CSCA 5622 Introduction to Machine Learning:Supervised Learning

Default of Credit Card Clients

Purpose

- The purpose of Default of Credit Card Client kaggle project is to build a supervised Machine learning model that predicts whether a credit card client will default on their payment in the next month.
- This is a binary classification problem based on real world financial and demographic data from clients in Taiwan, collected between April and Sept 2005.
- Multiple algorithms like Logistic Regression,Random Forest Classifier,XGBoost and SVC were trained and evaluated.

Data Source Citation

1.UCI Machine Learning Repository:

Yeh, I. (2009). Default of Credit Card Clients [Dataset]. UCI Machine Learning Repository. https://doi.org/10.24432/C55S3H.

2.Kaggle Platform:

The dataset was accessed from the Kaggle platform:

Default of Credit Card Clients Dataset. (n.d.). Www.kaggle.com. https://www.kaggle.com/datasets/uciml/default-of-credit-card-clients-dat

Data Source

Data Size:

- Number of rows: 30,000
- Number of features: 25

Data Types:

3 Categorical features and the remaining Numerical features.

- Education
- Sex
- Marriage

Data Cleaning

- The dataset contains 3 categorical variables: Education,Sex and Marriage
- Rare or undefined values in Education([0,5,6]) were grouped under category 4,and Marriage value 0 was reassigned to 3(“Others”)
- The column PAY_0 was renamed to PAY_1for consistency with other payment status features.
- The dataset contains no missing values or duplicate entries,ensuring data integrity.
- A significant class imbalance exists,with the majority of samples belonging to class 0(non-default).Techniques such as resampling or SMOTE may be applied to address this imbalance.

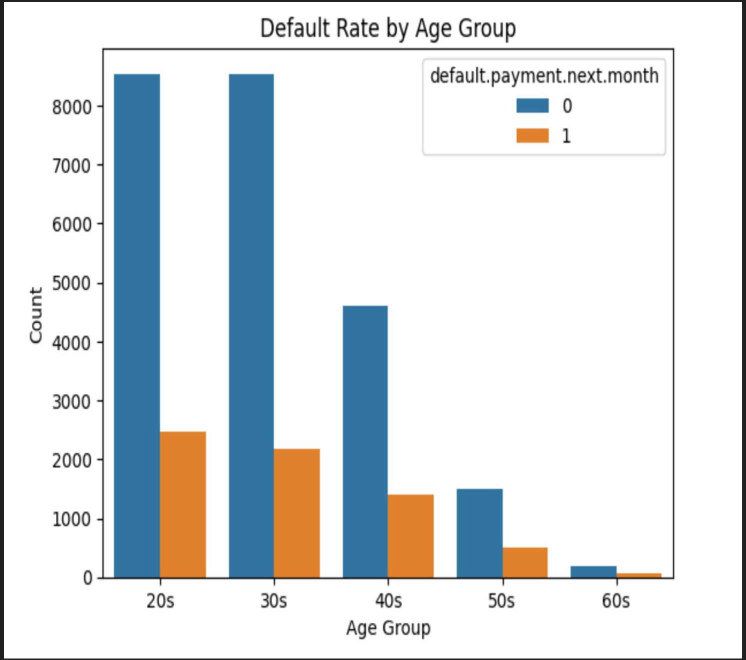
Exploratory Data Analysis

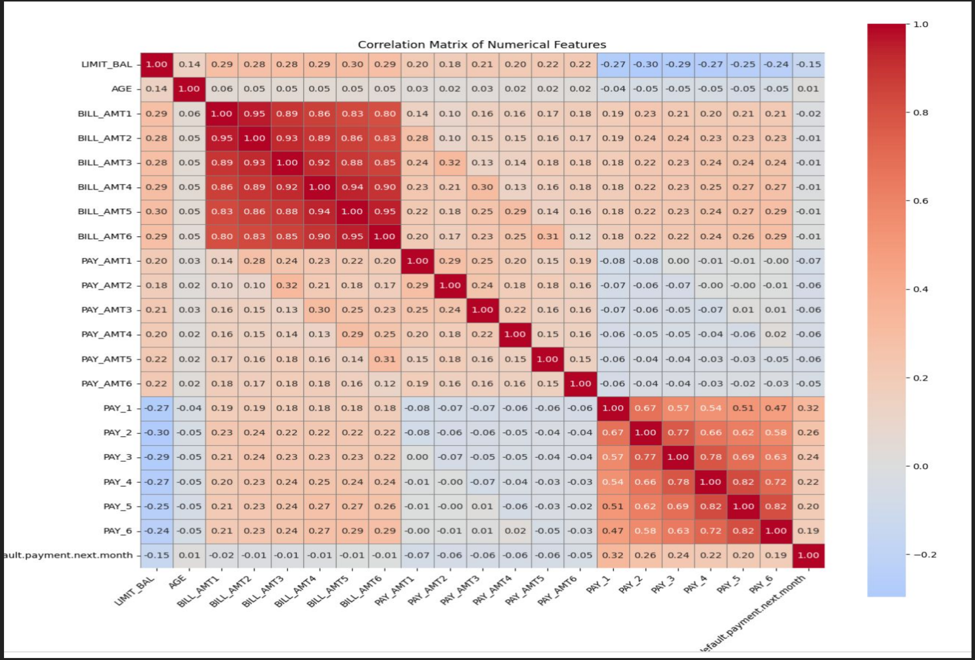
- **Categorical Feature Distributions:**The three categorical variables—EDUCATION, MARRIAGE, and SEX—were analyzed and visualized to understand their relationship with default status.
- **Gender vs. Default Status:**Although females make up 60% of the dataset, males show a higher default rate (24% of all males). Class 0 (non-default) dominates both genders due to class imbalance.
- **Education Levels:**Most clients, both defaulters and non-defaulters, belong to the Graduate School and University categories. Other education levels contribute minimally, likely due to sparse data.

Exploratory Data Analysis(Contd.)

- Marriage Status: Defaulters are primarily found in the 'Single' and 'Married' groups.
- Age Distribution: Age was binned for visualization; clients in their 20s show the highest default rate, possibly due to limited savings.
- Correlation Insights: LIMIT_BAL and PAY_1 show strong correlation with default status, indicating their predictive importance.
- Compute Variance Inflation factor to check Multicollinearity
- Check Outliers by Credit Limit
- Credit Limit Patterns: Non-defaulters tend to have higher and more flexible credit limits; defaulters are assigned lower limits, reflecting risk-based credit allocation.
- Perform Statistical Tests like chi square tests(Categorical columns) and Anova Test (Numerical columns)
- Kde plot on the strongest predictor

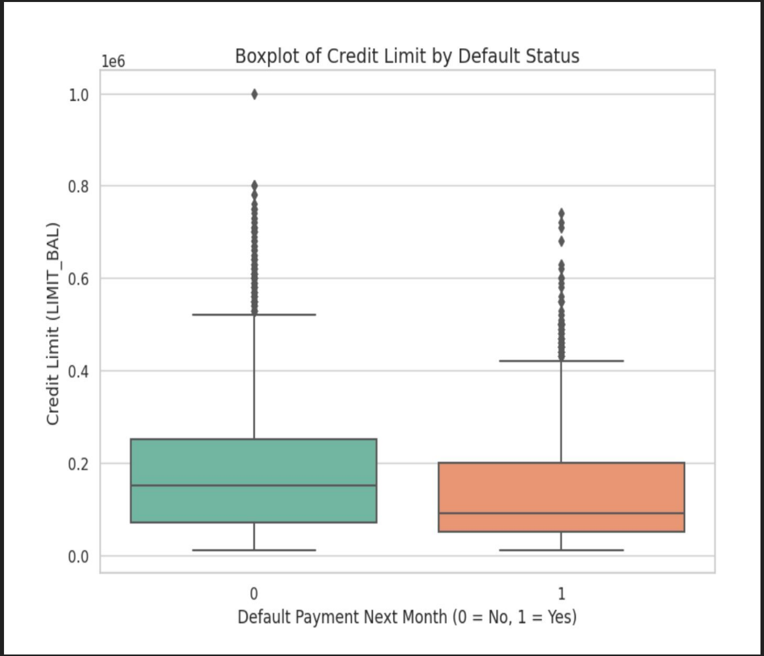
Exploratory Data Analysis(Contd.)

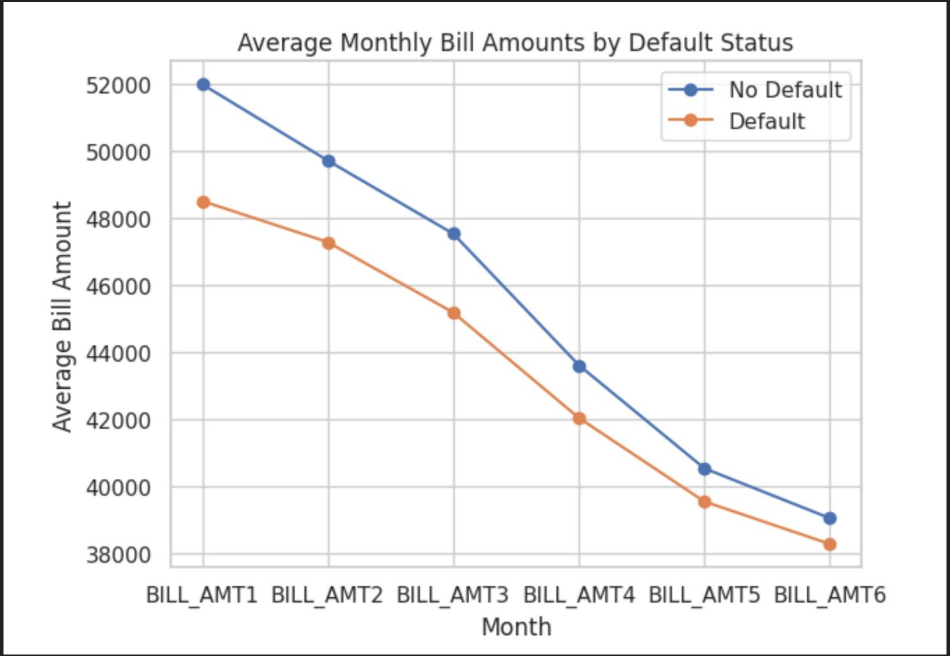




	LIMIT_BAL	AGE	BILL_AMT1	BILL_AMT2	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2	PAY_AMT3	PAY_AMT4	PAY_AMT5	PAY_AMT6	PAY_1	PAY_2	PAY_3	PAY_4	PAY_5	PAY_6	efault.payment.next.month
LIMIT_BAL	1.00	0.14	0.29	0.28	0.28	0.29	0.30	0.29	0.20	0.18	0.21	0.20	0.22	0.22	-0.27	-0.30	-0.29	-0.27	-0.25	-0.24	-0.15
AGE	0.14	1.00	0.06	0.05	0.05	0.05	0.05	0.05	0.03	0.02	0.03	0.02	0.02	0.02	-0.04	-0.05	-0.05	-0.05	-0.05	-0.05	0.01
BILL_AMT1	0.29	0.06	1.00	0.95	0.89	0.86	0.83	0.80	0.14	0.10	0.16	0.16	0.17	0.18	0.19	0.23	0.21	0.20	0.21	0.21	-0.02
BILL_AMT2	0.28	0.05	0.95	1.00	0.93	0.89	0.86	0.83	0.28	0.10	0.15	0.15	0.16	0.17	0.19	0.24	0.24	0.23	0.23	0.23	-0.01
BILL_AMT3	0.28	0.05	0.89	0.93	1.00	0.92	0.88	0.85	0.24	0.32	0.13	0.14	0.18	0.18	0.18	0.22	0.23	0.24	0.24	0.24	-0.01
BILL_AMT4	0.29	0.05	0.86	0.89	0.92	1.00	0.94	0.90	0.23	0.21	0.30	0.13	0.16	0.18	0.18	0.22	0.23	0.25	0.27	0.27	-0.01
BILL_AMT5	0.30	0.05	0.83	0.86	0.88	0.94	1.00	0.95	0.22	0.18	0.25	0.29	0.14	0.16	0.18	0.22	0.23	0.24	0.27	0.29	-0.01
BILL_AMT6	0.29	0.05	0.80	0.83	0.85	0.90	0.95	1.00	0.20	0.17	0.23	0.25	0.31	0.12	0.18	0.22	0.22	0.24	0.26	0.29	-0.01
PAY_AMT1	0.20	0.03	0.14	0.28	0.24	0.23	0.22	0.20	1.00	0.29	0.25	0.20	0.15	0.19	-0.08	-0.08	0.00	-0.01	-0.01	-0.00	-0.07
PAY_AMT2	0.18	0.02	0.10	0.10	0.32	0.21	0.18	0.17	0.29	1.00	0.24	0.18	0.18	0.16	-0.07	-0.06	-0.07	-0.00	-0.00	-0.01	-0.06
PAY_AMT3	0.21	0.03	0.16	0.15	0.13	0.30	0.25	0.23	0.25	0.24	1.00	0.22	0.16	0.16	-0.07	-0.06	-0.05	-0.07	0.01	0.01	-0.06
PAY_AMT4	0.20	0.02	0.16	0.15	0.14	0.13	0.29	0.25	0.20	0.18	0.22	1.00	0.15	0.16	-0.06	-0.05	-0.05	-0.04	-0.06	0.02	-0.06
PAY_AMT5	0.22	0.02	0.17	0.16	0.18	0.16	0.14	0.31	0.15	0.18	0.16	0.15	1.00	0.15	-0.06	-0.04	-0.04	-0.03	-0.03	-0.05	-0.06
PAY_AMT6	0.22	0.02	0.18	0.17	0.18	0.18	0.16	0.12	0.19	0.16	0.16	0.16	0.15	1.00	-0.06	-0.04	-0.04	-0.03	-0.02	-0.03	-0.05
PAY_1	-0.27	-0.04	0.19	0.19	0.18	0.18	0.18	0.18	-0.08	-0.07	-0.07	-0.06	-0.06	-0.06	1.00	0.67	0.57	0.54	0.51	0.47	0.32
PAY_2	-0.30	-0.05	0.23	0.24	0.22	0.22	0.22	0.22	-0.08	-0.06	-0.06	-0.05	-0.04	-0.04	0.67	1.00	0.77	0.66	0.62	0.58	0.26
PAY_3	-0.29	-0.05	0.21	0.24	0.23	0.23	0.23	0.22	0.00	-0.07	-0.05	-0.05	-0.04	-0.04	0.57	0.77	1.00	0.78	0.69	0.63	0.24
PAY_4	-0.27	-0.05	0.20	0.23	0.24	0.25	0.24	0.24	-0.01	-0.00	-0.07	-0.04	-0.03	-0.03	0.54	0.66	0.78	1.00	0.82	0.72	0.22
PAY_5	-0.25	-0.05	0.21	0.23	0.24	0.27	0.27	0.26	-0.01	-0.00	0.01	-0.06	-0.03	-0.02	0.51	0.62	0.69	0.82	1.00	0.82	0.20
PAY_6	-0.24	-0.05	0.21	0.23	0.24	0.27	0.29	0.29	-0.00	-0.01	0.01	0.02	-0.05	-0.03	0.47	0.58	0.63	0.72	0.82	1.00	0.19
ault.payment.next.month	-0.15	0.01	-0.02	-0.01	-0.01	-0.01	-0.01	-0.01	-0.07	-0.06	-0.06	-0.06	-0.06	-0.05	0.32	0.26	0.24	0.22	0.20	0.19	1.00

Exploratory Data Analysis(Contd.)





ML PipeLine

- Data Preprocessing
- Separate Features & Target and Train/Test/Validation Split
- Feature Selection and Scaling
- Handle Class Imbalance
- Model Training and Evaluation
- Model Performance Summary
- Implementing the best model on Testset
- Conclusion

Model Training and Evaluation

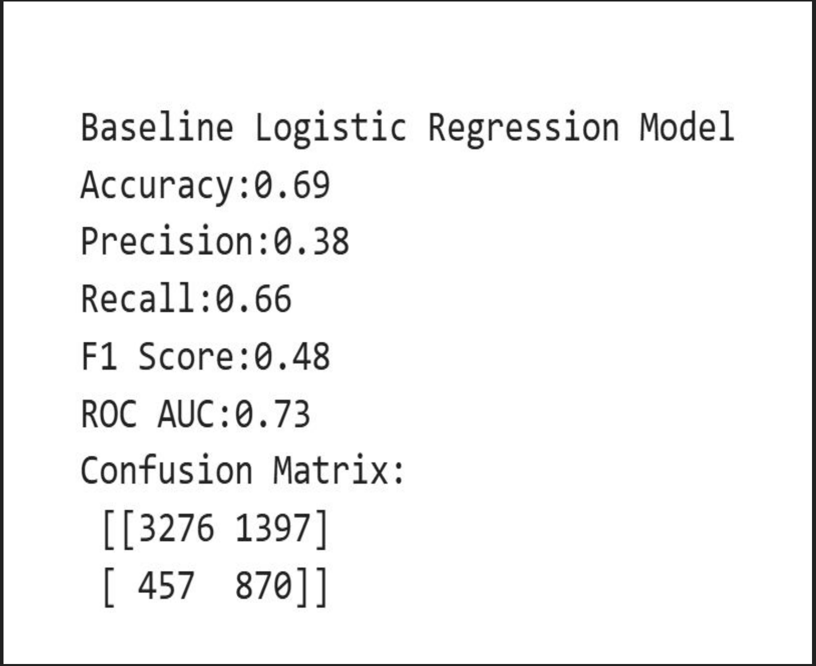
Default Credit Card Clients project has been trained on the following 4 Supervised learning algorithms:

- Logistic Regression
- Random Forest Classifier
- XGBoost
- SVC

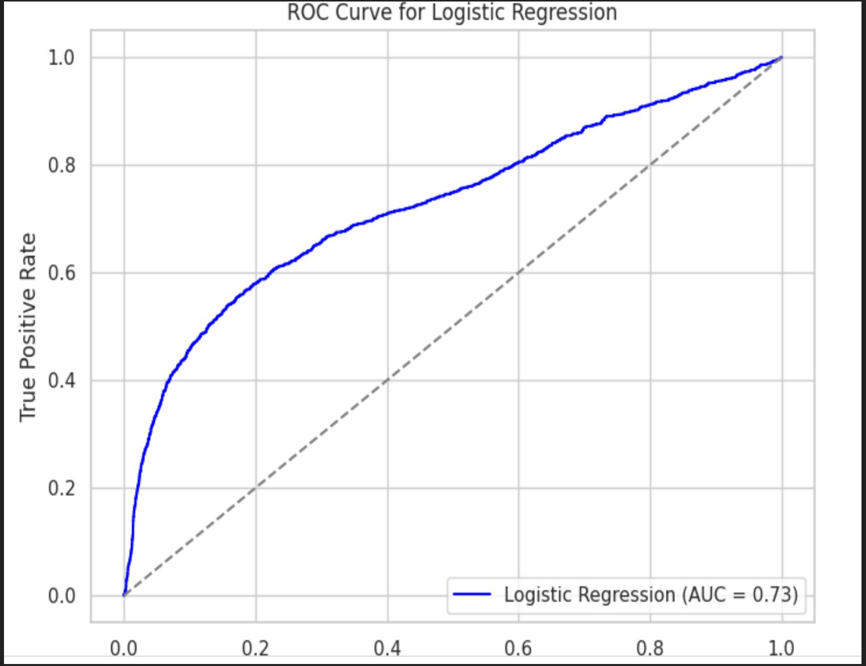
Model Optimization Summary

- Applied hyperparameter tuning to each model using GridSearch/Randomized CV to identify optimal settings.
- Select the best threshold value for classification and recalculated accuracy.
- Preformed Cross-validation on each model to compute the mean F1 score ensuring robust performance evaluation.
- Compared tuned models against baseline versions to assess improvements in recall,F1 score and ROC-AUC.
- Visualized performance metrics to support final model selection.

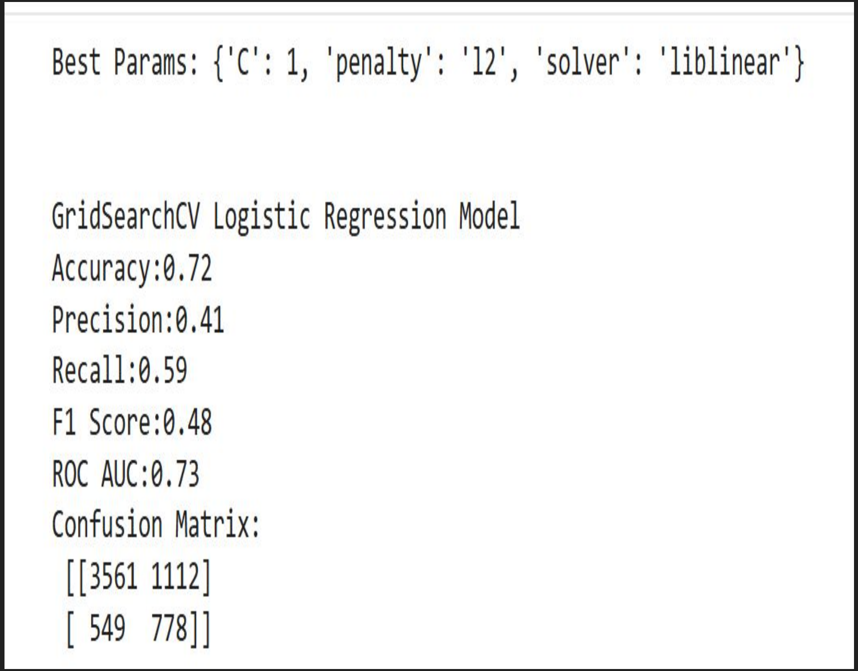
Logistic Regression(Baseline Model)



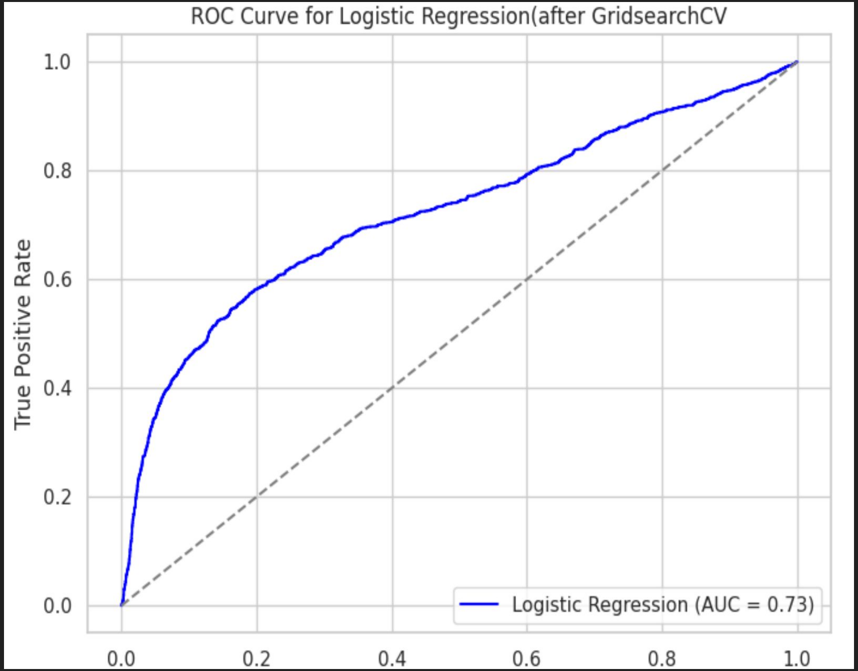
```
Baseline Logistic Regression Model
Accuracy:0.69
Precision:0.38
Recall:0.66
F1 Score:0.48
ROC AUC:0.73
Confusion Matrix:
 [[3276 1397]
 [ 457  870]]
```



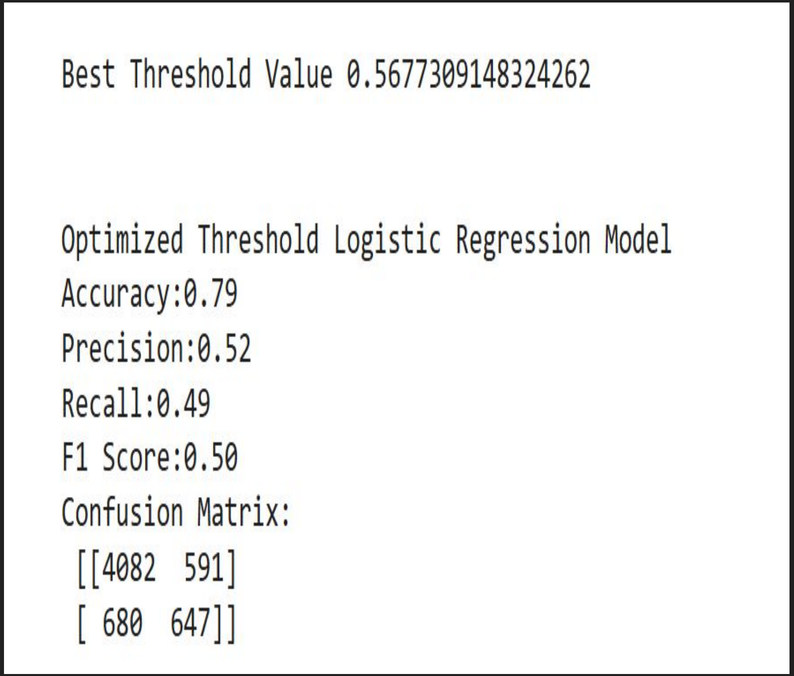
Logistic Regression (HyperTuned Model)



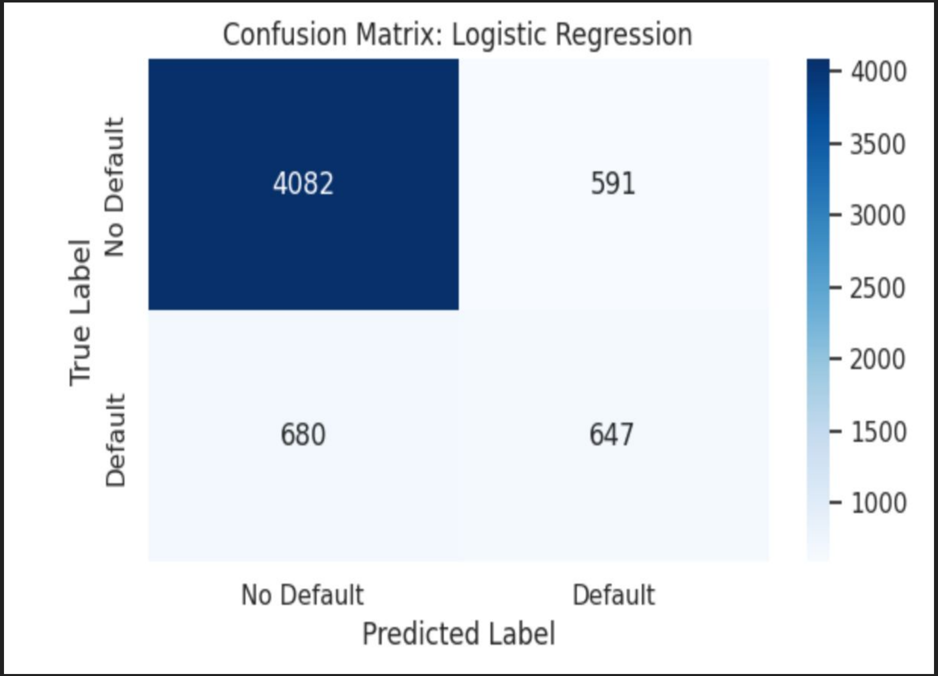
```
Best Params: {'C': 1, 'penalty': 'l2', 'solver': 'liblinear'}



GridSearchCV Logistic Regression Model
Accuracy:0.72
Precision:0.41
Recall:0.59
F1 Score:0.48
ROC AUC:0.73
Confusion Matrix:
 [[3561 1112]
 [ 549  778]]
```



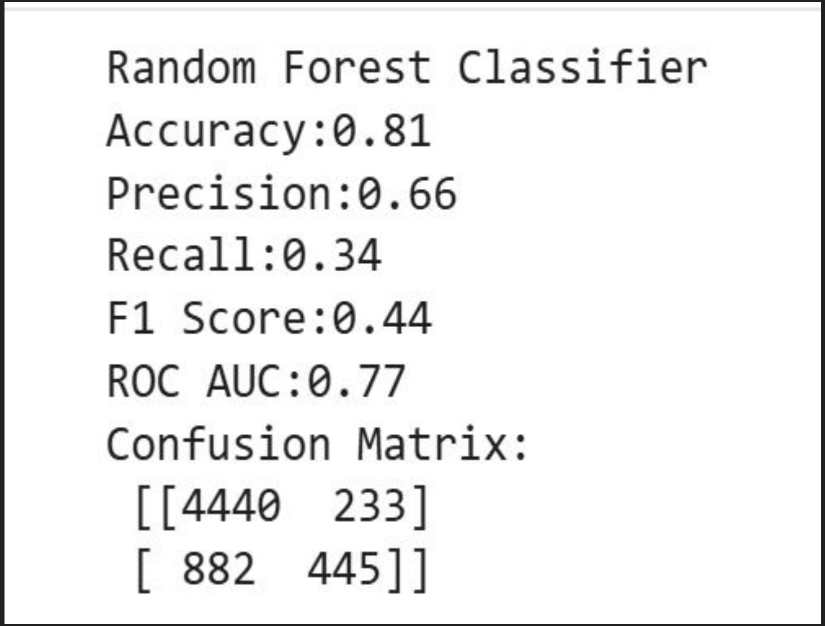
Logistic Regression(Best Threshold)



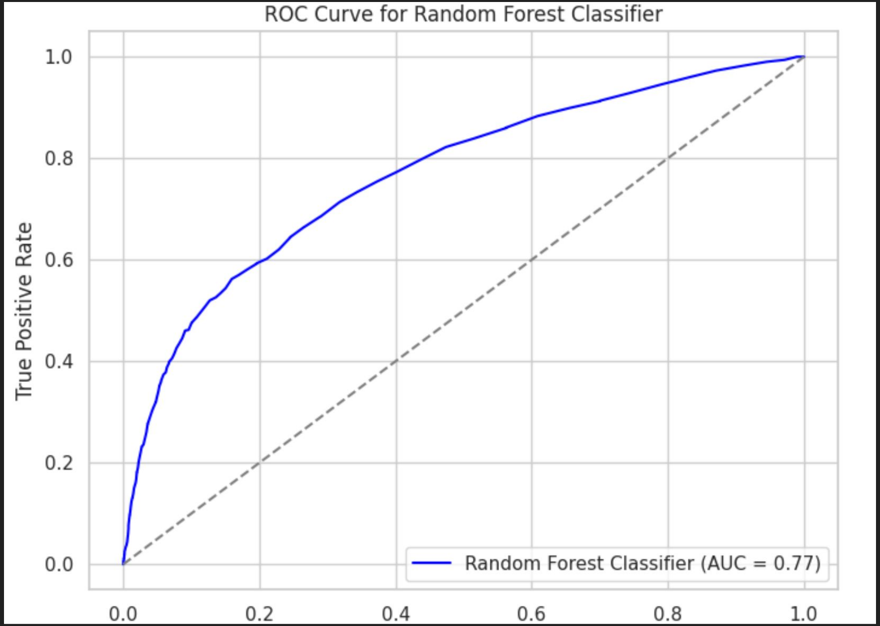
```
Best Threshold Value 0.5677309148324262



Optimized Threshold Logistic Regression Model
Accuracy:0.79
Precision:0.52
Recall:0.49
F1 Score:0.50
Confusion Matrix:
 [[4082  591]
 [ 680  647]]
```



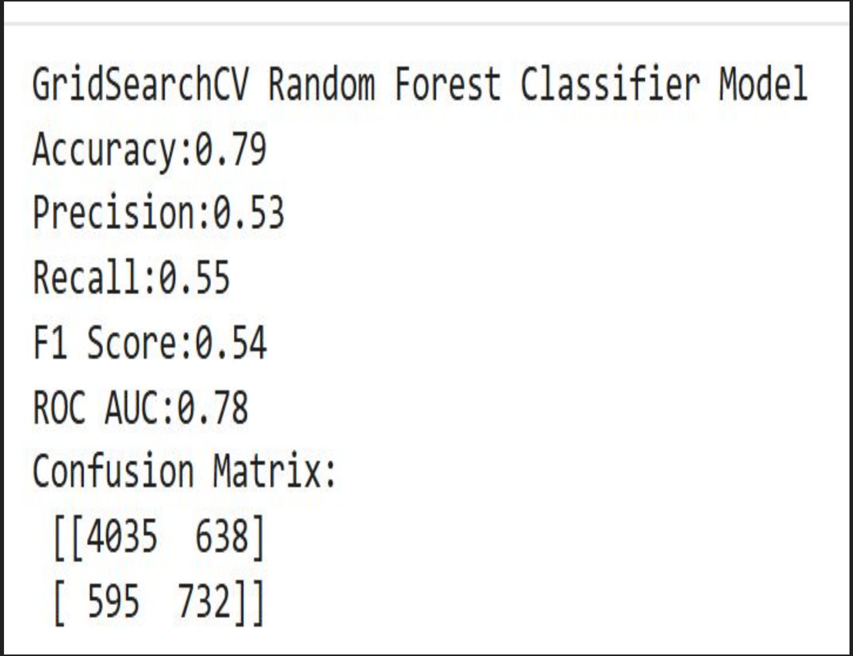
Random Forest Classifier(Baseline Model)



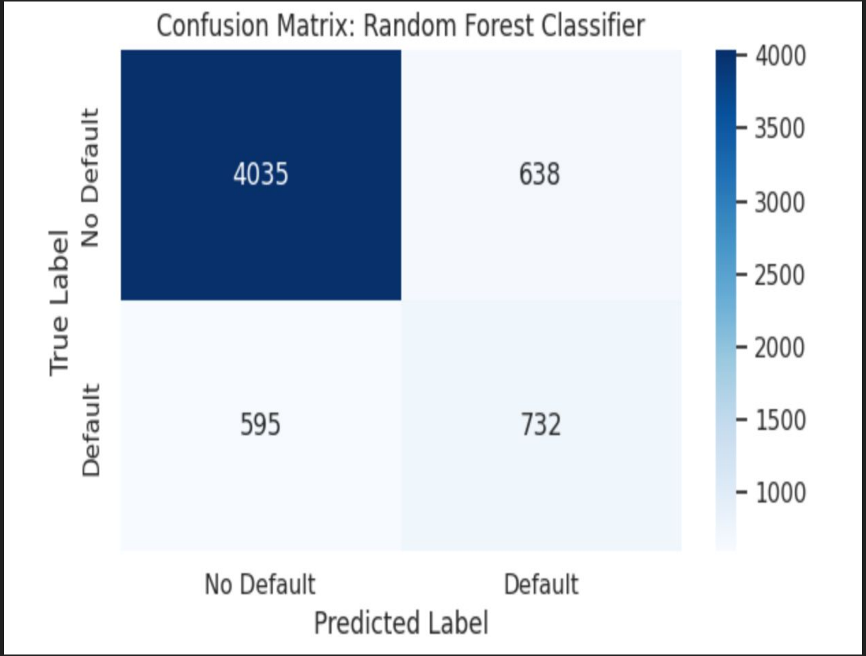
```
Random Forest Classifier
Accuracy:0.81
Precision:0.66
Recall:0.34
F1 Score:0.44
ROC AUC:0.77
Confusion Matrix:
 [[4440  233]
 [ 882  445]]
```



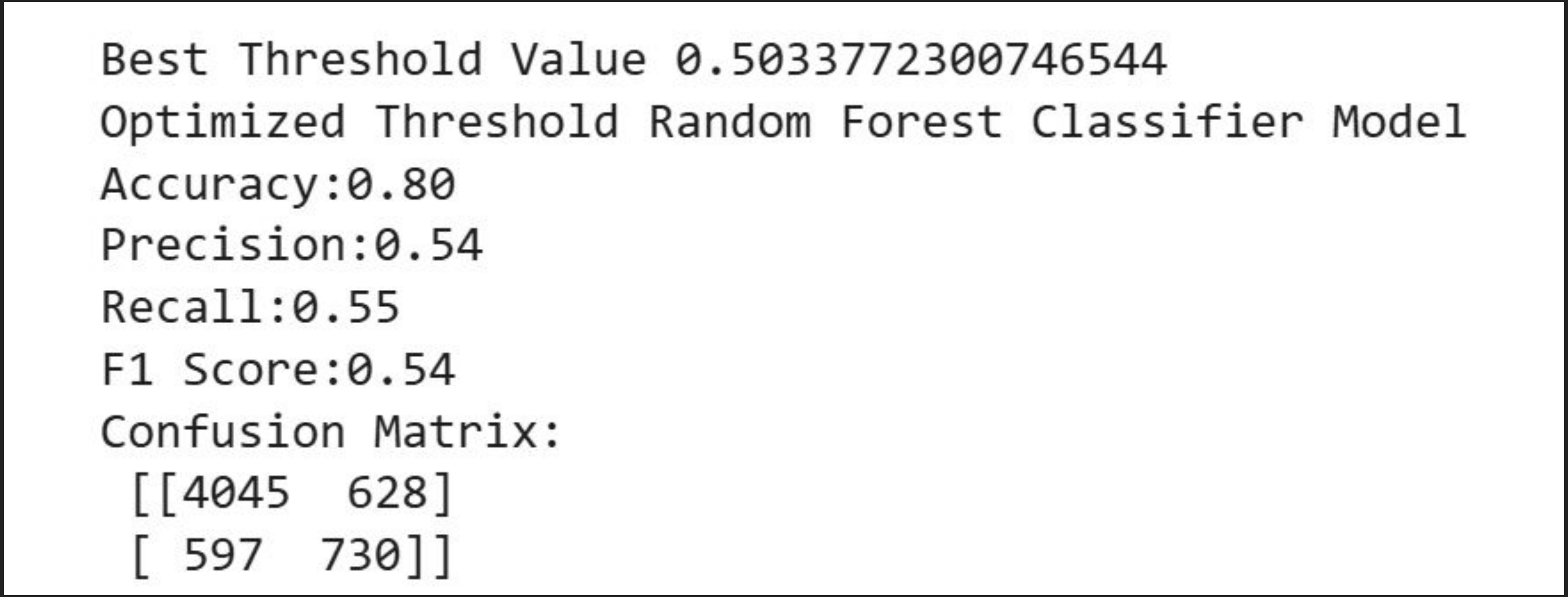
Random Forest Classifier(HyperTuned Model)



```
GridSearchCV Random Forest Classifier Model
Accuracy:0.79
Precision:0.53
Recall:0.55
F1 Score:0.54
ROC AUC:0.78
Confusion Matrix:
 [[4035  638]
 [ 595  732]]
```

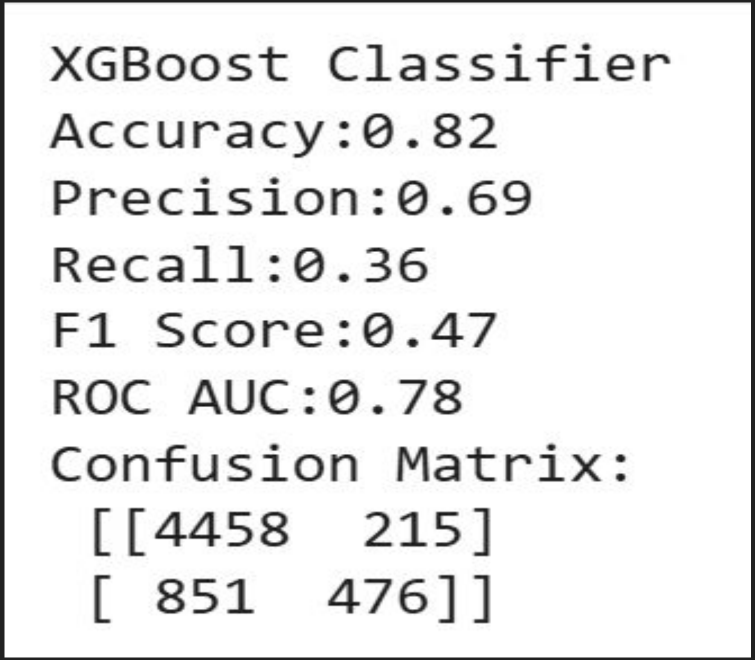


Random Forest Classifier(Best Threshold)

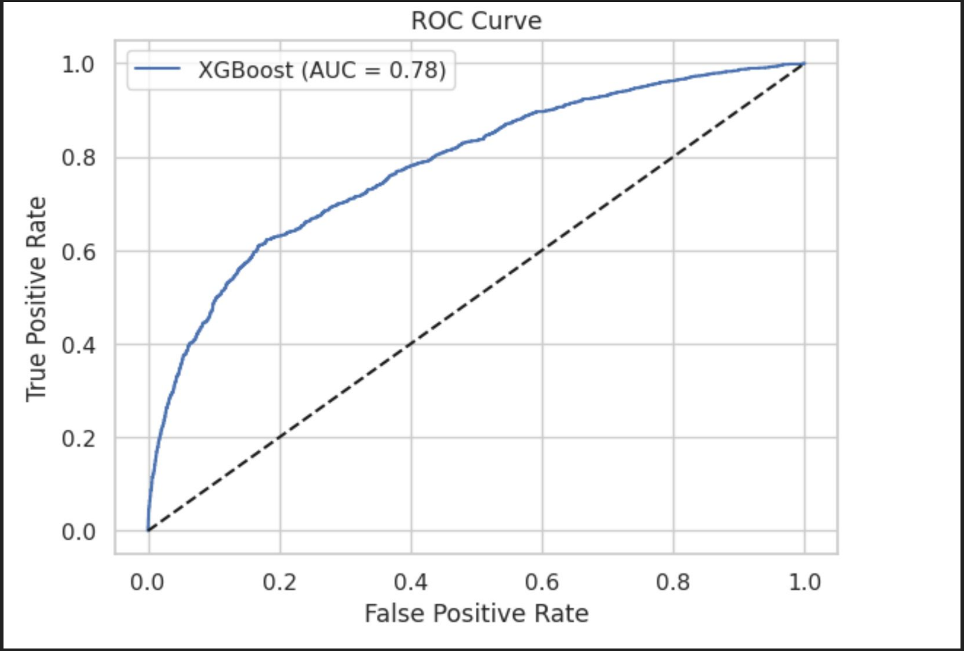


```
Best Threshold Value 0.5033772300746544
Optimized Threshold Random Forest Classifier Model
Accuracy:0.80
Precision:0.54
Recall:0.55
F1 Score:0.54
Confusion Matrix:
 [[4045  628]
 [ 597  730]]
```

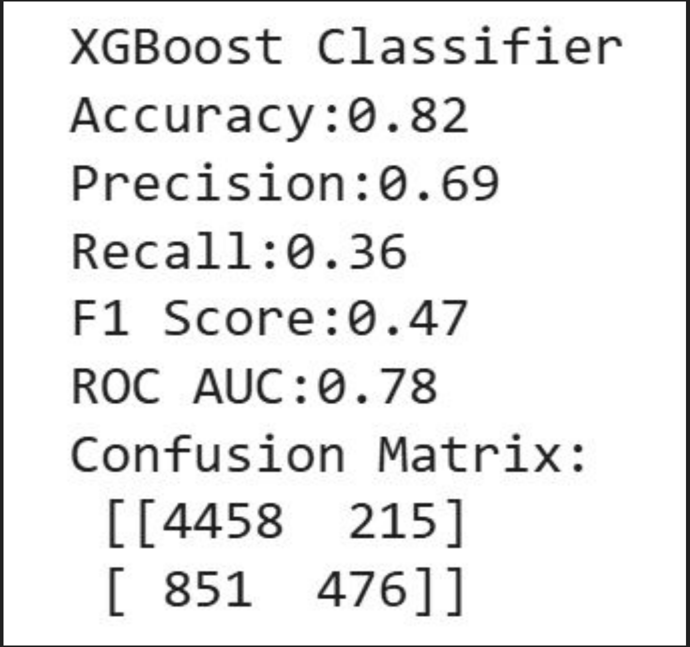
XGBoost (Baseline Model)



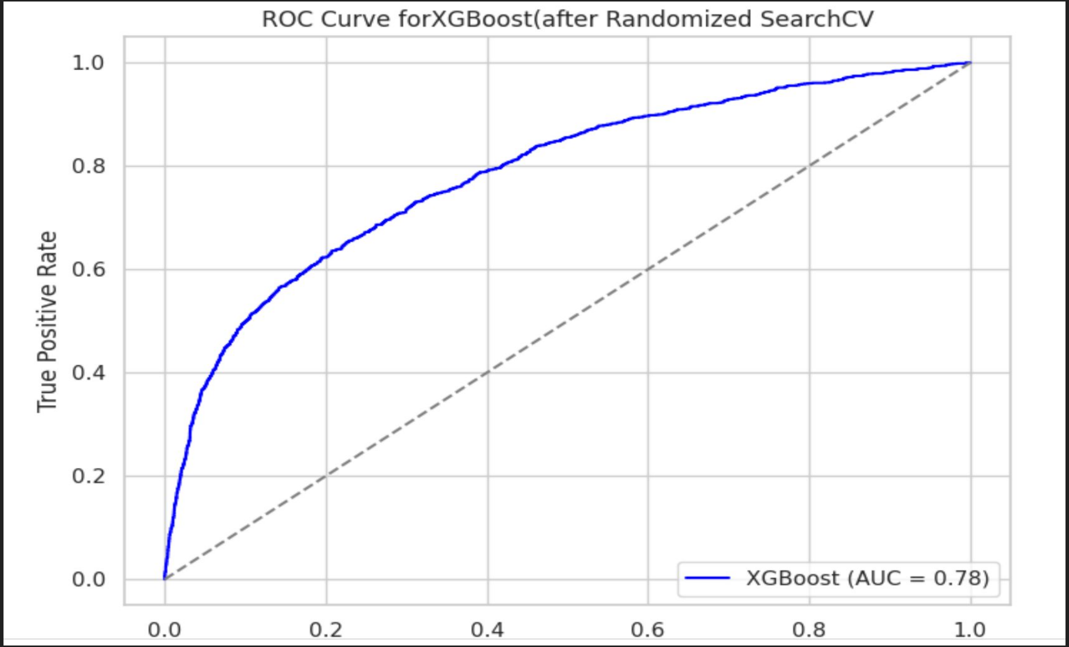
```
XGBoost Classifier
Accuracy:0.82
Precision:0.69
Recall:0.36
F1 Score:0.47
ROC AUC:0.78
Confusion Matrix:
 [[4458  215]
 [ 851  476]]
```



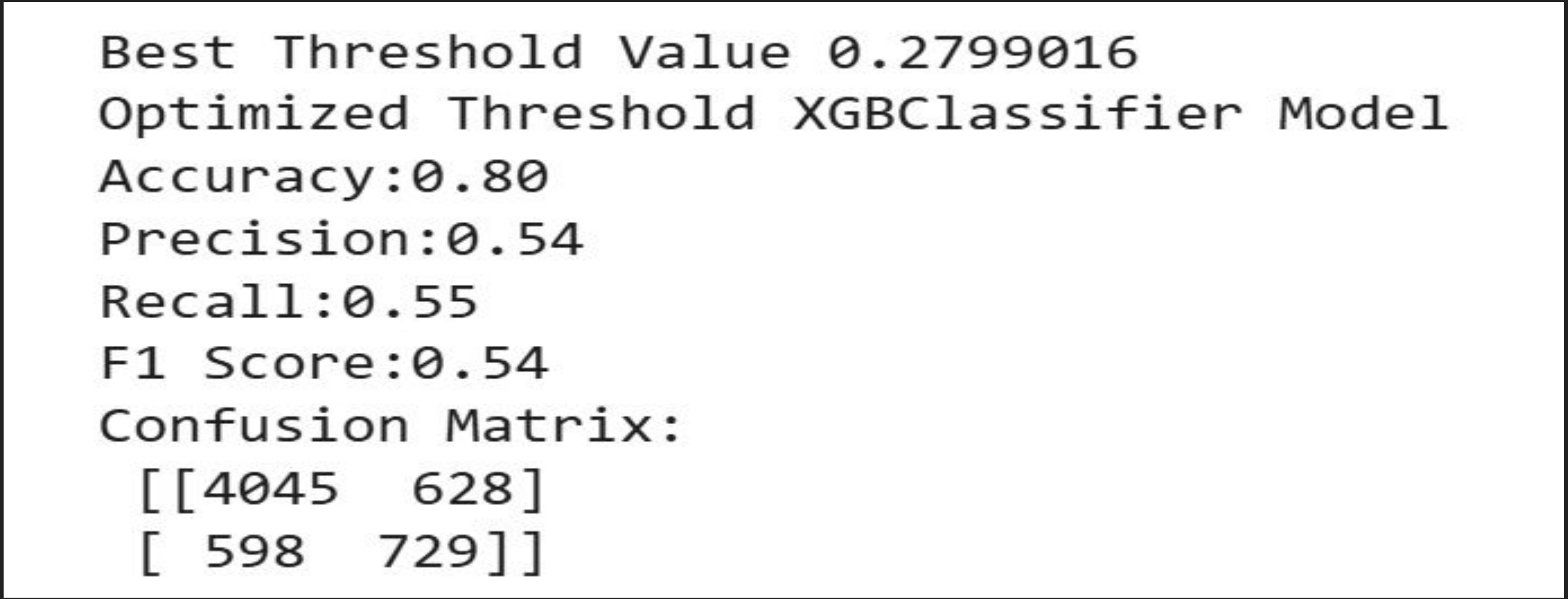
XGBoost (HyperTuned Model)



```
XGBoost Classifier
Accuracy:0.82
Precision:0.69
Recall:0.36
F1 Score:0.47
ROC AUC:0.78
Confusion Matrix:
 [[4458  215]
 [ 851  476]]
```

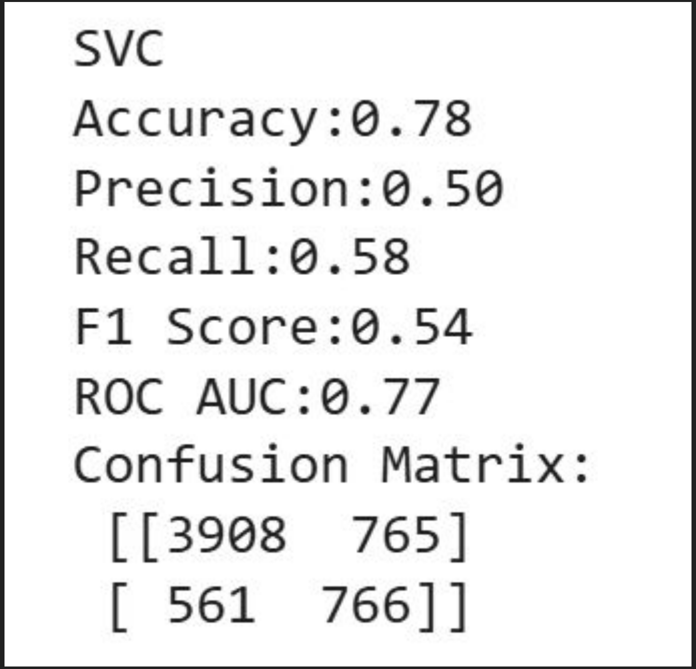


XGBoost (Best Threshold)

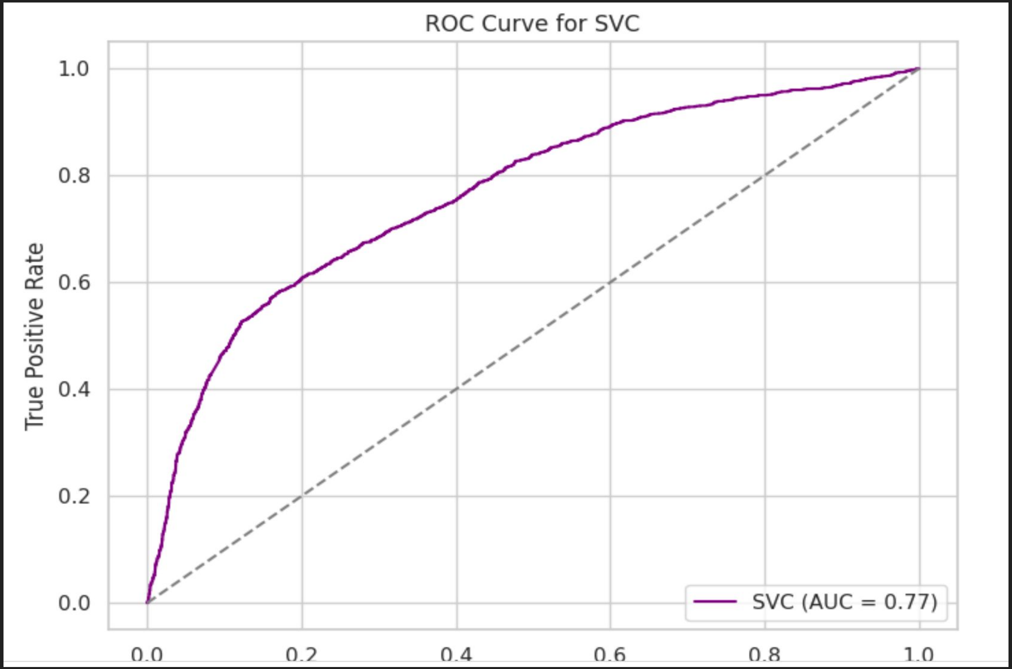


```
Best Threshold Value 0.2799016
Optimized Threshold XGBClassifier Model
Accuracy:0.80
Precision:0.54
Recall:0.55
F1 Score:0.54
Confusion Matrix:
 [[4045  628]
 [ 598  729]]
```

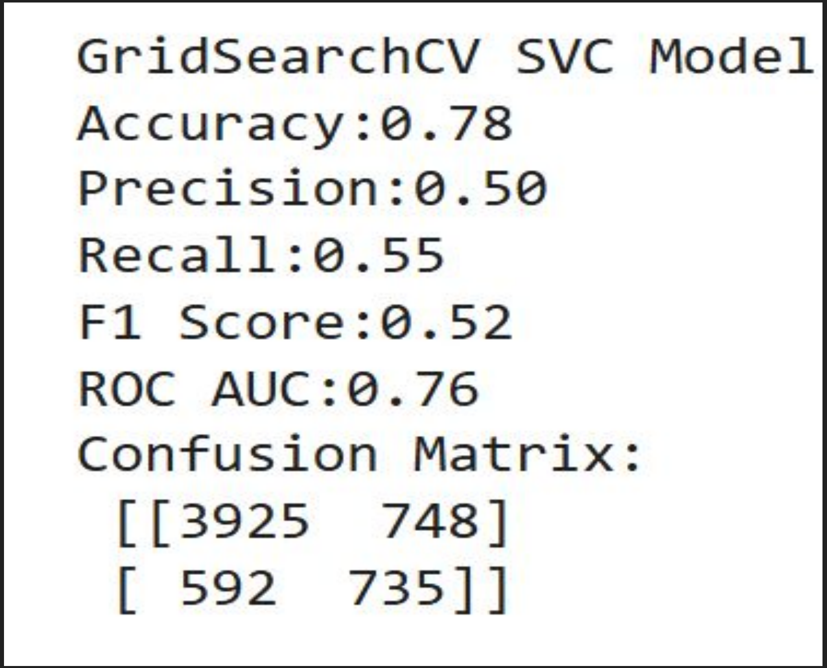
SVC(Baseline Model)



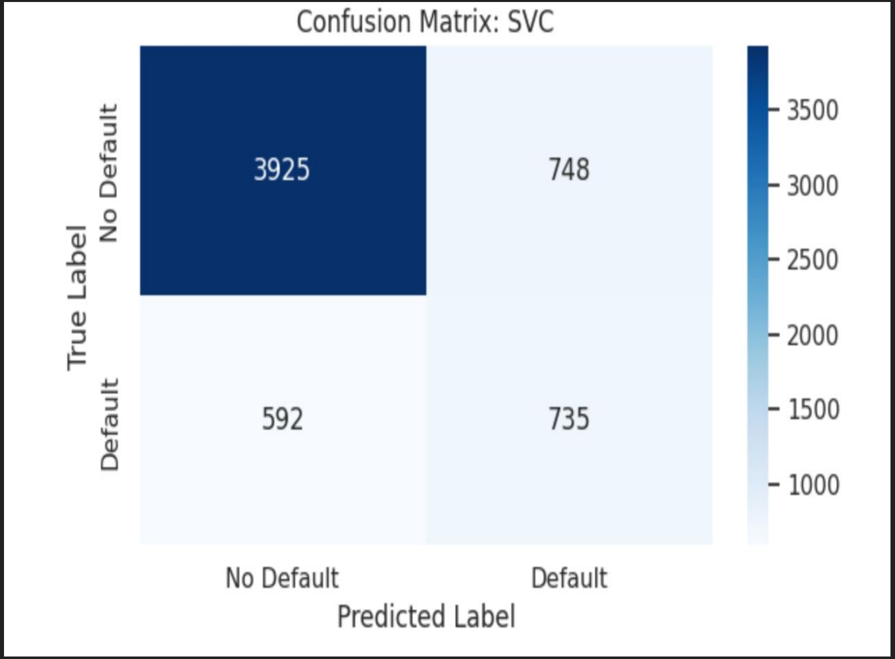
```
SVC
Accuracy:0.78
Precision:0.50
Recall:0.58
F1 Score:0.54
ROC AUC:0.77
Confusion Matrix:
 [[3908  765]
 [ 561  766]]
```



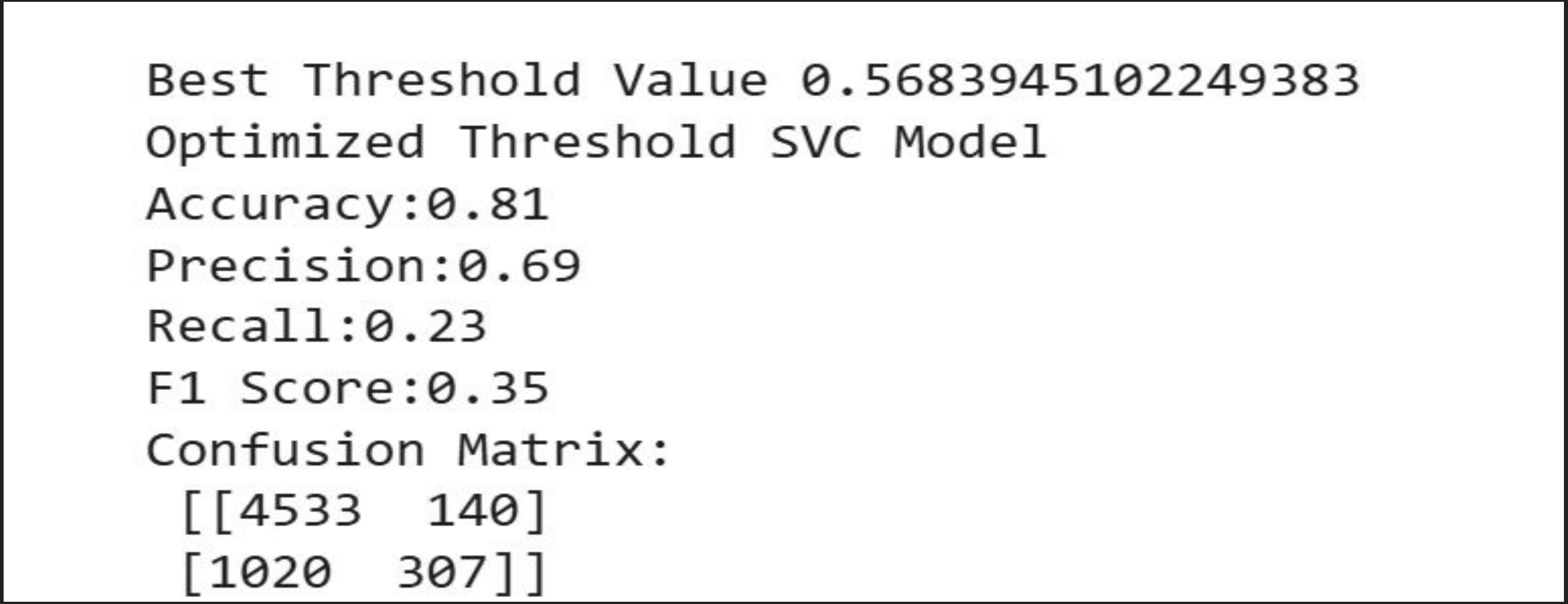
SVC(HyperTuned Model)



```
GridSearchCV SVC Model
Accuracy:0.78
Precision:0.50
Recall:0.55
F1 Score:0.52
ROC AUC:0.76
Confusion Matrix:
 [[3925  748]
 [ 592  735]]
```

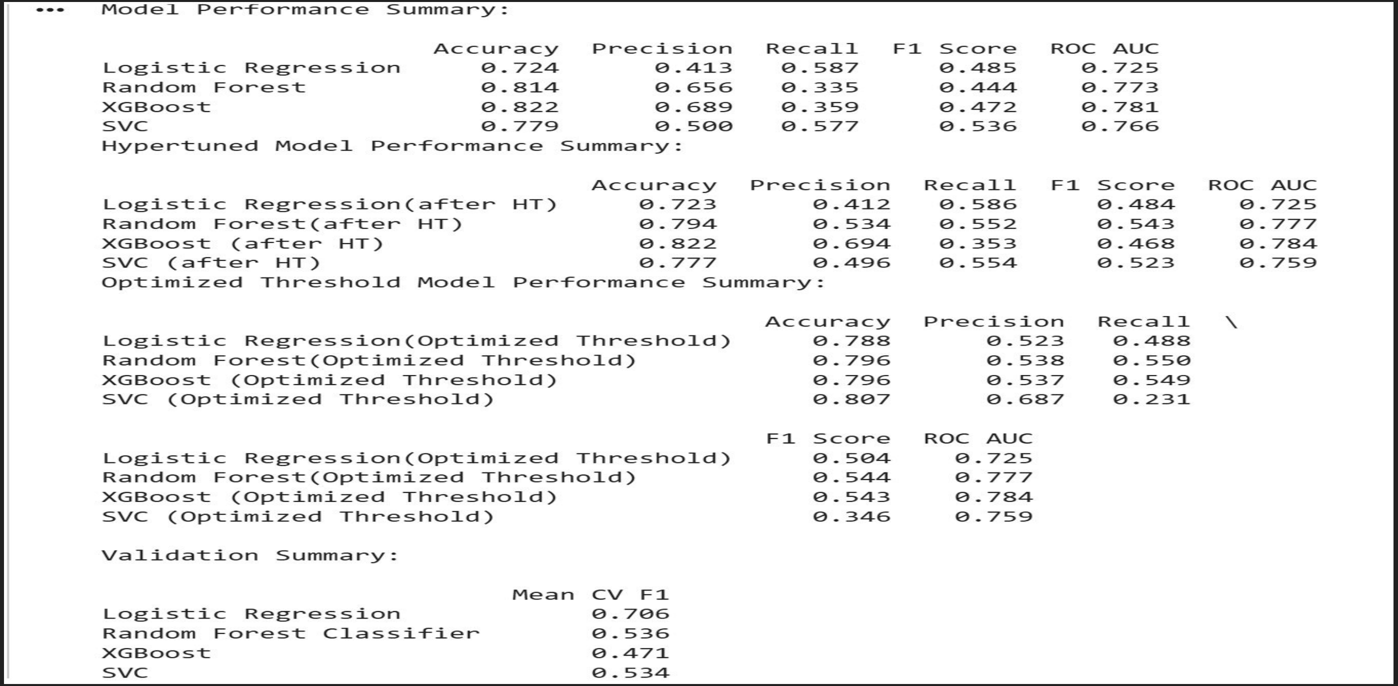


SVC(Best Threshold)



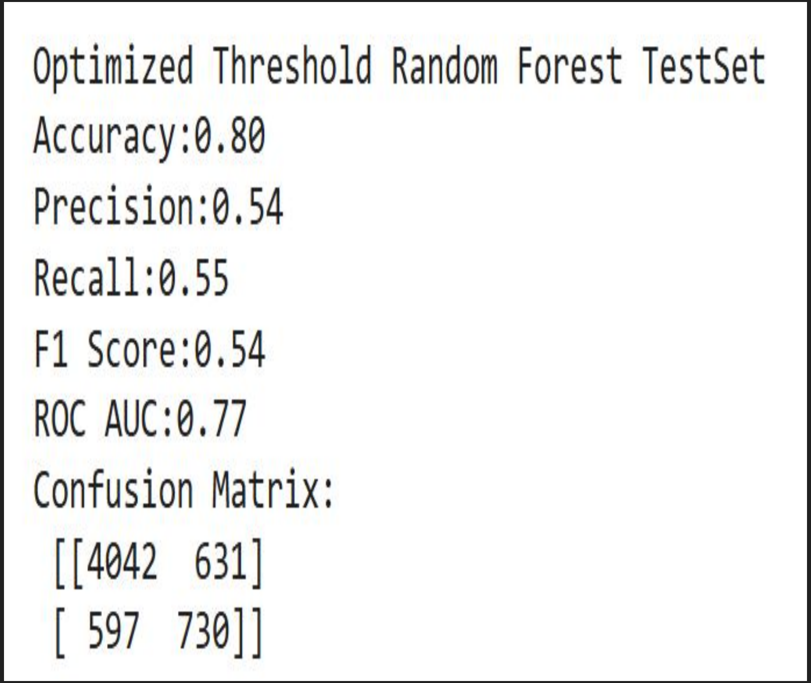
```
Best Threshold Value 0.5683945102249383
Optimized Threshold SVC Model
Accuracy:0.81
Precision:0.69
Recall:0.23
F1 Score:0.35
Confusion Matrix:
 [[4533  140]
 [1020  307]]
```

Model Performance Summary

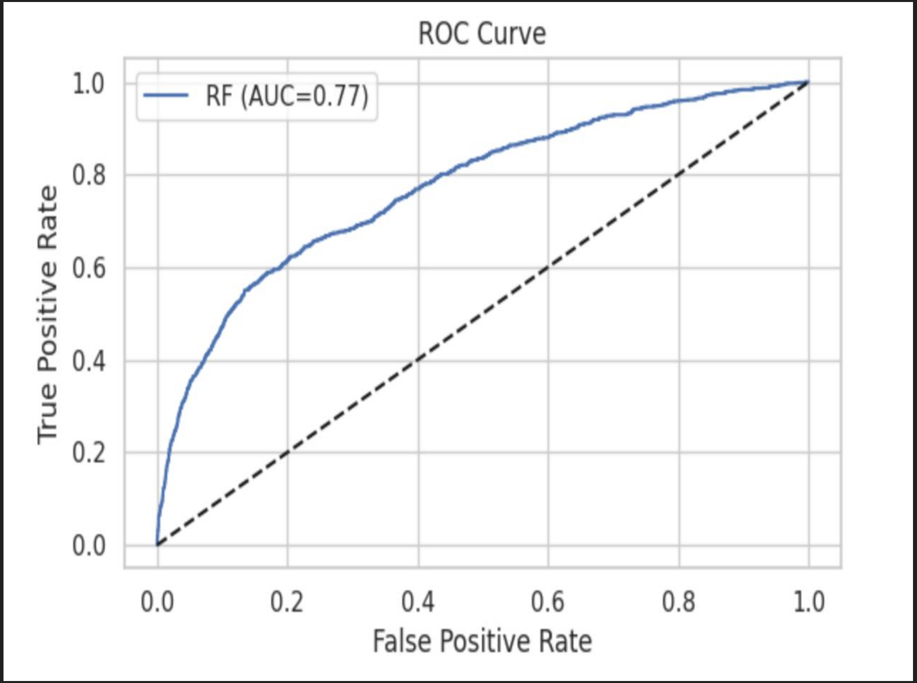


```
Model Performance Summary:

                     Accuracy  Precision  Recall  F1 Score  ROC AUC
Logistic Regression     0.724      0.413   0.587     0.485    0.725
Random Forest           0.814      0.656   0.335     0.444    0.773
XGBoost                 0.822      0.689   0.359     0.472    0.781
SVC                     0.779      0.500   0.577     0.536    0.766
Hypertuned Model Performance Summary:

                               Accuracy  Precision  Recall  F1 Score  ROC AUC
Logistic Regression(after HT)     0.723      0.412   0.586     0.484    0.725
Random Forest(after HT)           0.794      0.534   0.552     0.543    0.777
XGBoost (after HT)                0.822      0.694   0.353     0.468    0.784
SVC (after HT)                    0.777      0.496   0.554     0.523    0.759
Optimized Threshold Model Performance Summary:

                                          Accuracy  Precision  Recall  \
Logistic Regression(Optimized Threshold)     0.788      0.523   0.488
Random Forest(Optimized Threshold)           0.796      0.538   0.550
XGBoost (Optimized Threshold)                0.796      0.537   0.549
SVC (Optimized Threshold)                    0.807      0.687   0.231

                                          F1 Score  ROC AUC
Logistic Regression(Optimized Threshold)     0.504    0.725
Random Forest(Optimized Threshold)           0.544    0.777
XGBoost (Optimized Threshold)                0.543    0.784
SVC (Optimized Threshold)                    0.346    0.759

Validation Summary:

                          Mean CV F1
Logistic Regression            0.706
Random Forest Classifier       0.536
XGBoost                        0.471
SVC                            0.534
```

Implementing the Best Model on the Test set - RFC(Optimized Threshold)



```
Optimized Threshold Random Forest TestSet
Accuracy:0.80
Precision:0.54
Recall:0.55
F1 Score:0.54
ROC AUC:0.77
Confusion Matrix:
 [[4042  631]
 [ 597  730]]
```



Conclusion

Key Learnings:

1)Threshold Optimization is essential for imbalanced datasets.Optimizing threshold(0.5) for Random Forest and XGBOOST significantly improve their recall and F1 scores.

2)Other Metrics like F1 Score,Recall , Precision,ROC AUC should be taken for imbalance data sets other than accuracy.

3)Cross Validation reveals true model stability. Though Logistic Regression has the highest CV (0.7) it lacked in the other metrics. XGBOOST has the lowest CV score 0.4 showing it is more sensitive to data splits. Random Forest showed strong performance.

4)Ensemble models like XGBOOST and Random Forest showed better performance than the other 2 models.

5)Confusion Matrix Importance should be given to False positives and False negatives for this imbalanced dataset than accuracy.

Highlights:

1)EDA has been done on the data set and found that the class is imbalanced.So, SMOTE and class weight='balanced' techniques were used.

2)Data preprocessing and scaling were done on the numerical and categorical columns.

3)Evaluated models across Baseline,Hyperparameter Tuning , Optimized threshold and Cross Validation.

4)Best Overall Model: Random Forest(Optimized Threshold) Random Forest(Optimized Threshold) 0.796(accuracy),0.538(precision), 0.550(recall),0.544 (F1 SCORE) and ROC AUC (0.777)

5)Random Forest Classifier(Optimized Threshold) was applied on the test set and the below metric were calculated.

Optimized Threshold Random Forest TestSet Accuracy:0.80 Precision:0.54 Recall:0.55 F1 Score:0.54 ROC AUC:0.77
Confusion Matrix: [[4042 631] [597 730]]

6)Out of 6000 samples in the testset, we can see that 4772 samples have been predicted accurately.

Things to Improve

1)Feature Engineering - Model performance can be improved by adding more features,domain specific variables

2)Advanced Ensemble methods like LightBOOST and also other models like knn can be implemented.

3)Have to address False negatives since this is very expensive.This can be done using cost sensitive learning.

4)For improving code usability,Pipeline models can be used.

References

- https://seaborn.pydata.org/generated/seaborn.kdeplot.html
- https://www.geeksforgeeks.org/python/detecting-multicollinearity-with-vif-python/
- https://www.geeksforgeeks.org/maths/chi-square-test/
- https://github.com/HillidatulIlmi/EDA-and-Machine-Learning-Default-of-Credit-Card-Clients/blob/main/EDA_and_Machine_Learning_Default_of_Credit_Card_Clients.ipynb

Links

Github Link:

MS-AI/Pathway Machine Learning: Theory & Hands-On Practice with Python Specialization/CSCA 5622: Introduction to Machine Learning: Supervised Learning /Module 7 - Final Project at main · kmadhu181090/MS-AI

Video Presentation Link:

https://youtu.be/eFIQEvEVZxM

THANK YOU