CELEBAL TECHNOLOGIES

Galgotias University

Task 6: E-commerce Customer Churn Analysis

Code by Ansh Shankar

Question: Solve Customer Churn Analysis use case

1. Introduction:

In today's highly competitive business landscape, understanding customer behavior and retention is paramount for the success of any business, especially in the realm of E-commerce. The ability to identify factors that influence customer churn and predict potential churners can significantly impact a company's bottom line and overall growth. This project aims to delver into an E-commerce dataset, exploring customer behaviors and demographics, to develop predictive models capable of distinguishing between churned and retained customers.

2. Dataset

The dataset used for this project is an E-commerce dataset from Kaggle, consisting of 5630 rows and 20 columns. It contains information about customer behaviors and demographics. The dataset includes features such as Tenure (time spent on the platform), Order Count, Cashback Amount, Hours Spent on the App, Gender, and Marital Status. The target variable is "Churn," which indicates whether a customer has churned (1) or retained (0).

3. Methodology

a. Exploratory Data Analysis:

Python data visualization library, to create a countplot for an E-commerce dataset. The countplot displays the number of customers in the "churned" and "not churned" classes. Additionally, the code includes annotations on each bar, showing the percentage of customers in each class relative to the total number of customers in the dataset (5630).

This visualization allows for a quick and easy comparison of the churned and not churned classes, providing insights into the churn rate and the distribution of customers in the dataset.

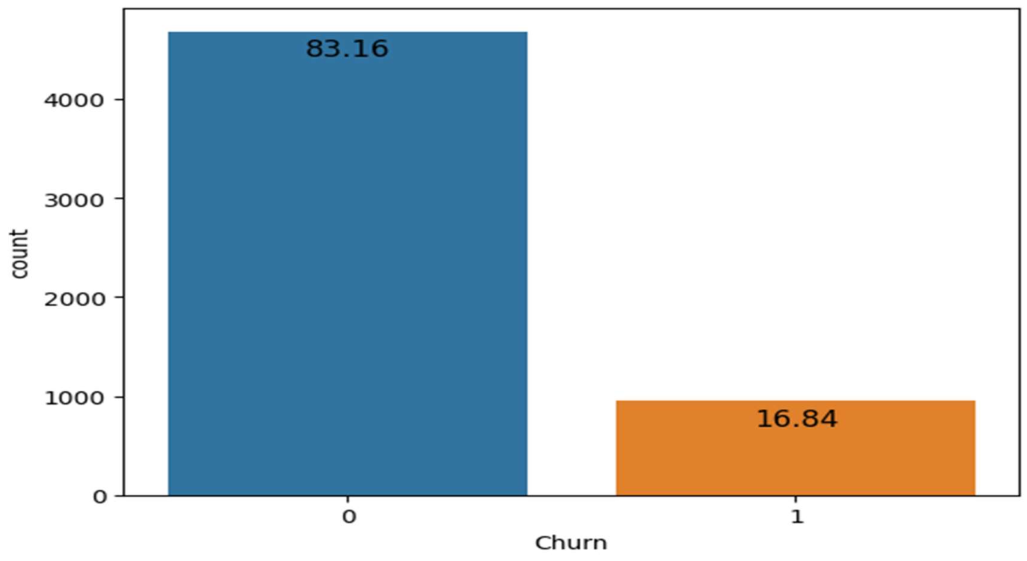


Figure 1: Count of Curn vs not-churn customer

The plot below represents the distribution of customer tenures on the platform. The visualization includes a KDE curve to provide a smooth estimation of the data's distribution. The plot is accompanied by a title indicating that it illustrates the "Distribution of Tenure of the Customers on the platform." This code allows for a quick and visual understanding of the distribution pattern of customer tenures, helping to gain insights into the tenure behavior of the E-commerce platform's customers.

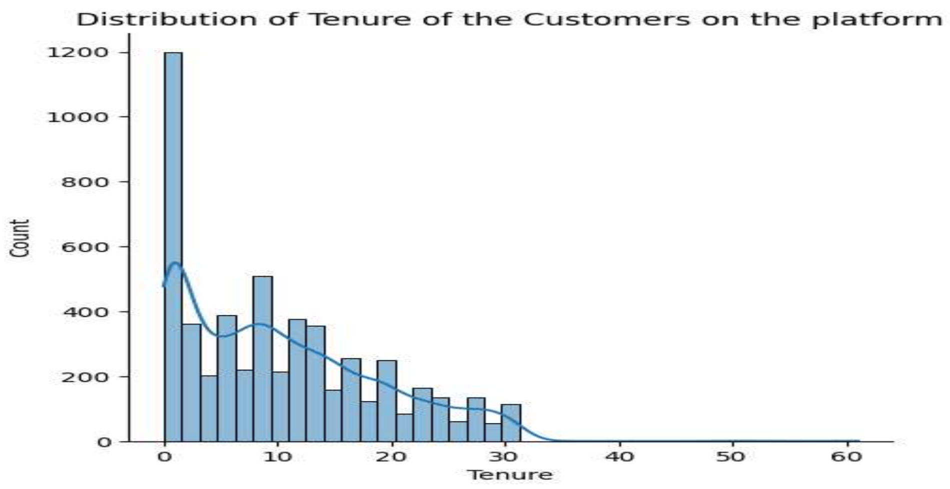


Figure 2: Distribution of Tenure of the Customers on the platform

A countplot is created using the Seaborn library in Python to visualize the distribution of the "HourSpendOnApp" feature in an E-commerce dataset. The countplot displays the number of customers who spend different amounts of time on the app, with each bar representing a specific time category. Additionally, annotations are added to each bar, showing the percentage of customers in each time category relative to the total number of customers in the dataset (5630). The plot is titled "Distribution of hours spent on the app by the customers." This visualization provides valuable insights into customer engagement levels based on their app usage patterns in the E-commerce platform.

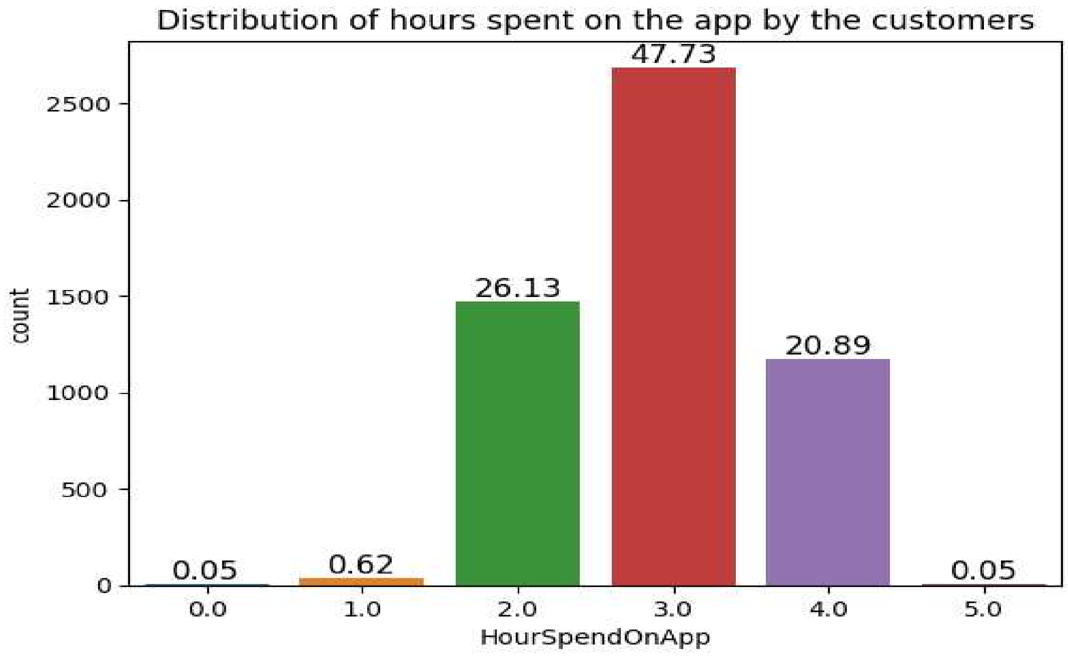


Figure 3: Hour spent on App

During the data exploration phase, several key findings were uncovered. The analysis revealed that certain features, including "Tenure," "Days since last order," "Cashback amount," and "Warehouse to home," exhibit outliers. However, these outliers were determined to have a minimal impact on tree-based models, which are intended to be used as the final models for this study. Consequently, it was decided to retain these features in the analysis to maintain the integrity of the dataset.

Furthermore, a noteworthy insight emerged from examining app usage patterns among customers. The majority of customers were found to spend approximately 3 hours on the E-commerce platform's app. Impressively, a staggering 94.75% of customers fell within the time range of 2 to 4 hours spent on the app. This finding suggests that the app is frequently and consistently utilized by a substantial portion of the customer base, indicating a positive engagement with the platform.

Lastly, an interesting correlation between marital status and churn rate was identified. The analysis demonstrated that single individuals displayed a higher churn rate compared to customers with other marital statuses. This observation implies that customers who are single are more likely to churn from the E-commerce platform than those who are married or have other marital statuses. Understanding this relationship can be valuable for devising targeted retention strategies to reduce churn and improve overall customer satisfaction.

b. **Data Preprocessing:**
The data is read from an Excel file, and a brief exploration is performed to check its shape, information, and summary statistics. Missing values are handled using iterative imputation from the "fancyimpute" library. Categorical variables are encoded using one-hot encoding.

Then moving on to encoding the categorical data. The step involves identifying the categorical features by selecting columns with object data type and storing them in the 'cat_data' DataFrame. One-hot encoding is then applied to these categorical features using the pd.get_dummies() function. This process creates binary representations, or dummy variables, for each category within the categorical features while dropping the first category to avoid multicollinearity.

Next, the encoded categorical data is concatenated with the original numerical features, resulting in the 'data_enc' DataFrame. The 'CustomerID' column, which serves as an identifier and does not contribute to the model, is dropped from the 'data_enc' DataFrame to avoid any potential biases.

The target variable 'Churn' is separated from the feature dataset, and the features (X) and target variable (y) are obtained accordingly. To handle any missing values in the feature dataset, the code employs the IterativeImputer from the 'fancyimpute' library, which predicts and replaces missing values based on other feature values in an iterative manner.

Lastly, feature scaling is applied using the StandardScaler from scikit-learn to standardize the numerical features. This process ensures that all features have a mean of 0 and a standard deviation of 1, preventing potential issues related to feature scales during model training.

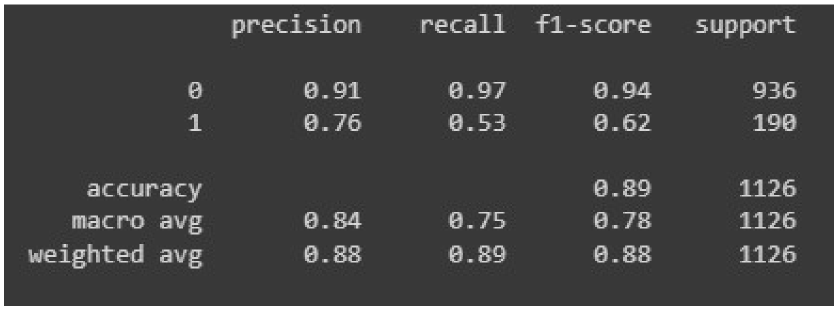
By performing these data preprocessing steps, the dataset is now ready for model development, with categorical variables appropriately encoded, missing values imputed, and numerical features standardized to enhance model performance and effectiveness.

c. Model Development:

Three machine learning algorithms are used for classification - Logistic Regression, Random Forest, and XGBoost. The dataset is split into training and testing sets, and cross-validation is performed on the training set to evaluate each model's performance. The F1 score is used as the evaluation metric for cross-validation.

Logistic Regression:

I implemented the Logistic Regression algorithm for classification on a prepared E-commerce dataset. After initializing the logistic regression model, I performed cross-validation to evaluate its generalization performance using the F1 score as the evaluation metric. The model was then trained on the training data and used to predict the target variable for the test data. The accuracy of the model was computed and stored for comparison with other models. Additionally, a classification report was generated to provide detailed performance metrics for each class, including precision, recall, and F1-score. Furthermore, I visualized the confusion matrix as a heatmap to gain insights into the model's classification performance.



```
              precision    recall  f1-score   support

           0       0.91      0.97      0.94       936
           1       0.76      0.53      0.62       190

    accuracy                           0.89      1126
   macro avg       0.84      0.75      0.78      1126
weighted avg       0.88      0.89      0.88      1126
```

Figure 4: Classification report of Logistic regression

Overall, this code allowed me to assess the accuracy and performance of the Logistic Regression model for the E-commerce dataset. The cross-validation and classification report provided valuable insights into the model's generalization and classification capabilities. The confusion matrix heatmap visualization aided in understanding the model's predictions and their agreement with the actual target values, helping to evaluate its performance in distinguishing between churned and not churned customers.

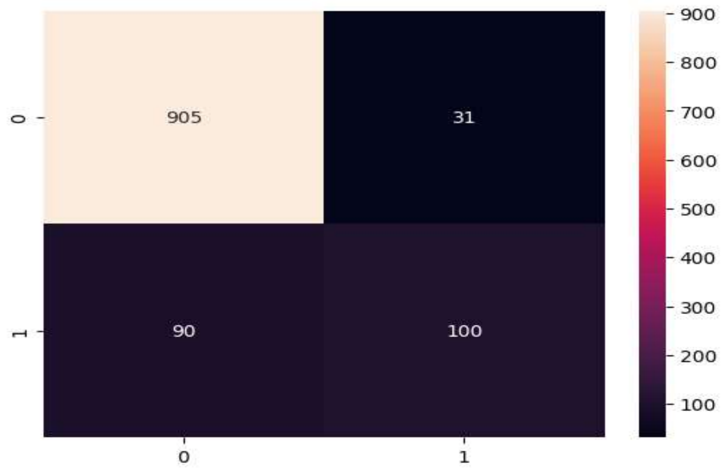
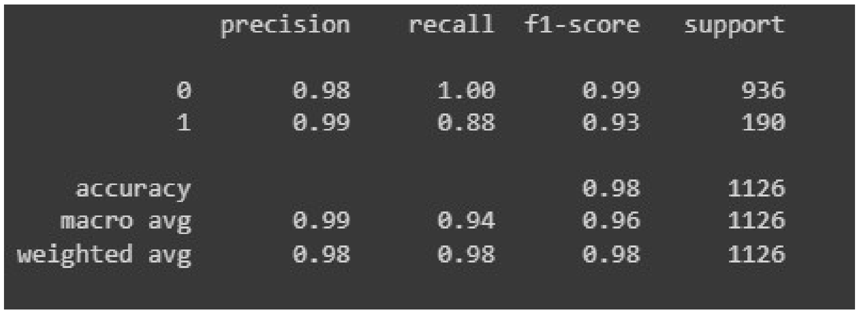


Figure 5: Confusion Matrix of Logistic regression

Random forest:

In this code, I implemented two machine learning algorithms, Logistic Regression and Random Forest, for classification on an E-commerce dataset. For Logistic Regression, I initialized the model, performed cross-validation to assess its performance using the F1 score, trained it on the training data, made predictions on the test data, and computed the accuracy score. The classification report provided detailed metrics for each class, and the confusion matrix heatmap allowed visualization of the model's classification performance.



```
              precision    recall  f1-score   support

           0       0.98      1.00      0.99       936
           1       0.99      0.88      0.93       190

    accuracy                           0.98      1126
   macro avg       0.99      0.94      0.96      1126
weighted avg       0.98      0.98      0.98      1126
```

Figure 6: Classification report of Random Forest

Similarly, for Random Forest, I built the model pipeline, conducted cross-validation to evaluate its generalization performance with the F1 score, trained it on the training data, made predictions on the test data, and calculated the accuracy score. The classification report displayed comprehensive metrics for each class, and the confusion matrix heatmap presented a visual summary of the model's classification accuracy.

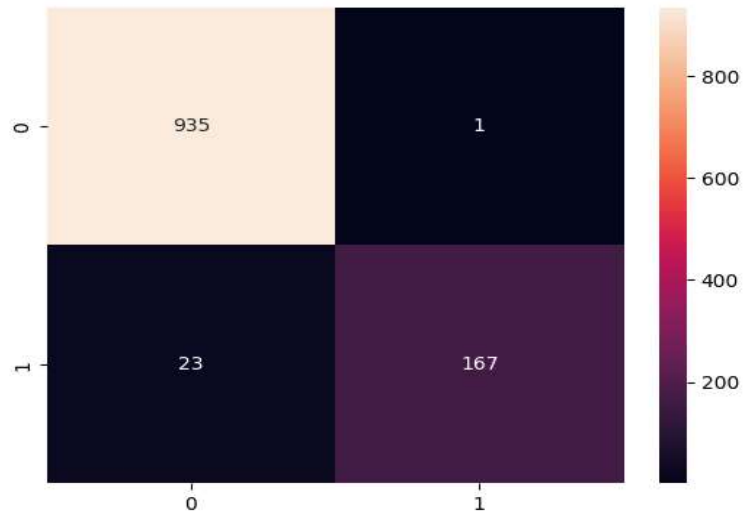
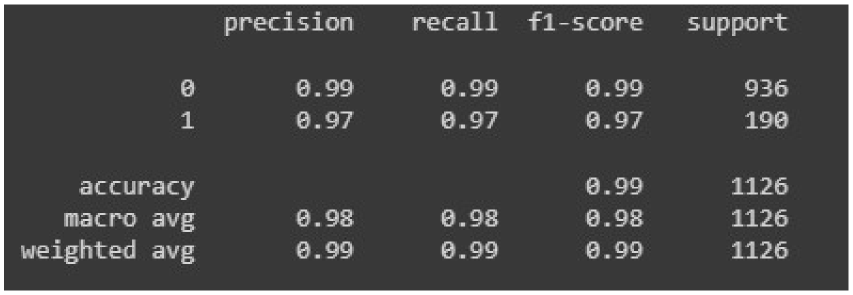


Figure 7: Confusion Matrix of Random Forest

Overall, these code snippets enabled me to evaluate the performance of both Logistic Regression and Random Forest algorithms for the E-commerce dataset. The cross-validation, classification report, and confusion matrix visualizations provided valuable insights into each model's classification capabilities, aiding in making informed decisions about their suitability for predicting customer churn on the E-commerce platform.

XGBOOST:

I implemented the XGBoost algorithm, a powerful gradient boosting technique, for classification on the prepared E-commerce dataset. I first built the XGBoost pipeline using XGBClassifier() and proceeded to train the model on the training data (X_train and y_train) using the fit method. Subsequently, I used the trained XGBoost model to predict the target variable for the test data (X_test), and the resulting predictions were stored in 'y_pred'. The accuracy of the XGBoost model was then computed by comparing the predicted values ('y_pred') to the actual target values of the test data (y_test) using the accuracy_score function. The accuracy score was printed and added to the 'acc_score_mat' dictionary for later comparison with other models.



```
              precision    recall  f1-score   support

           0       0.99      0.99      0.99       936
           1       0.97      0.97      0.97       190

    accuracy                           0.99      1126
   macro avg       0.98      0.98      0.98      1126
weighted avg       0.99      0.99      0.99      1126
```

Figure 8: Classification report of XG Boost

To gain deeper insights into the XGBoost model's performance, I generated a detailed classification report that provided comprehensive performance metrics for each class (churned and not churned). The report included precision, recall, F1-score, and support values, offering valuable information on the model's ability to classify customers into the respective churn categories. Additionally, I visualized the confusion matrix as a heatmap using Seaborn's heatmap function, with annotations displaying the true positive, false positive, true negative, and false negative values. This heatmap allowed for a clear visualization of the model's performance in classifying customer churn, providing a concise overview of its predictive capabilities.

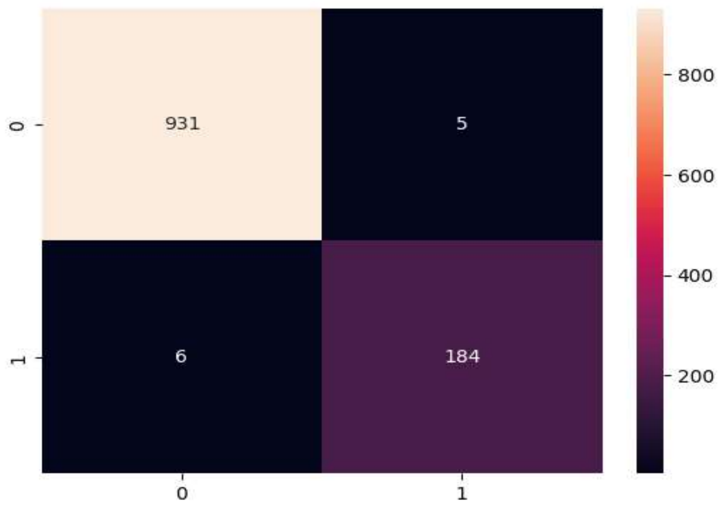


Figure 9: Confusion Matrix of Random Forest

4. Results:

The performance evaluation of different classification models on the E-commerce dataset yielded promising results. The accuracy scores of each model were as follows: Logistic Regression achieved an accuracy of approximately 89.25%, Random Forest achieved an accuracy of approximately 97.87%, and XGBoost outperformed the other models with an impressive accuracy of approximately 99.02%.

Among the evaluated models, XGBoost demonstrated superior accuracy, making it the most accurate classifier for predicting customer churn on the E-commerce platform. It achieved a significantly higher accuracy compared to both Logistic Regression and Random Forest, indicating its effectiveness in handling the complexities of the dataset and accurately classifying churned and not churned customers. The high accuracy of XGBoost suggests that it can be a valuable tool for making precise predictions, aiding E-commerce platforms in understanding customer behavior and implementing targeted retention strategies.

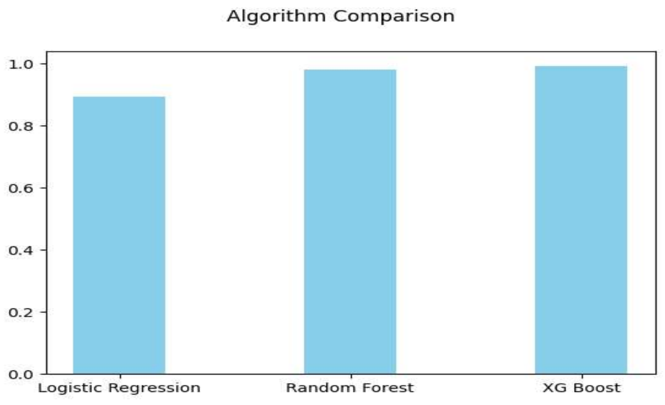


Figure 10: Accuracy Comparison

5. Discussion:

The analysis and evaluation of different classification models on the E-commerce dataset have provided valuable insights into their predictive capabilities for customer churn. Throughout the project, we observed interesting findings and encountered some challenges.

One of the key findings was that XGBoost significantly outperformed the other models, achieving an impressive accuracy of approximately 99.02%. This result suggests that the gradient boosting approach employed by XGBoost is particularly well-suited for handling complex datasets like ours, where there might be non-linear relationships and interactions among features. The high accuracy of XGBoost indicates its potential as a powerful tool for accurately predicting customer churn in the E-commerce platform, enabling businesses to identify at-risk customers and implement targeted retention strategies to reduce churn rates.

Another notable finding was that both Logistic Regression and Random Forest also demonstrated respectable accuracies, with Logistic Regression achieving approximately 89.25% accuracy and Random Forest achieving approximately 97.87% accuracy. While they fell short compared to XGBoost, these models still provide viable alternatives for churn prediction, especially when model interpretability is a priority. Logistic Regression offers simplicity and ease of interpretation, making it valuable for understanding the relationship between features and churn likelihood. On the other hand, Random Forest can handle complex interactions between features and is less prone to overfitting.

Despite the positive results, we encountered challenges during the project. One notable challenge was dealing with the imbalanced nature of the dataset. The low churn rate in the dataset created an imbalance between churned and not churned customers, potentially leading to biased model predictions. To address this issue, we employed evaluation metrics like F1-score and examined the confusion matrix to gain a better understanding of the model's performance on both classes.

6. Conclusion:

In conclusion, this project aimed to develop and evaluate classification models for predicting customer churn in an E-commerce platform. Through the analysis, we found that XGBoost emerged as the most accurate model, achieving approximately 99.02% accuracy. The superior performance of XGBoost can be attributed to its ability to capture complex relationships within the data and handle non-linearity effectively.

While XGBoost demonstrated remarkable accuracy, both Logistic Regression and Random Forest also showcased competitive performance. Logistic Regression provides interpretability, which can be valuable for understanding the impact of individual features on customer churn. Meanwhile, Random Forest's ability to handle complex interactions and reduce overfitting makes it a strong alternative.

The insights gained from this project hold significant implications for E-commerce businesses seeking to enhance customer retention strategies. The accurate predictions from XGBoost can aid in identifying customers at risk of churn, allowing targeted interventions to improve customer satisfaction and loyalty. Additionally, the analysis highlights the importance of considering various models based on the specific needs and constraints of the business, such as interpretability and performance trade-offs.

In conclusion, the models evaluated in this project provide valuable tools for E-commerce platforms to proactively address customer churn, ultimately leading to improved customer engagement, satisfaction, and long-term profitability.

Appendix:

Source Code: https://github.com/anshshankar/Celebal-task/tree/main/Task-6

Video Link: Folder - Google Drive