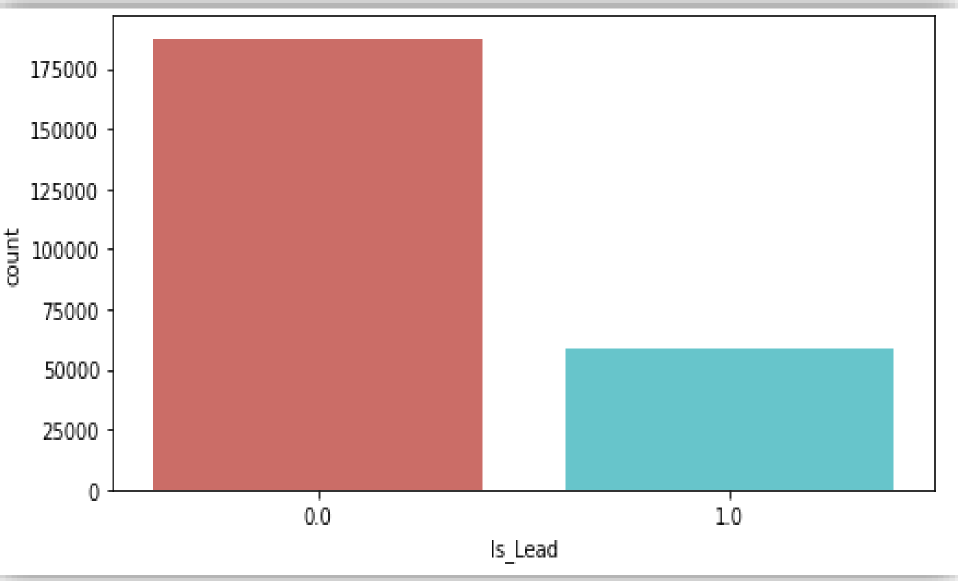
Visualizations

Univariate Analysis:

Countplot for 'Gender' variable



Countplot for 'Is_Lead' (Target) variable



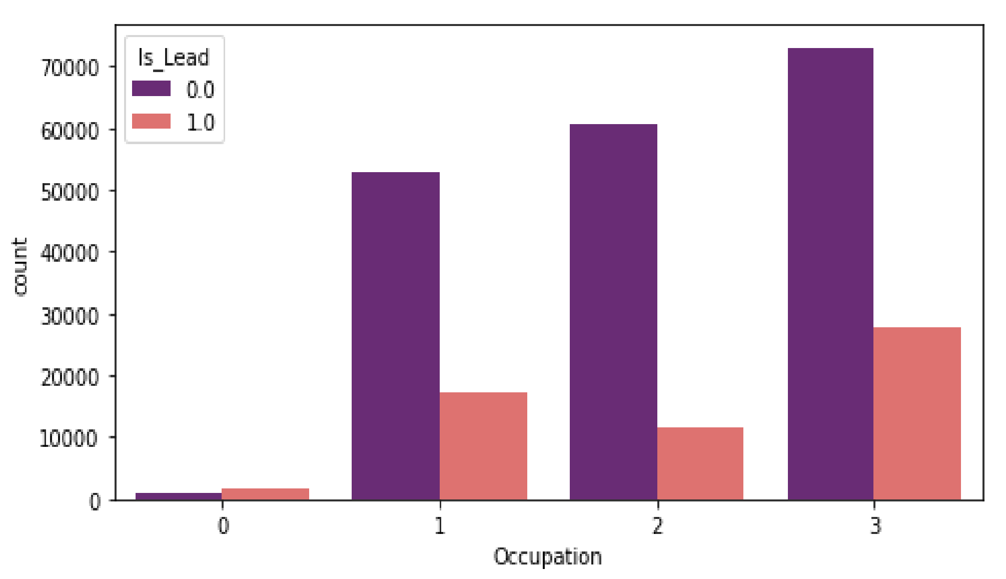
Observation: More Male customers are present in the dataset.

Observation: It shows that data is highly imbalanced and needs to be corrected .

Visualizations

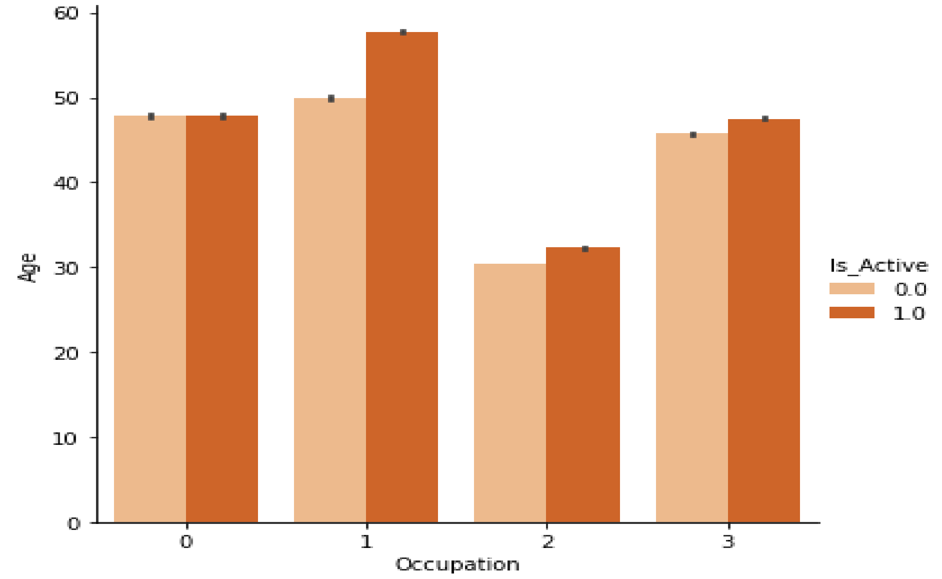
Bivariate Analysis:

Checking occupation with customers interest



Observation: Self employed customers are less likely to get the credit card. Whereas entrepreneurs (though limited) are most likely to get credit card.

Checking Activness of customer in last 3 months



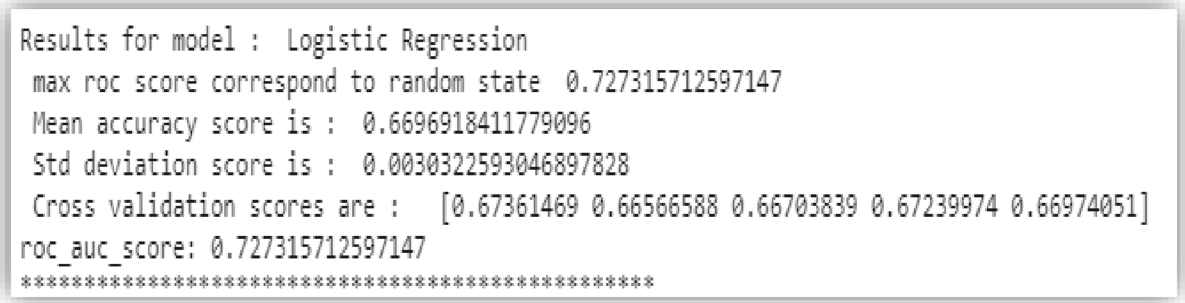
Observation: Active customers are more in salaried, self_employed and others as compared to entrepreneur in last 3 months.

Modelling Parts

1. Imbalanced dataset is **normalized** for final modeling .
2. After splitting the data for input and output **Standard Scaler** is applied to data.
3. After **train test split** applied all the classification algorithms to find the best scoring one.
4. As this is a Classification Problem so we have used F1 score, Accuracy Score, Confusion Matrix and ROC curve for evaluation of final model.
5. We have used ROC score and ROC_AUC curve **to finalize testing dataset prediction results**.
6. On the basis of AUC score , finalized Random Forest Algorithm for initial predictions.

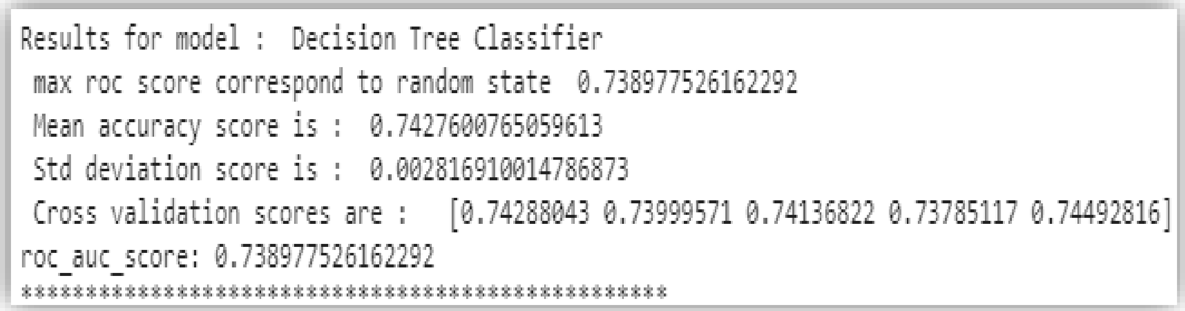
Algorithms Used

Logistic Regression



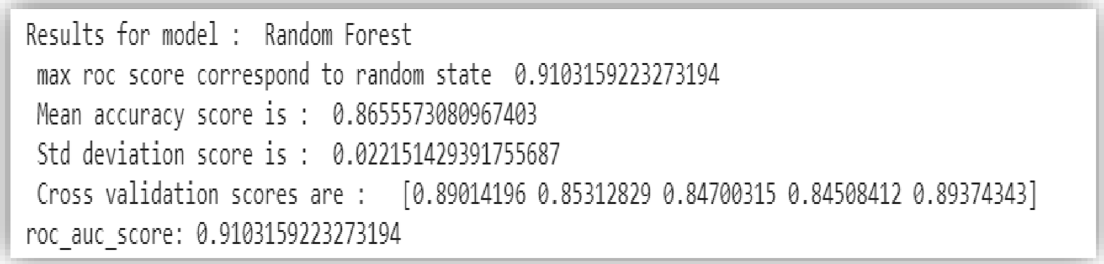
```
Results for model :  Logistic Regression
 max roc score correspond to random state  0.727315712597147
 Mean accuracy score is :  0.6696918411779096
 Std deviation score is :  0.0030322593046897828
 Cross validation scores are :   [0.67361469 0.66566588 0.66703839 0.67239974 0.66974051]
roc_auc_score: 0.727315712597147
**************************************************
```

Decision Tree Classifier



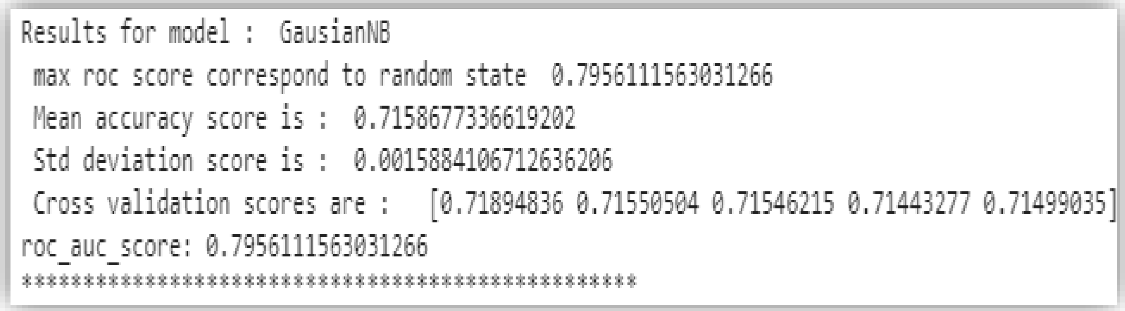
```
Results for model :  Decision Tree Classifier
 max roc score correspond to random state  0.738977526162292
 Mean accuracy score is :  0.7427600765059613
 Std deviation score is :  0.002816910014786873
 Cross validation scores are :   [0.74288043 0.73999571 0.74136822 0.73785117 0.74492816]
roc_auc_score: 0.738977526162292
**************************************************
```

Random Forest



```
Results for model :  Random Forest
 max roc score correspond to random state  0.9103159223273194
 Mean accuracy score is :  0.8655573080967403
 Std deviation score is :  0.022151429391755687
 Cross validation scores are :   [0.89014196 0.85312829 0.84700315 0.84508412 0.89374343]
roc_auc_score: 0.9103159223273194
```

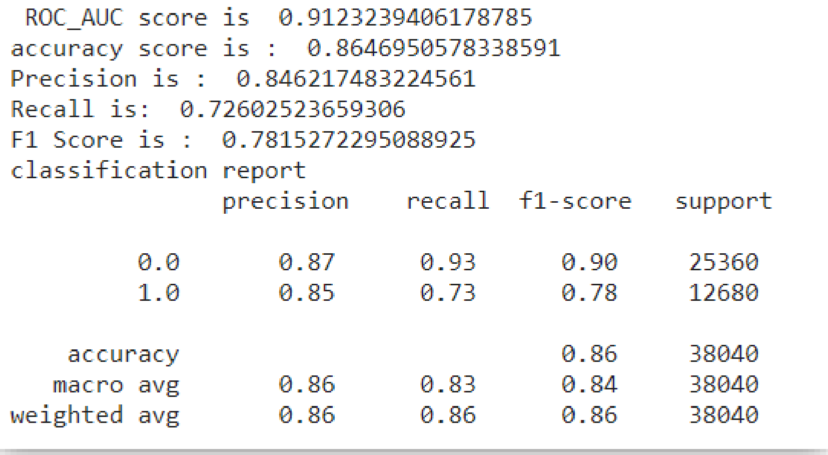
GausianNB



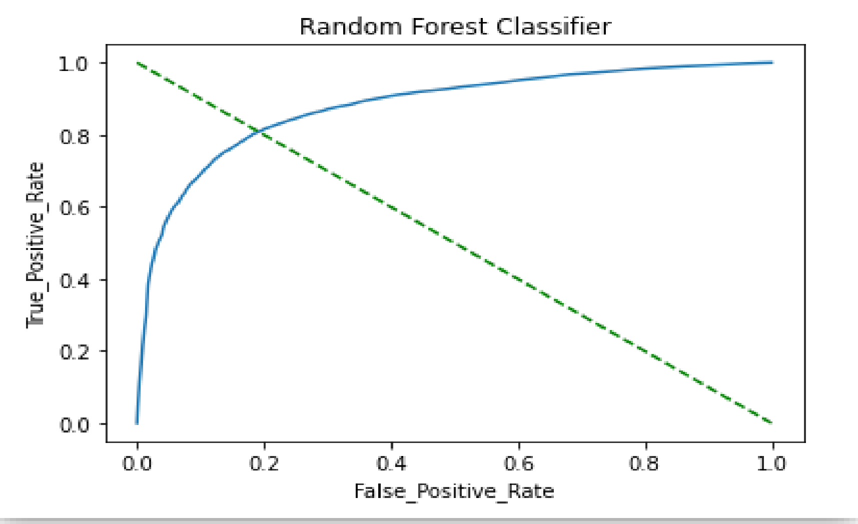
```
Results for model :  GausianNB
 max roc score correspond to random state  0.7956111563031266
 Mean accuracy score is :  0.7158677336619202
 Std deviation score is :  0.0015884106712636206
 Cross validation scores are :   [0.71894836 0.71550504 0.71546215 0.7143277 0.71499035]
roc_auc_score: 0.7956111563031266
**************************************************
```

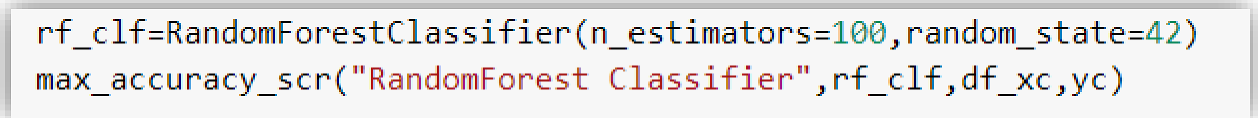
Observation: On the basis of AUC score , finalized Random Forest Algorithm for initial predictions.

Attempt 1 : Random Forest Classifier



```
 ROC_AUC score is  0.9123239406178785
accuracy score is :  0.8646950578338591
Precision is :  0.846217483224561
Recall is:  0.72602523659306
F1 Score is :  0.7815272295088925
classification report
               precision    recall  f1-score   support

         0.0       0.87      0.93      0.90     25360
         1.0       0.85      0.73      0.78     12680

    accuracy                           0.86     38040
   macro avg       0.86      0.83      0.84     38040
weighted avg       0.86      0.86      0.86     38040
```



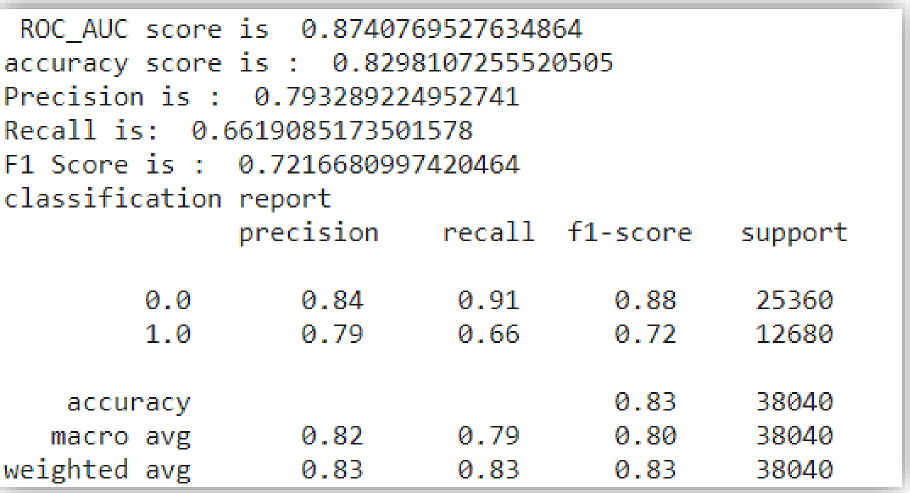


```
rf_clf=RandomForestClassifier(n_estimators=100,random_state=42)
max_accuracy_scr("RandomForest Classifier",rf_clf,df_xc,yc)
```

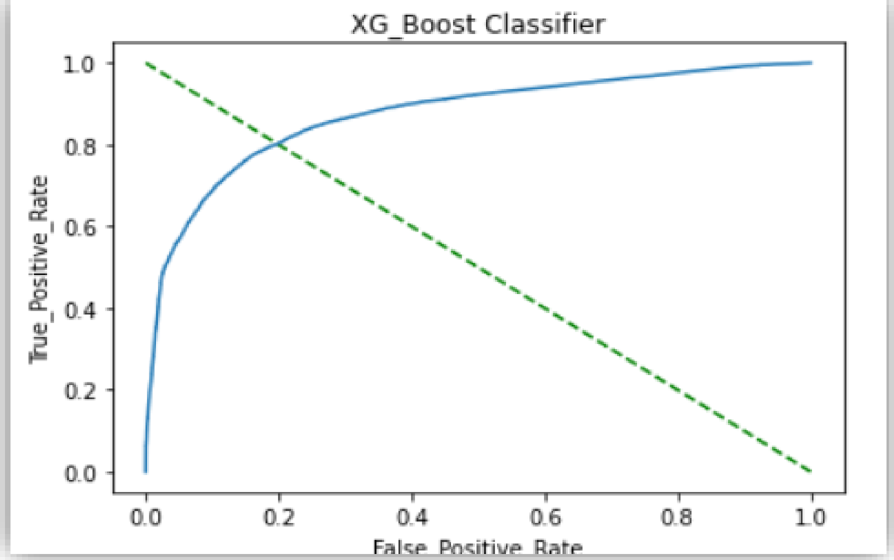
Result

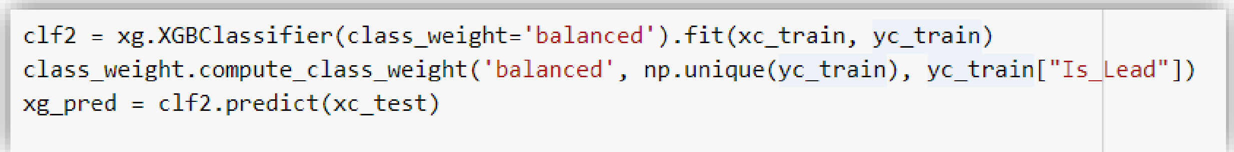
- **Base model selected is Random Forest (selected on basis of AUC score) which provides max ROC score of 0.91**
- Plotted AOC/ROC line that shows good match between test a nd predicted values.
- Also plotted confusion matrix, Overall model fit is good
- **However, as the predicted probability was meaned in RF mo del upto 2 decimal places the resultant AUC score with test data was found to be ~ 0.85**

Attempt 2 : XG Boost Classifier



```
 ROC_AUC score is  0.8740769527634864
accuracy score is :  0.8298107255520505
Precision is :  0.793289224952741
Recall is:  0.6619085173501578
F1 Score is :  0.7216680997420464
classification report
               precision    recall  f1-score   support

         0.0       0.84      0.91      0.88     25360
         1.0       0.79      0.66      0.72     12680

    accuracy                           0.83     38040
   macro avg       0.82      0.79      0.80     38040
weighted avg       0.83      0.83      0.83     38040
```



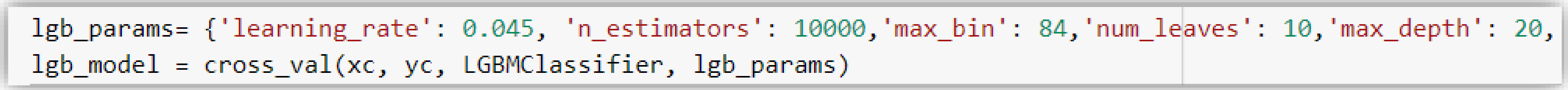


```
clf2 = xg.XGBClassifier(class_weight='balanced').fit(xc_train, yc_train)
class_weight.compute_class_weight('balanced', np.unique(yc_train), yc_train["Is_Lead"])
xg_pred = clf2.predict(xc_test)
```

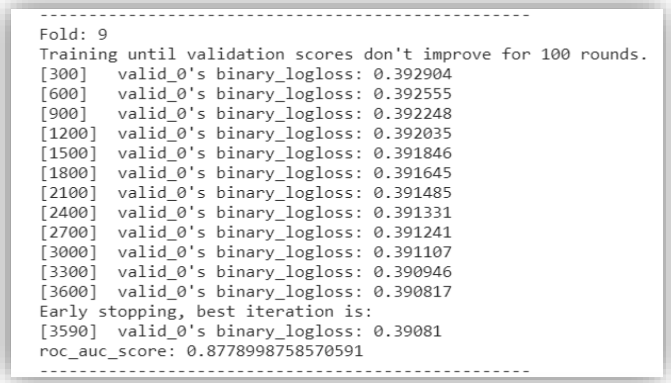
Result

- **Base model selected is XG Boost Classifer selected to boost accuracy in imbalanced class classification program.**
- Plotted AOC/ROC line that shows good match between test and predicted values.
- Max ROC score is 0.87, Overall model fit is good.
- **However, XG boost AUC score with test data dropped to ~ 0.86 due to overfitting issues.**

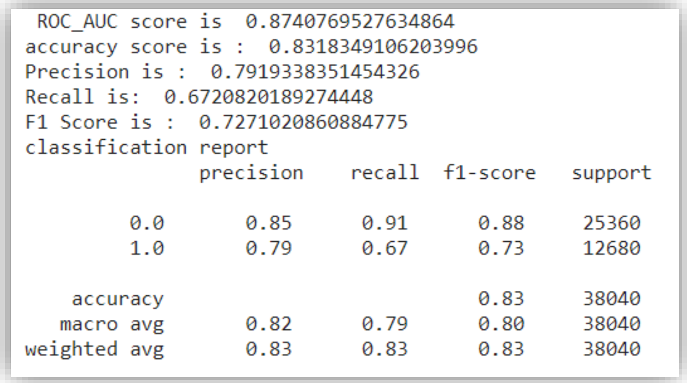
Attempt 3 : LGBM Model with Stratification Folds



```
lgb_params= {'learning_rate': 0.045, 'n_estimators': 10000,'max_bin': 84,'num_leaves': 10,'max_depth': 20,
lgb_model = cross_val(xc, yc, LGBMClassifier, lgb_params)
```



```
--------------------------------------------------------
Fold: 9
Training until validation scores don't improve for 100 rounds.
[300]   valid_0's binary_logloss: 0.392904
[600]   valid_0's binary_logloss: 0.392555
[900]   valid_0's binary_logloss: 0.392248
[1200]  valid_0's binary_logloss: 0.392035
[1500]  valid_0's binary_logloss: 0.391846
[1800]  valid_0's binary_logloss: 0.391645
[2100]  valid_0's binary_logloss: 0.391485
[2400]  valid_0's binary_logloss: 0.391331
[2700]  valid_0's binary_logloss: 0.391241
[3000]  valid_0's binary_logloss: 0.391107
[3300]  valid_0's binary_logloss: 0.390946
[3600]  valid_0's binary_logloss: 0.390817
Early stopping, best iteration is:
[3590]  valid_0's binary_logloss: 0.39081
roc_auc_score: 0.8778998758570591
--------------------------------------------------------
```



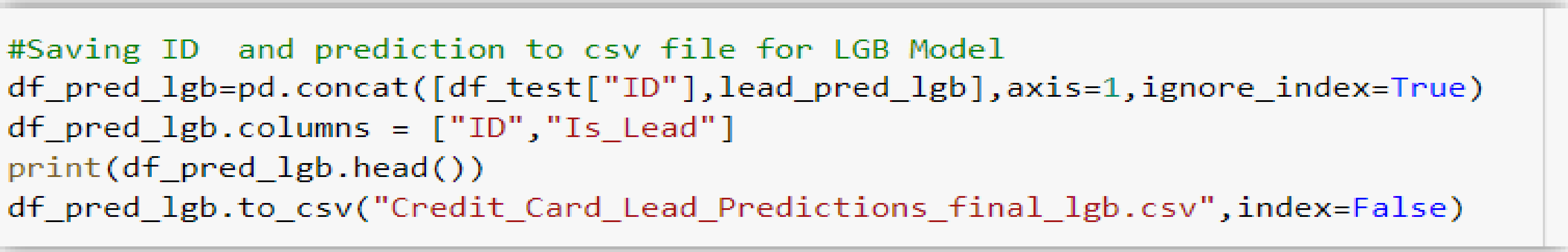
```
 ROC_AUC score is  0.8740769527634864
accuracy score is :  0.8318349106203996
Precision is :  0.7919338351454326
Recall is:  0.6720820189274448
F1 Score is :  0.7271020860884775
classification report
              precision    recall  f1-score   support

         0.0       0.85      0.91      0.88     25360
         1.0       0.79      0.67      0.73     12680

    accuracy                           0.83     38040
   macro avg       0.82      0.79      0.80     38040
weighted avg       0.83      0.83      0.83     38040
```

Result

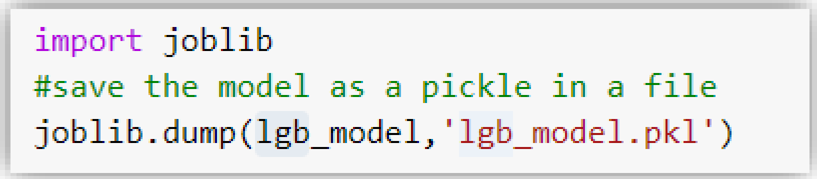
- **Base model selected is LGBM classifier model along with stratified cross-validation of 10 folds .**
- This was done to remove any overfitting issues in the model.
- Plotted AOC/ROC line that shows good match between test and predicted values.
- Max ROC score is 0.874

Final model is selected as LGBM model as it is most consistent model with highest AUC score in test data

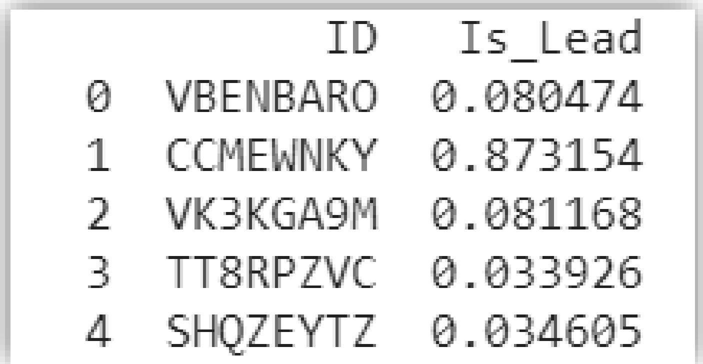
Final Model - Prediction



```
#Saving ID  and prediction to csv file for LGB Model
df_pred_lgb=pd.concat([df_test["ID"],lead_pred_lgb],axis=1,ignore_index=True)
df_pred_lgb.columns = ["ID","Is_Lead"]
print(df_pred_lgb.head())
df_pred_lgb.to_csv("Credit_Card_Lead_Predictions_final_lgb.csv",index=False)
```



```
import joblib
#save the model as a pickle in a file
joblib.dump(lgb_model,'lgb_model.pkl')
```



```
         ID   Is_Lead
0  VBENBARO  0.080474
1  CCMEWNKY  0.873154
2  VK3KGA9M  0.081168
3  TT8RPZVC  0.033926
4  SHQZEYTZ  0.034605
```

Result

- **Predictions were made using various models against test data – RandomForest , XG Boost and LGBM.**
- Following AUC score was observed:
  - RandomForest – 0.854
  - XG Boost – 0.86
  - LGBM – 0.87
- Final predictions with LGBM is chosen and model saved to **pkl file** and **predictions saved to csv file**

Conclusion

- Data contained both **categorical and numerical data**. Converted categories to numerical for EDA analysis.
- Also conducted **visual analysis** to observe following:
  - IndentActive customers are more in salaried,self_employed and others as compared to entrepreneur in last 3 months.
  - Data is skewed towards left in Avg_Account_Balance
  - Target Variable is imbalanced and needed to be corrected for proper modelling.
- Dataset was balanced by **using under sampling technique.**
- **Random Forest Classifier:**
  - Found RandomForest model had the highest AUC score(0.91) among various base models.
  - However, as the predicted probability was mean in RF model upto 2 decimal places the resultant AUC score with test data was found to be ~ 0.85
- **XG Boost Classifier:**
  - To further boost the accuracy XG Boost method was used and AUC score of 0.87 was found with the training data.
  - However, XG boost AUC score with test data dropped to ~ 0.86 due to overfitting issues.
- **LGBM Classifier with stratified cross-validation:**
  - To solve overfitting issues, LGBM model with 10-fold cross-validation was used and AUC score 0.874 with training data.
  - Model performed very well with test data and provided AUC score of ~0.871
- **Hence, final model is selected as LGBM model as it is most consistent model with highest AUC score**.

Thank You