Customer Churn Prediction

GROUP-13

TEAM MEMBERS	ROLL NO
Anu Prasad	AM.EN.U4AIE22105
Aparna S	AM.EN.U4AIE22106
Bhamini R	AM.EN.U4AIE22111
Krishna Prabha S	AM.EN.U4AIE22129

Abstract:

Churn prediction is an important activity for any business focused on the retention process for customer loyalty. The primary objective of this study is to analyze these sectors of customer churning using various techniques of supervised learning, such as banking, credit card servicing, and telecommunications. We use a comprehensive dataset from Kaggle about customer demographic information, transaction history, and service usage patterns. In this research, we will apply the following predictive models: Logistic Regression, Random Forest, XGBoost, and SVM. The performance of these models will, in turn, be evaluated and compared based on some key metrics like accuracy, precision, recall, and the F1-score. The results indicate that both Random Forest and XGBoost outperform the others, thus giving useful insights to businesses to design relevant retention strategies. The study, therefore, underlies sophisticated machine learning algorithms relevant for the low rate of attrition and emphasizes the need for predictive analytics-tailored interventions.

Introduction

Churn refers to a situation whereby a customer stops doing business with a company. Churn is an issue most large companies are concerned about because retaining old customers is usually easier and cheaper than having to find new ones. Such forewarning of customers at risk can be acted upon through preventive measures to retain the customer, therefore

enhancing customer satisfaction and loyalty.

This report is established in the attempt to establish how machine learning could be helpful in cases outlining customer churn in some industries, such as banking, credit card, and telecom. Datasets are pulled from Kaggle concerning the above-mentioned industries, and different information contains all sorts of useful data such as customers' demographics, their transaction histories, and how they use the services.

We arrive at the best machine learning methods to predict churn through this. Afterward, we see the different techniques that Logistic Regression, Random Forest, XGBoost, and Support Vector Machines are made up of. We compare these techniques to arrive at the most accurate model for each dataset and give insights on the best way to predict customer churn.

The outcome of this research finally proposes that the businesses understand the churn in the customers and do something to stop them from leaving. Advanced analytics tools are required to address churn proactively and formulate better strategies so the customer retention will increase, thus increasing revenues. This report does not only contain technical discussions around the aspect of churn prediction; it also tries to highlight several practical implications churn prediction has for the businesses.

Literature Review

In our quest to figure out effective customer churn prediction models, we opted for various supervised learning algorithms to check their performance in detecting customer churn. Logistic Regression, one of the most in-demand models, was used as it is so easy and simple. Logistic regression is a model that learns from the probability of the customer churning by modelling the relationship between the dependent variable, 'churn,' and the independent ones, such as demographics, transaction history, and service usage. Logistic Regression not only produces probabilities via logistic function but also gives insight into the role of variables in the churn process, thus leading to data-based recommendations, which will improve customer retention.

In addition, we also obtained more advanced models such as XGBoost, SVM, and RF. XGBoost is a popular machine learning method that hails from a complicated neural network structure. The device gets good results when weak learners help decision trees in a sequence all combining to make a good predictive model. It reduces loss functions and overfitting through the use of regularization techniques. SVMs are good at classifying data by finding the best hyperplane to separate churners and non-churners, and they work well in high-dimensional spaces and can handle complex data distributions. Random Forest, another ensemble method, is created by the construction of multiple decision trees and the use of their collective outputs to facilitate better generalization and thus protect against overfitting. These models, each with its own unique strengths, enabled us to grasp the intricate connections among the data, thus they are a must for making accuracy customer churn predictions.

Methodology

Data Analysis:

Our project utilized of three datasets namely the Bank Customer Dataset, Telco Customer Dataset and Online Retail Customer Dataset that provided exclusive attributes helping us to screen and identify customer behaviors by numerous sectors.

Bank Customer Dataset

The Bank Customer dataset consists of 10,000 records with 14 attributes. It is comprised of 11 Numeric and 3 Categorical features. This dataset was largely used for understanding customer characteristics and behaviors regarding bank facilities. The absence of missing values meant the task of analyzing the dataset was seamless since the data was complete. Main features include customer demographics, account information, and transactional behavior. These features help in predicting customer churn.

Telco Customer Dataset

The Telco Customer dataset comprises 7,043 records with 21 attributes. The dataset contains 4 numeric and 17 categorical attributes. this dataset includes demographic details, account information, and service usage. It provides users with a holistic view of how customer interactions and services are designed across trade types i.e. the telecommunication industry, in this instance. Just like Bank Customer dataset, the values in Telco Customer dataset are also foolproof which further made the process of analysis easy for us.

Financial service industry Dataset

1,000 Records — 21 Attributes (19 Numeric, 2 Categorical). This dataset primarily focusing on customer transactions and account balances. It provides a comprehensive view of customer activity and potential churn predictors. This dataset gave us explicit insights about the customer behavior in the online retail domain. Further In, concatination we said the table and had none missing values so out of that we got a perfect outcome for our analysis.

Visual Analysis

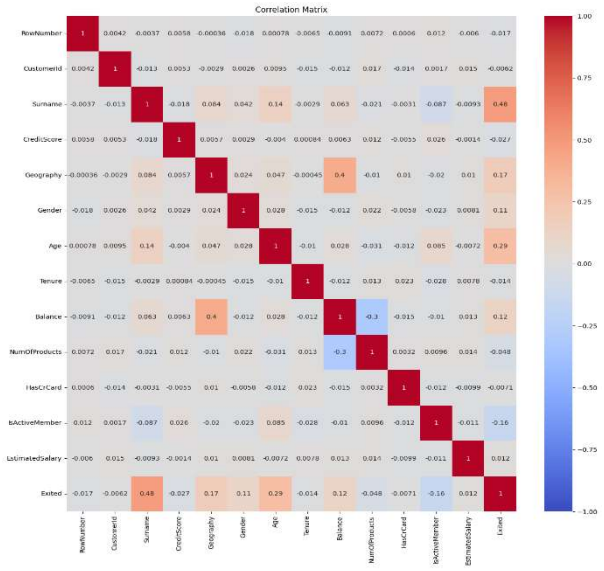
Histograms: Visual representation is done easily using histogram. Looking at this representation of numerical features helped in finding the data skewness, dispersion and central tendencies. This graph helped Pythagoras (and us too!) in understanding the spread amongst the distribution of various data points we collected learn about it.

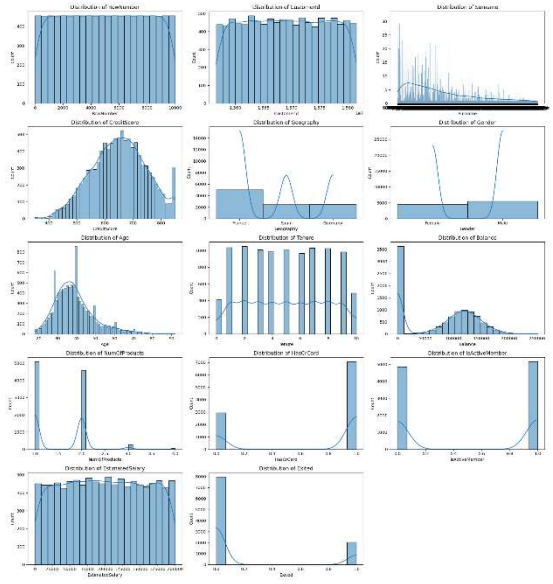
Correlation Matrices

This was done to look further in the associations b/w the numerical features and this certainly helped us in comprehending the magnitude of the relationship(not feeling logical).

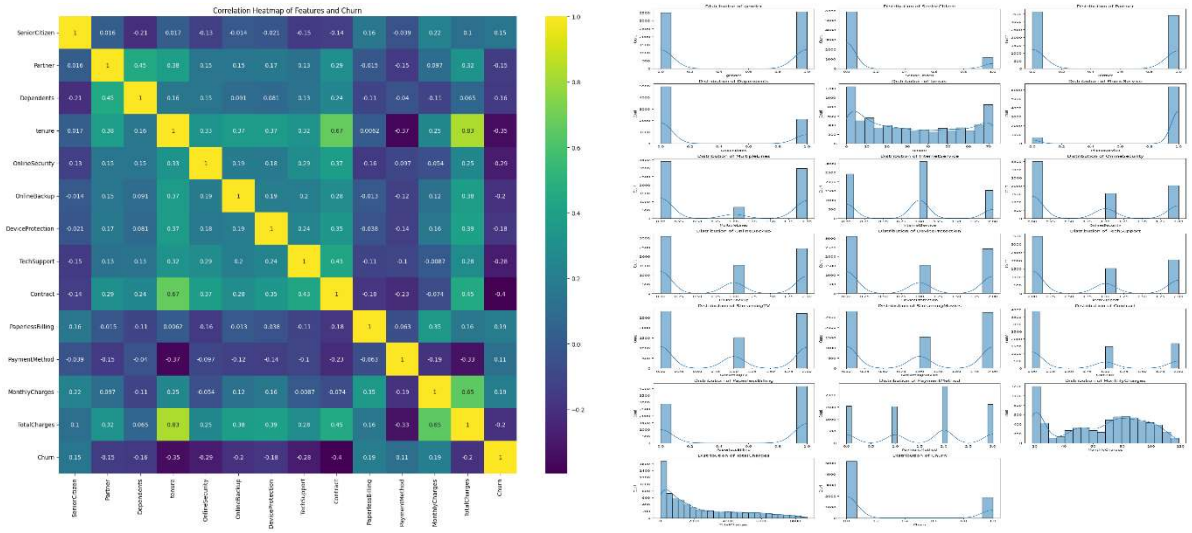
We have used correlation matrices a few times to check the relationships between the numerical attributes available. This helped us to know the features that had maximum strong relationships to make a decision in which features to be including or excluding after fitting the model because of Multicollinearity. However, by checking these we were able to take an informed decision on what attribute to keep for model training and what to be discarded.

Dataset-1

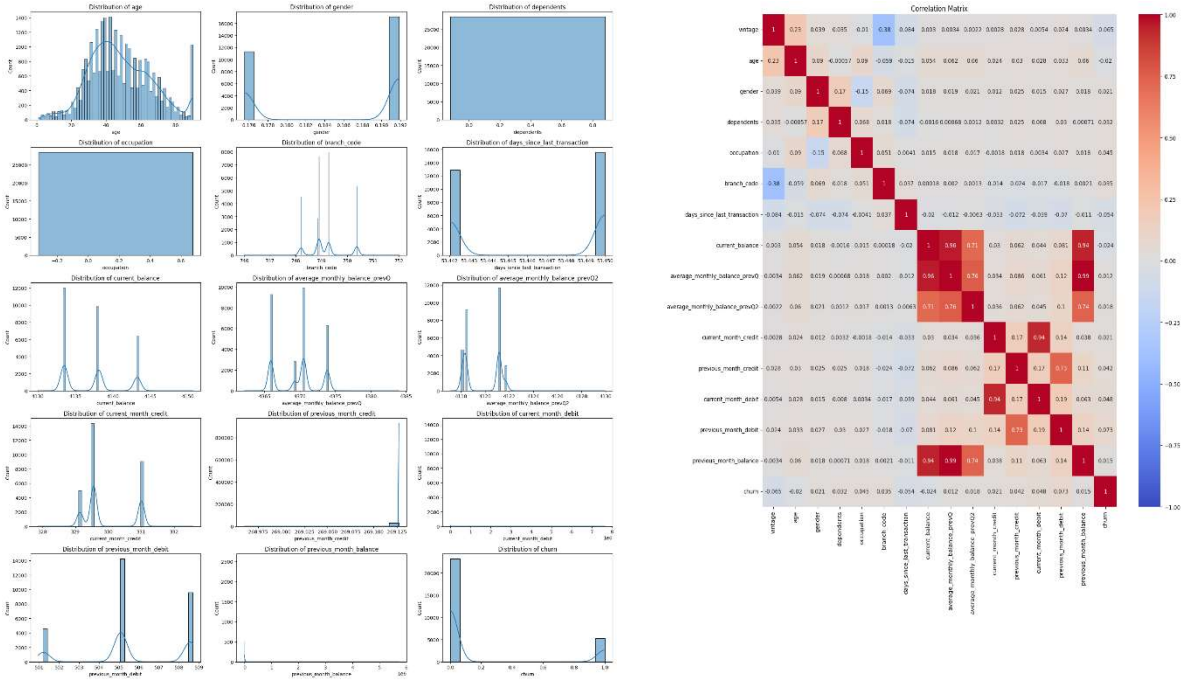




Dataset –2



Dataset-3



Data Preprocessing

Data preprocessing is a crucial step in preparing datasets for model training. This phase involves handling missing values, encoding categorical variables, feature selection, and data cleaning.

Handling Missing Values: Despite the absence of missing values in our datasets, we established protocols for handling potential missing data in other scenarios. For numerical attributes, in which 0 data values won't be possible is handled using mean imputation was applied, which preserves the central tendency of data, suitable for features with a normal

distribution. For categorical attributes, mode imputation was utilized, ensuring the preservation of the most frequent category.

Encoding

Categorical Transformation to Numerical: To convert categorical variables into numerical values, two encoding techniques were employed: Target Encoding and Label Encoding.

Target Encoding method transforms categorical variables into numeric values based on the target variable (whether a customer churned or not),considering the statistical significance of the establishing a direct correlation with the target variable and enhancing model performance.

Label Encoding is a straightforward simple algorithm converts non-numerical categorical data into numeric form where the order of categories is irrelevant, ensuring simple numerical representation.

Feature Selection

To optimize model performance, reduce training time, and mitigate overfitting, a correlation matrix was utilized. This enabled the identification and elimination of less relevant features, or noises ensuring only the most influential features were used for model training.

Data Cleaning

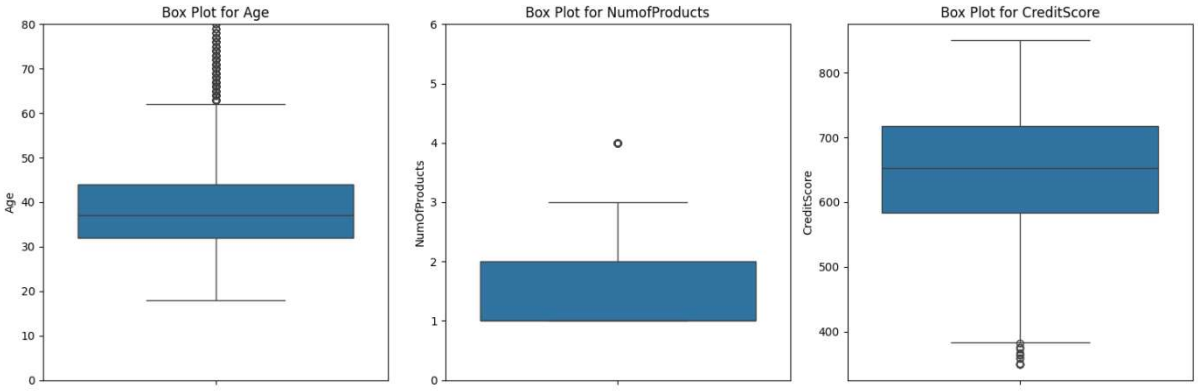
Data cleaning was conducted to enhance the quality and reliability of the datasets. Robust techniques were employed to ensure data integrity prior to model training.

Outlier Detection and Treatment

Outliers were identified using the Interquartile Range (IQR) method, with box plots providing visual insights into feature distributions. Outlier treatment techniques like mean imputation .were then applied to normalize data distributions and improve the accuracy of model predictions. We also tried median imputation ,but didn't proceed further since our dataset is imbalanced.

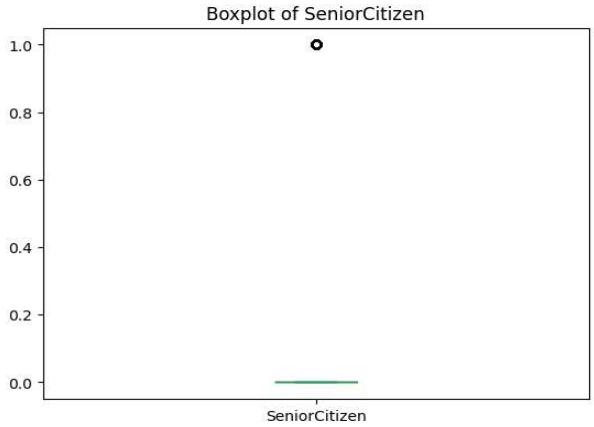
Outlier Visualisation by Boxplot for features

Dataset-1



Outliers for Age ,NumofProducts and CreditScore is shown here. By considering the range we have only replaced the outliers of CreditScore using mean imputation.

Dataset-2



There were outilers shown for SeniorCitizen since it was a minority feature. Since it is an important feature for churn prediction it was not handled.

Resampling

Resampling techniques were employed to address class imbalance, a common challenge that can lead to biased models and reduced predictive accuracy.

In Trial and Error Approach Various resampling techniques were tested, including SMOTE (Synthetic Minority Over-sampling Technique), ADASYN (Adaptive Synthetic Sampling), and manual resampling (over sampling ,under sampling). we didn't use over sampling and under sampling because they are copying the majority or minority classes or features resulting in noise.

SMOTE (Synthetic Minority Over-sampling Technique) technique was selected due to its effectiveness in improving accuracy by generating synthetic samples for the minority class. SMOTE helps in balancing the dataset, thereby enhancing model robustness and performance metrics such as accuracy, precision, recall, and F1-score.

Reasons for Choosing SMOTE: Beyond its accuracy improvements, SMOTE also aids in preventing overfitting by creating synthetic samples that expand the dataset's representation of the minority class. This approach ensures that the model generalizes well to unseen data, contributing to its reliability in real-world applications.

Results

The models were evaluated on the basis of the results for accuracy, precision, recall, F1 score, and the ROC curve. These metrics presented a clear view of how each of the models has performed in predicting customer churn. The accuracy metric measured the total correctness of the model; hence, proportion of true results—both true positives and true negatives—among the total number of cases tested. Precision expressed the model's ability to correctly identify positive instances, taking into account the accuracy of th

e positive predictions. Recall measured how well the model identified the relevant instances, making emphasis on how accurate the model is for the actual positives. The F1 score, as a harmonic mean of precision and recall, balanced these metrics to produce one measure of performance that gives greater views of the effectiveness of a model where there is an uneven class distribution. It was seen that the ROC curve showed how the true positive rate balances with the false positive rate under various threshold settings. The quantification of this for any given model was captured by the Area Under the Curve, which described the separability of the classes. Among all the trained models, especially XGBoost and Random Forest, show a very strong capability in predicting customer churn.

DATASET – 1: Bank Customer

Logistic Regression model achieved an accuracy of 80.45%, with an F1 score of 0.87 for class 0 and 0.59 for class 1, and an AUC of 0.862.

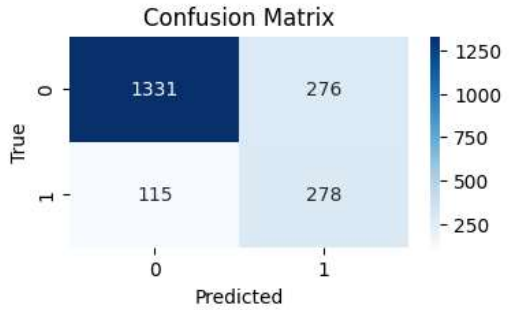
The XGBoost model performed better with an accuracy of 89.3%, an F1 score of 0.93 for class 0 and 0.72 for class 1, and an AUC of 0.922.

Random Forest also demonstrated high performance with an accuracy of 88.1%, an F1 score of 0.93 for class 0 and 0.70 for class 1, and an AUC of 0.913.

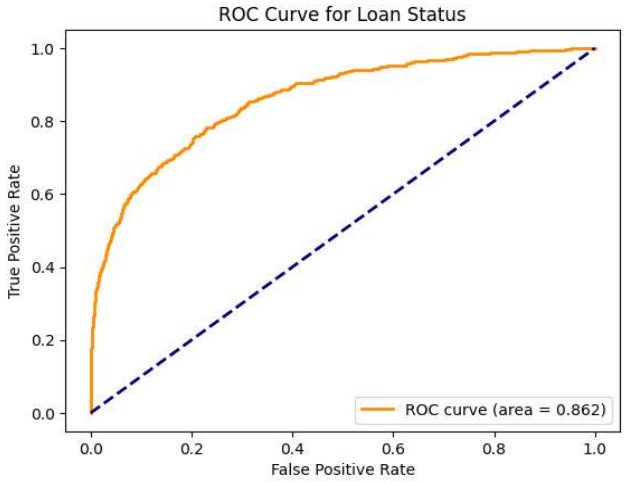
Lastly, the SVM model showed an accuracy of 87%, an F1 score of 0.92 for class 0 and 0.68 for class 1, and an AUC of 0.92.

Logistic regression

```
Accuracy score: 0.8045
Confusion Matrix:
[[1331  276]
 [ 115  278]]
```

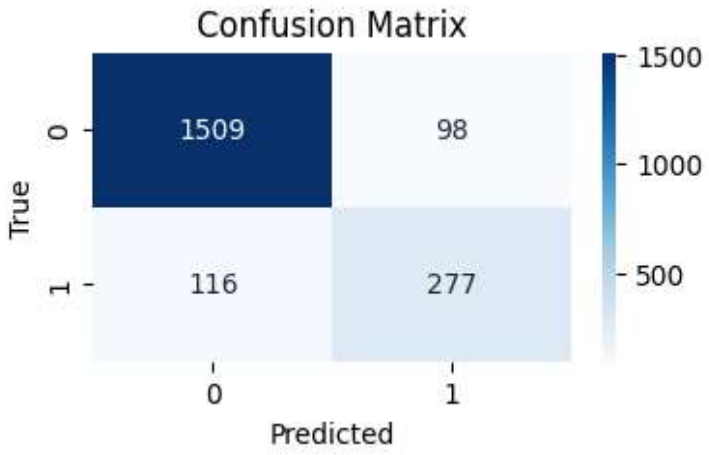



```
Metrics:
Precision: 0.5018050541516246
Recall: 0.7073791348600509
F1-score: 0.5871172122492081
```

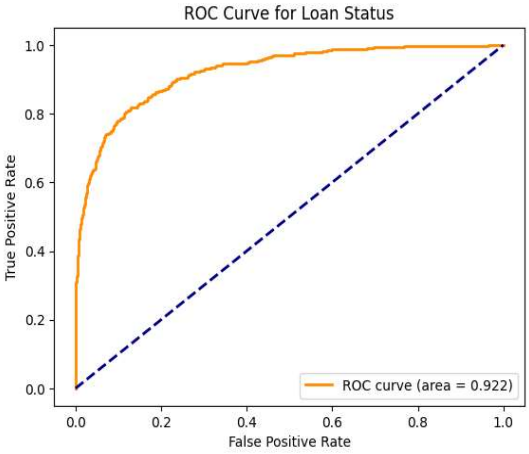


XGBoost

```
Accuracy: 0.893
ROC AUC: 0.9223198126517098
```

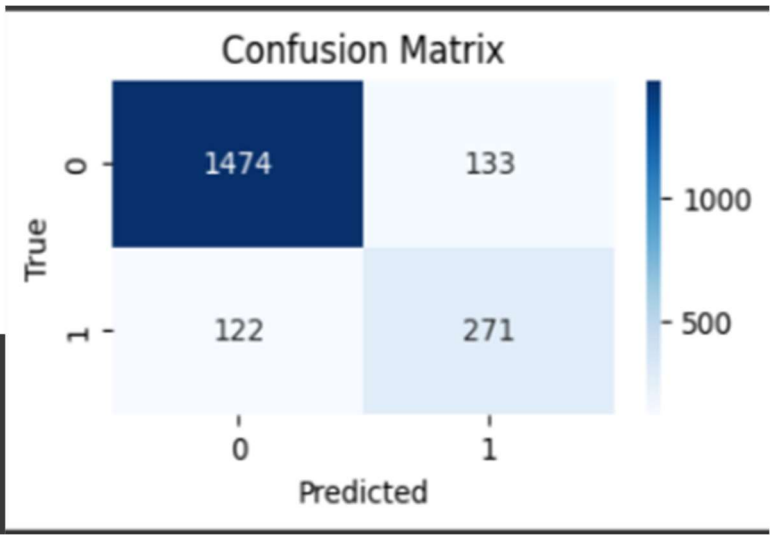


```
Metrics:
Precision: 0.7386666666666667
Recall: 0.7048346055979644
F1-score: 0.7213541666666666
```

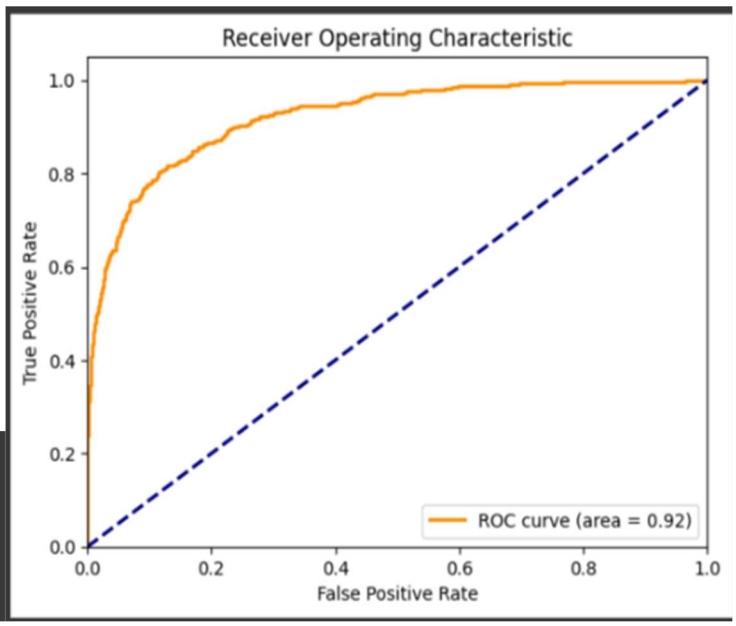


Random forest

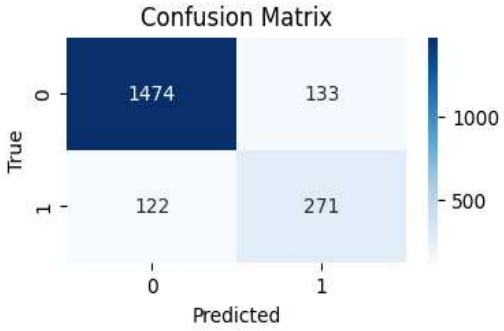
```
Accuracy: 0.881
Confusion Matrix:
 [[1482  125]
 [ 113  280]]
```



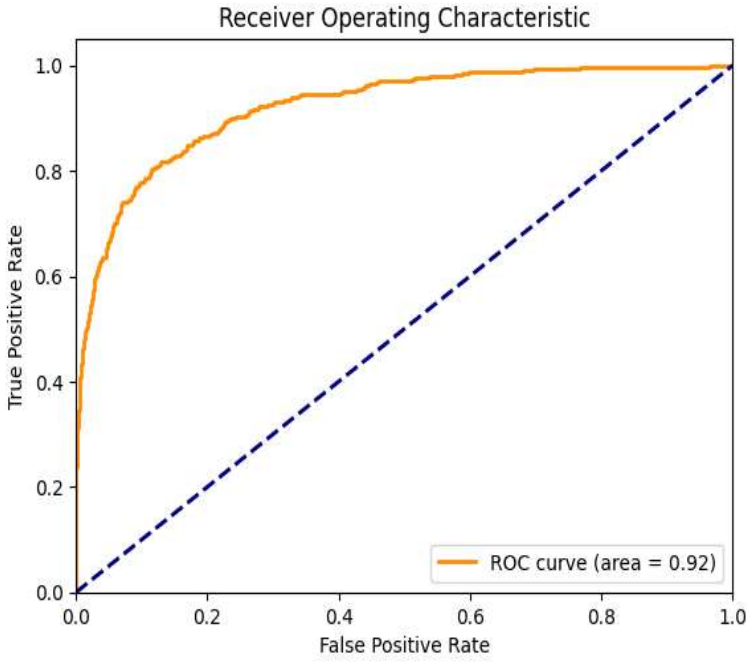

```
Metrics:
Precision: 0.6707920792079208
Recall: 0.6895674300254453
F1-score: 0.6800501882057717
```



SVM




```
Accuracy: 0.8725
```




```
Metrics:
Precision: 0.6707920792079208
Recall: 0.6895674300254453
F1-score: 0.6800501882057717
```

DATASET-2: Telco Customer Churn Prediction

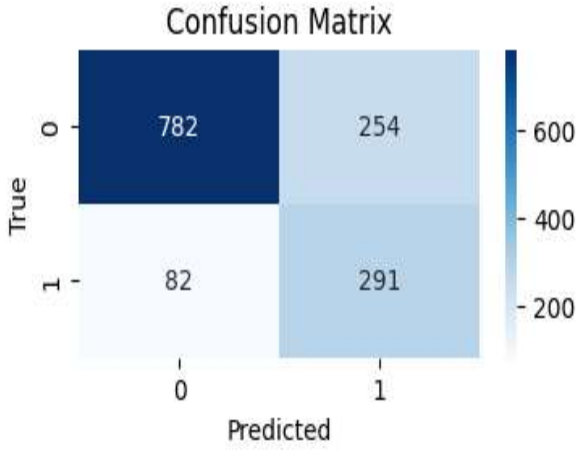
Logistic Regression reached an accuracy of 76%, with an F1 score of 0.82 for class 0 and 0.63 for class 1, and an AUC of 0.82.

XGBoost achieved a slightly higher accuracy of 77.92%, an F1 score of 0.85 for class 0 and 0.60 for class 1, and an AUC of 0.84.

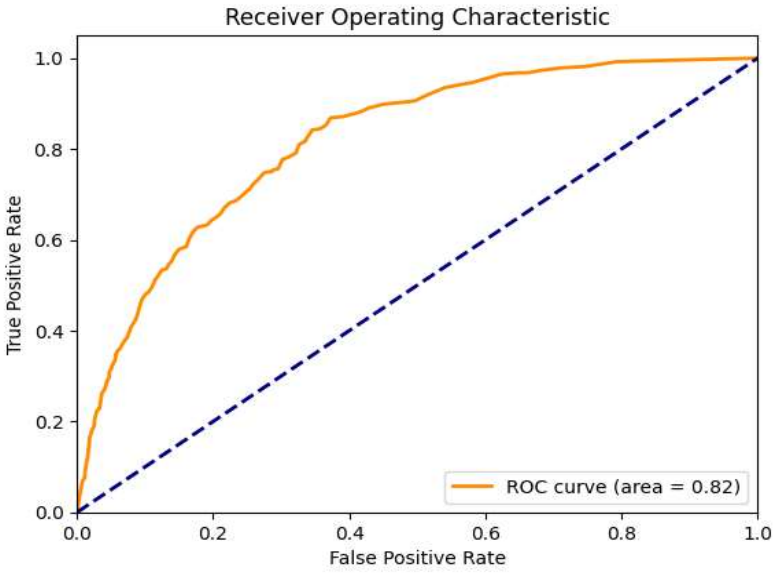
The Random Forest model had an accuracy of 77.85%, an F1 score of 0.85 for class 0 and 0.58 for class 1, and an AUC of 0.81.

The SVM model resulted in an accuracy of 76.57%, with an F1 score of 0.83 for class 0 and 0.60 for class 1, and an AUC of 0.83.

Logistic regression

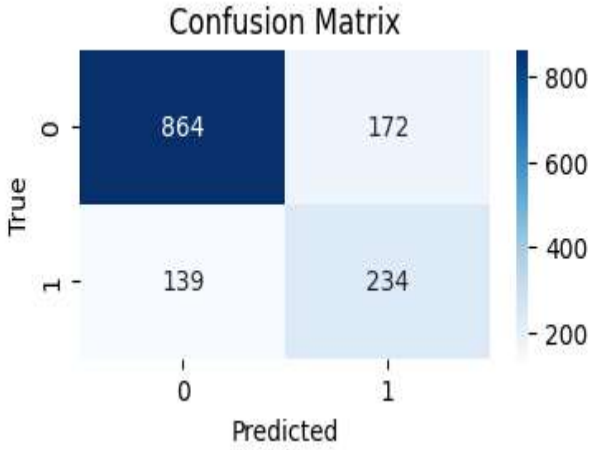



```
Accuracy: 0.7615330021291696
```

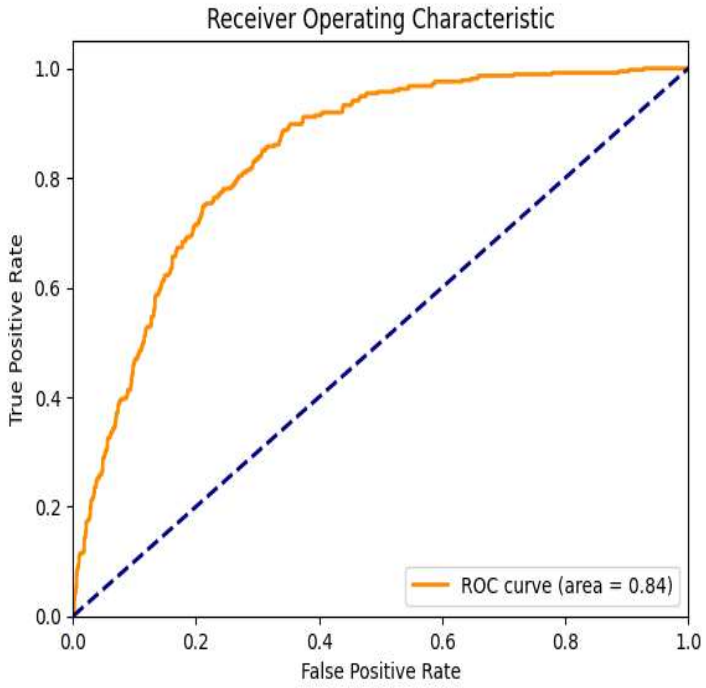



```
Metrics:
Precision: 0.5339449541284403
Recall: 0.7801608579088471
F1-score: 0.6339869281045751
```

XGBoost



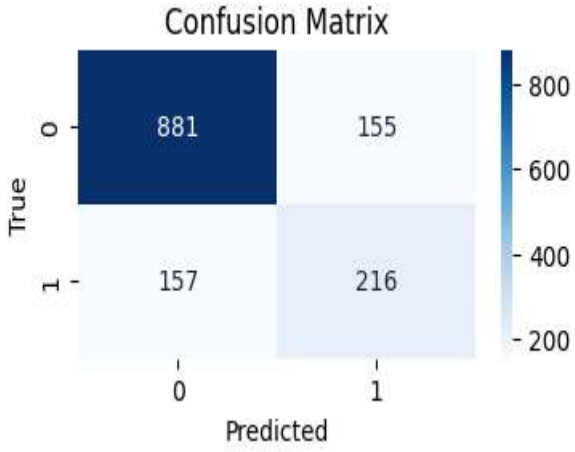

```
Accuracy: 0.7792760823278921
```



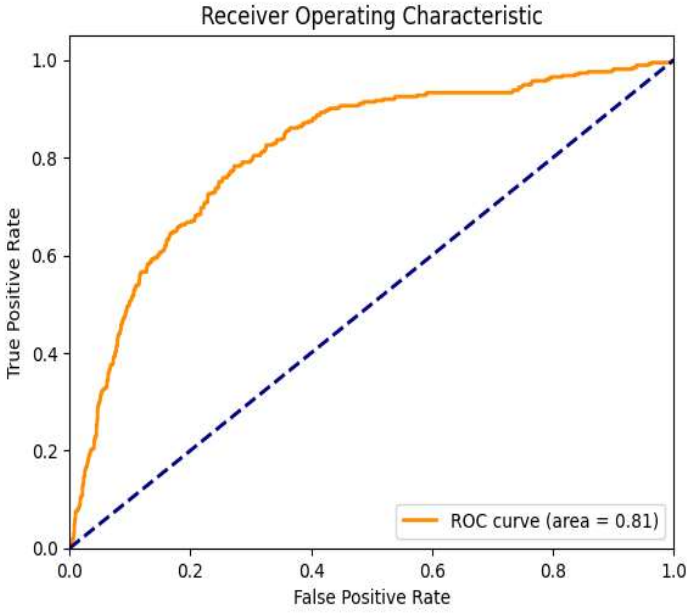

```
Metrics:
Precision: 0.5763546798029556
Recall: 0.6273458445040214
F1-score: 0.6007702182284981
```

Random forest

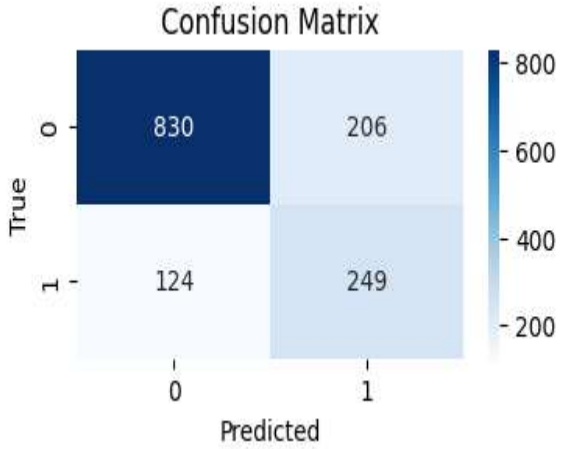
```
Accuracy: 0.7785663591199432
Confusion Matrix:
 [[881 155]
 [157 216]]
```



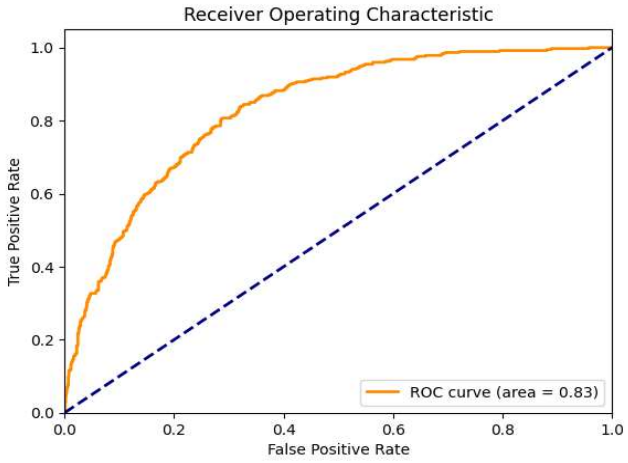

```
Metrics:
Precision: 0.5822102425876011
Recall: 0.579088471849866
F1-score: 0.5806451612903226
```



SVM




```
Accuracy: 0.765791341376863
```




```
Metrics:
Precision: 0.5472527472527473
Recall: 0.6675603217158177
F1-score: 0.601449275362319
```

DATASET-3: Financial Service Industry Customer Churn Prediction

Logistic Regression obtained an accuracy of 82.75%, with an F1 score of 0.90 for class 0 and 0.14 for class 1, and an AUC of 0.766.

XGBoost again showed superior performance with an accuracy of 85.76%, an F1 score of 0.92 for class 0 and 0.53 for class 1, and an AUC of 0.811.

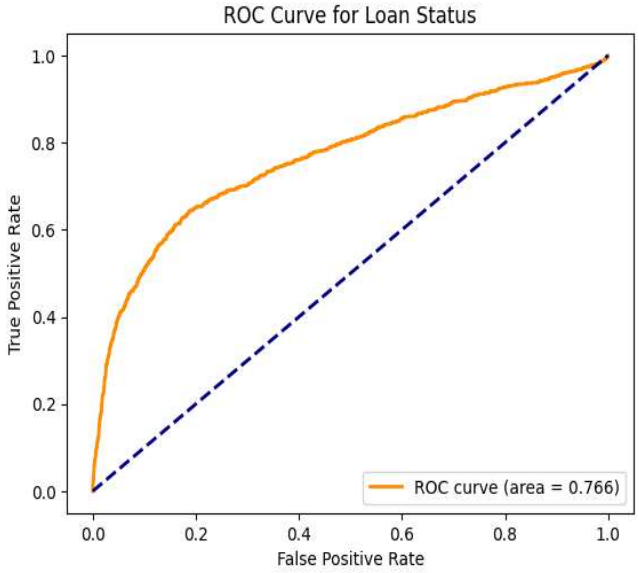
Random Forest achieved the highest accuracy of 86.52%, with an F1 score of 0.92 for class 0 and 0.53 for class 1, and an AUC of 0.836.

Finally, the SVM model had an accuracy of 81.73%, an F1 score of 0.90 for class 0 and 0.07 for class 1, and an AUC of 0.69.

Logistic regression

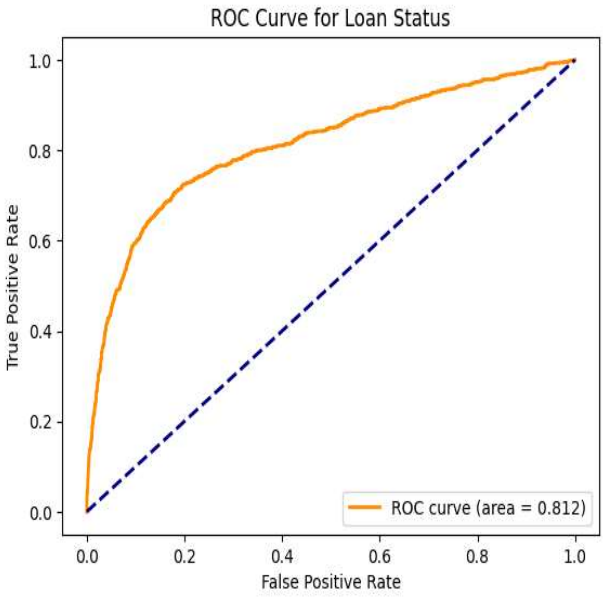
```
Accuracy score: 0.8275497621983442
Confusion Matrix:
[[4615   24]
 [ 955   83]]
```

	precision	recall	f1-score
0	0.83	0.99	0.90
1	0.78	0.08	0.14



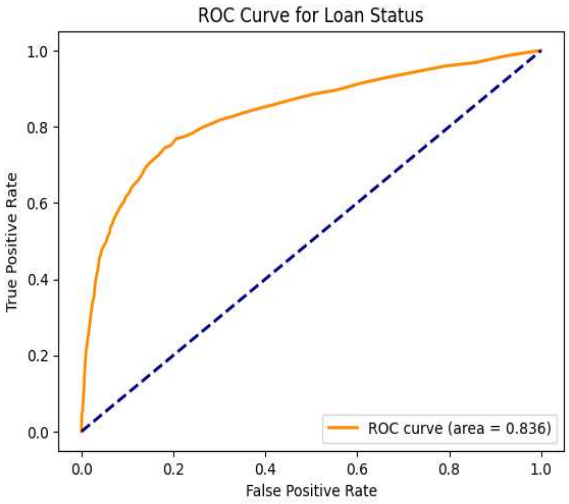
XGBoost

```
Accuracy: 0.8576713052668663
ROC AUC: 0.8118033377899779
              precision    recall  f1-score

           0       0.88      0.95      0.92
           1       0.67      0.45      0.53
```



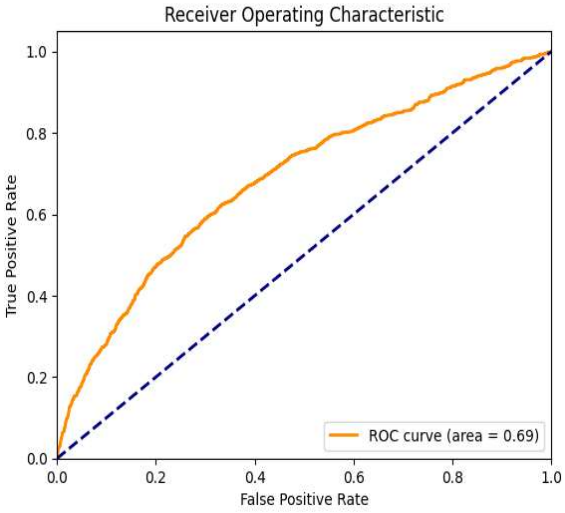
Random forest

```
Accuracy: 0.8652457283776642
Confusion Matrix:
 [[4487  152]
 [ 613  425]]
Classification Report:
              precision    recall  f1-score

           0       0.88      0.97      0.92
           1       0.74      0.41      0.53
```



SVM

```
Accuracy: 0.8173330984675005

Classification Report:
              precision    recall  f1-score

           0       0.82      0.99      0.90
           1       0.51      0.04      0.07
```



Conclusion

From our observation, in all the datasets, XGBoost and Random Forest generally performed consistently better than Logistic Regression and SVM, since they had higher accuracy, better F1 scores, and higher AUC values. This simply shows that ensemble methods are much better at handling many diverse datasets and hence yield reliable predictions. The result therefore dictates that in case future projects involve such datasets, XGBoost and Random Forest be applied as primary models since they perform better.

Among the various reasons as to why XGBoost and Random Forest performed best include:

Handling Nonlinearity: It is evident that both algorithms can capture nonlinearities in the data, unlike traditional methods such as Logistic Regression, that might have failed in doing so.

Robustness to overfitting: Algorithms such as Random Forest, which is a part of bagging, and XGBoost, which is regularization and boosting combined, are pretty robust against overfitting. The former combine many weak learners into a stronger model, thus providing better generalization on unseen data.

Handling imbalanced data: Since class imbalance is handled in-built inside both of them, class weighting is possible in the case of XGBoost, and class balancing is possible in Random Forest. This makes them very powerful because they perform better on data with underrepresented classes.

Most especially for XGBoost, this is known for efficiency and scalability because of various optimizations, including parallel processing. Therefore, it will do well with big datasets and complex computations to ensure faster training and prediction times.

Versatility: Both of them are versatile models that can handle any kind of data, numerical and categorical features. They are less sensitive to the need for extensive data preprocessing and can handle missing values and outliers quite efficiently.