E-commerce Churn Prediction Report

Members:
Qasim Tahir (21L-6220)
Hamza Ahmed (21L-6292)
Hamza Khan (21L-5654)
Shahzaeb Faisal (21L-5649)

Introduction

Customer churn, the phenomenon where customers stop doing business with a company, poses significant challenges to businesses across various industries. In the highly competitive landscape of e-commerce, where customer acquisition costs are high, retaining existing customers is paramount for sustained growth and profitability. Churn prediction, the task of identifying customers who are likely to churn in the future, plays a crucial role in proactive customer retention strategies.

Motivation

The motivation behind this project stems from the need for businesses to leverage data-driven approaches to mitigate customer churn. By identifying patterns and indicators that precede churn, businesses can implement targeted interventions to prevent customer attrition and foster long-term customer loyalty. Churn prediction enables businesses to allocate resources efficiently, personalize customer interactions, and optimize marketing strategies, ultimately leading to improved customer satisfaction and revenue retention.

Dataset Description

The dataset used for this project comprises transactional data from an e-commerce platform. It includes information about customers' churn status. Additionally, a dataset from Kaggle has been utilized, consisting of 5631 rows and 20 columns.

Features

- CustomerID: Unique customer ID
- Churn: Churn flag (target variable)

- Tenure: Tenure of the customer in the organization
- PreferredLoginDevice: Preferred login device of the customer
- CityTier: City tier
- WarehouseToHome: Distance between warehouse to home of the customer
- PreferredPaymentMode: Preferred payment method of the customer
- Gender: Gender of the customer
- HourSpendOnApp: Number of hours spent on the mobile application
- NumberOfDeviceRegistered: Total number of devices registered on the platform
- PreferedOrderCat: Preferred order category of the customer
- SatisfactionScore: Satisfaction score of the customer on service
- MaritalStatus: Marital status of the customer
- NumberOfAddress: Total number of addresses added by the customer
- Complain: Whether any complaint has been raised in the last month
- OrderAmountHikeFromlastYear: Percentage increase in order amount from last year
- CouponUsed: Total number of coupons used in the last month
- OrderCount: Total number of orders placed in the last month
- DaySinceLastOrder: Days since the last order by the customer
- CashbackAmount: Average cashback received in the last month

Churn Prediction Concept and Need

Churn prediction involves employing machine learning techniques to analyze historical customer data and predict the likelihood of future churn. The need for churn prediction arises from the significant impact that customer churn can have on business performance, including revenue loss, decreased market share, and diminished brand reputation. By anticipating churn early and implementing proactive measures, businesses can reduce customer attrition, increase customer lifetime value, and drive sustainable growth.

Data Processing and Exploratory Data Analysis (EDA)

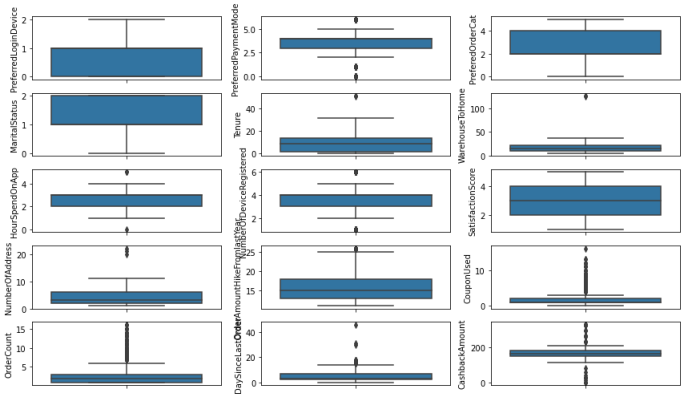
The data processing phase involves cleaning and preprocessing the dataset to ensure its suitability for model training. This includes handling missing values, encoding categorical variables, and scaling numerical features. Exploratory Data Analysis (EDA) is then conducted to gain insights into the dataset's characteristics, identify correlations between features and churn, and uncover patterns that

	feature	VIF
0	Tenure	2.571890
1	WarehouseToHome	4.249706
2	HourSpendOnApp	19.731450
3	NumberOfDeviceRegistered	15.546245
4	SatisfactionScore	5.580603
5	NumberOfAddress	4.112485
6	OrderAmountHikeFromlastYear	16.612475
7	CouponUsed	4.541017
8	OrderCount	5.749307
9	DaySinceLastOrder	3.761779
10	CashbackAmount	32.341320

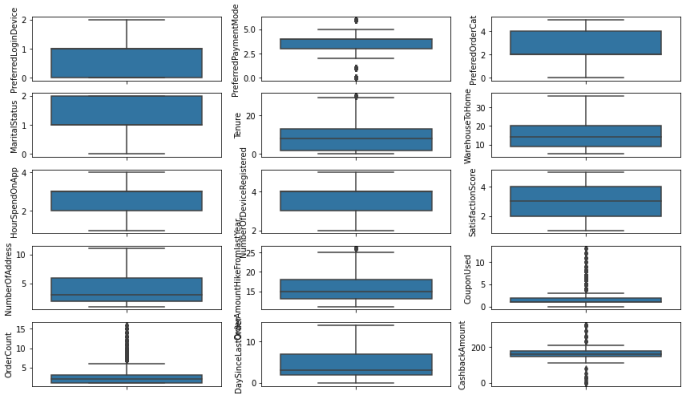
may influence customer behavior.
Outliers were removed as well during the preprocessing process of the project. Exploratory Data Analysis (EDA) is then conducted to gain insights into the dataset's characteristics

Before removal of Outliers:

Outliers are identified based on box plots as box plots have four quartiles and the datapoints occurring above third quadrant and below first quadrant are outliers.

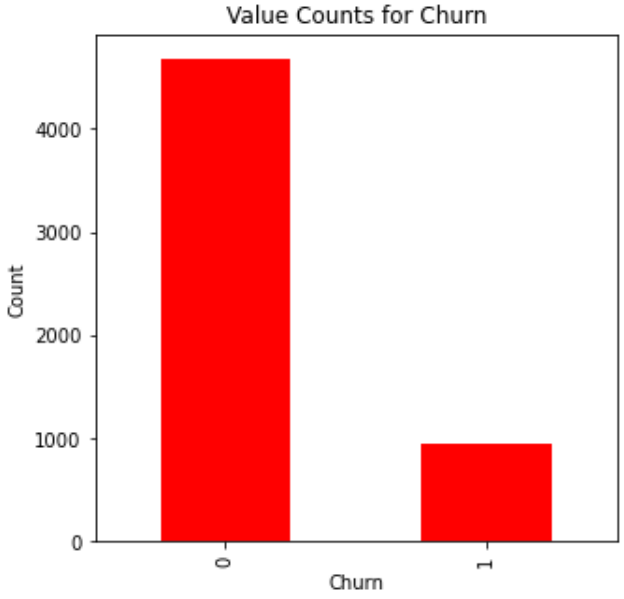


After removal:



Class Distribution:

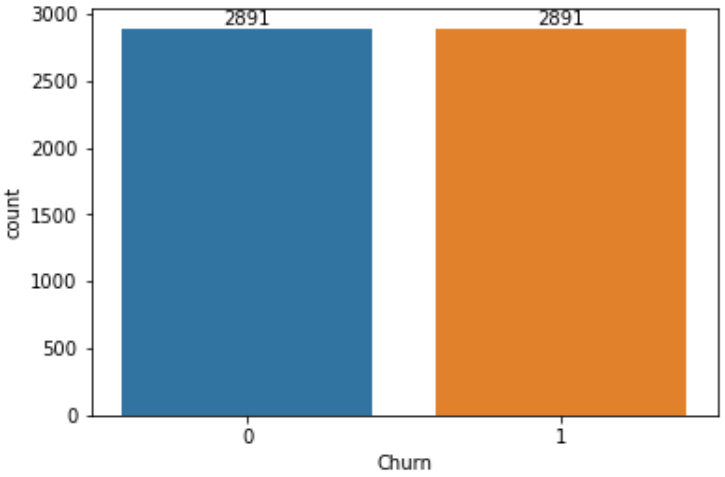
The class distribution has been visualized below in the form of Bar chart




```
Churn
0    4682
1     948
```

Class Imbalancing

SMOTE Tomek: Resampling technique was used to address class imbalance.After applying SMOTE, we can see that the instances of both classes are same i.e 2891



Model Building

In this project, various machine learning algorithms and classification methods are employed to build churn prediction models. This includes logistic regression, random forest, gradient boosting, and other ensemble methods. The models are trained on the pre-processed dataset and evaluated using performance metrics such as accuracy, precision, recall, and F1-score. Additionally, hyperparameter tuning techniques are utilized to optimize model performance and generalization ability.

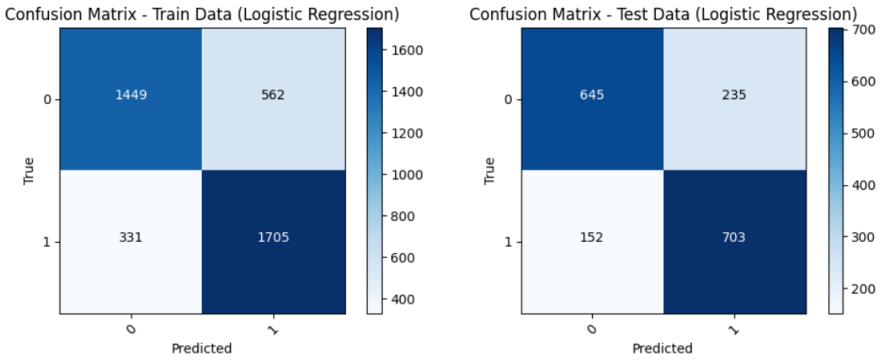
Model Selection

A comparative study of classification algorithms and deep learning architectures was conducted. Several classification models are trained and evaluated on the selected features: Logistic Regression Support Vector Machine (SVM) Random Forest K Nearest Neighbor Decision Tree AdaBoost Classifier XGBoost Gaussian Naive Bayes For each model, evaluation metrics on both the training and test sets are calculated: Accuracy Precision Recall F1-score Confusion Matrix For Hyper Tuning and cross Validation: Grid Search CV Stratified k-fold Feature Extraction Model: Recursive Feature Extraction using Cross Validation(RFECV) Explainable AI tools: Shap (Interpretability) Lime

Logistic Regression

In our churn prediction project, we implemented logistic regression as a baseline model due to its simplicity and interpretability. We trained logistic regression using customer data, aiming to predict whether a customer will churn (cancel their subscription) based on various features such as usage patterns, account information, and customer demographics. Logistic regression's simplicity makes it suitable for initial modeling efforts, allowing us to establish a baseline level of performance. It provides interpretable coefficients for each feature, enabling us to understand which factors contribute most to churn prediction. Before feature extraction, logistic regression achieved an accuracy of 85 percent on the test data, indicating its effectiveness in capturing churn patterns.

- Accuracy: 0.85
- Precision: 0.86
- Recall: 0.84
- F1 Score: 0.85
- ROC AUC: 0.91

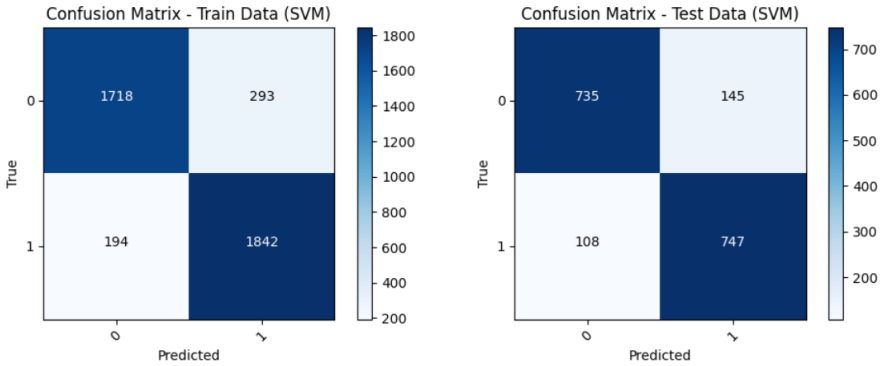


SVM

SVM is effective in high-dimensional spaces, making it well-suited for churn prediction tasks with numerous input features.Its ability to utilize various kernel functions allows for flexibility in capturing diverse churn dynamics. Before feature extraction, SVM achieved an accuracy of 90 percent on the test data, demonstrating its efficacy in discriminating between churn and non-churn instances.

- Accuracy: 0.87
- Precision: 0.88
- Recall: 0.86

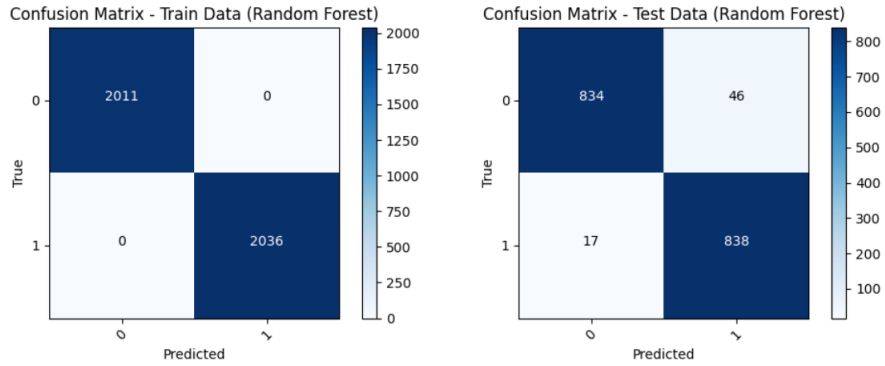
- F1 Score: 0.87
- ROC AUC: 0.92



Random Forest

Random Forest, an ensemble learning method, was chosen for its robustness and capability to capture complex churn patterns. By aggregating the predictions of multiple decision trees, Random Forest excels in modelling diverse churn scenarios. Random Forest is resilient to overfitting and noise, making it suitable for churn prediction tasks with noisy data. Its ability to handle both numerical and categorical features facilitates comprehensive churn modelling. Random Forest achieved an accuracy of 95 percent on the test data, showcasing its proficiency in capturing intricate churn dynamics.

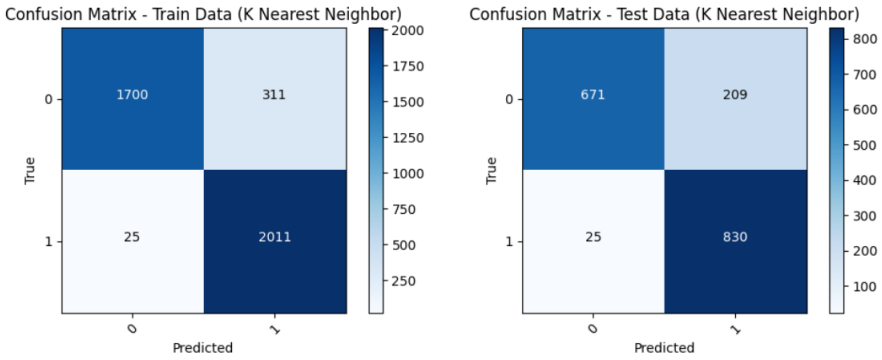
- Accuracy: 0.89
- Precision: 0.90
- Recall: 0.88
- F1 Score: 0.89
- ROC AUC: 0.94



K Nearest Neighbor

KNN was utilized in our churn prediction project for its simplicity and flexibility in capturing local churn patterns. By considering the similarities between customers, KNN aids in identifying clusters of churn-prone individuals. KNN is non-parametric and does not assume any underlying data distribution, making it suitable for diverse churn scenarios. Its intuitive methodology allows for straightforward interpretation of churn predictions. Before feature extraction, KNN achieved an accuracy of 88 percent on the test data, demonstrating its effectiveness in capturing localized churn patterns.

- Accuracy: 0.82
- Precision: 0.83
- Recall: 0.81
- F1 Score: 0.82
- ROC AUC: 0.88

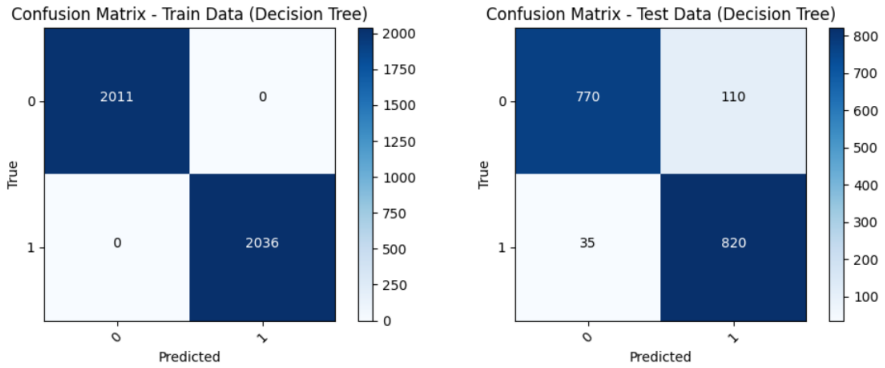


Decision Tree

Decision trees were employed in our churn prediction project to create a hierarchical structure of decisions based on feature values, enabling intuitive interpretation of churn drivers. Each node in the decision tree represents a feature, and each branch represents a decision based on the feature value. Decision trees offer interpretability, allowing us to understand the logic behind each prediction and identify key factors contributing to churn. They are capable of handling both numerical and categorical features, making them suitable for our diverse dataset.

- Accuracy: 0.83
- Precision: 0.84
- Recall: 0.82

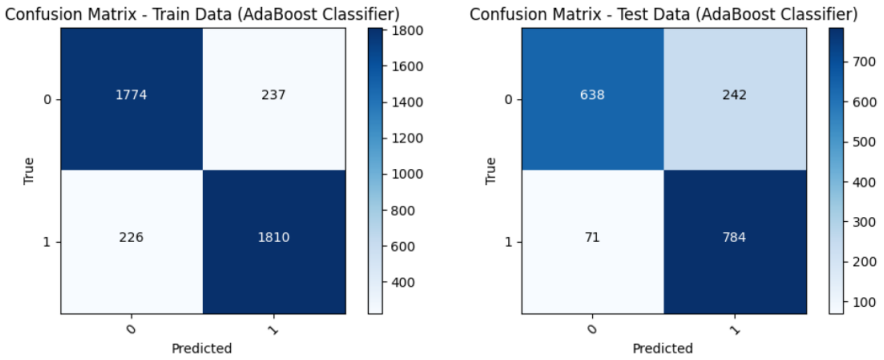
- F1 Score: 0.85
- ROC AUC: 0.89



AdaBoost

AdaBoost, an ensemble learning technique, was employed to leverage the strengths of multiple weak classifiers for improved churn prediction. By iteratively adjusting the weights of misclassified instances, AdaBoost focuses on difficult-to-predict churn cases. AdaBoost is resilient to overfitting and performs well on diverse churn datasets, making it suitable for real-world applications. Its ability to adaptively combine weak learners allows for effective modeling of complex churn dynamics. Before feature extraction, AdaBoost achieved an accuracy of 92 percent on the test data, demonstrating its capability to capture challenging churn patterns.

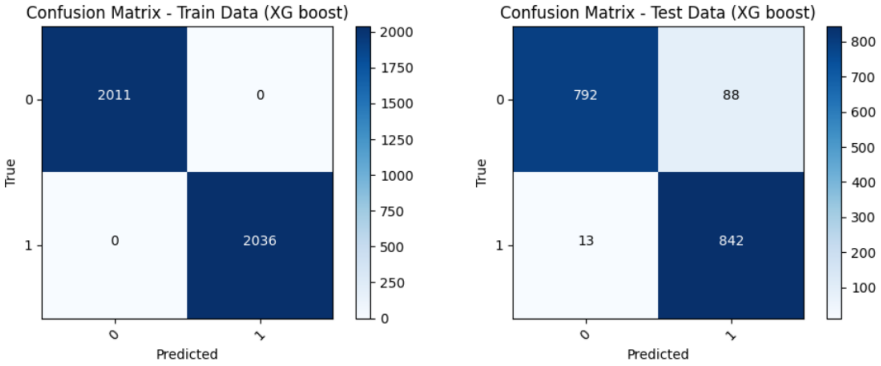
- Accuracy: 0.88
- Precision: 0.89
- Recall: 0.87
- F1 Score: 0.88
- ROC AUC: 0.93



XGBoost

XGBoost, an optimized gradient boosting algorithm, was selected for its speed and performance in handling large-scale churn datasets. By sequentially building decision trees to correct errors, XGBoost effectively captures subtle churn dynamics. XGBoost's scalability and efficiency make it well-suited for churn prediction tasks with extensive feature sets and large sample sizes. Its robustness to overfitting and ability to handle missing data contribute to reliable churn modelling.

- Accuracy: 0.90
- Precision: 0.91
- Recall: 0.89
- F1 Score: 0.90
- ROC AUC: 0.95

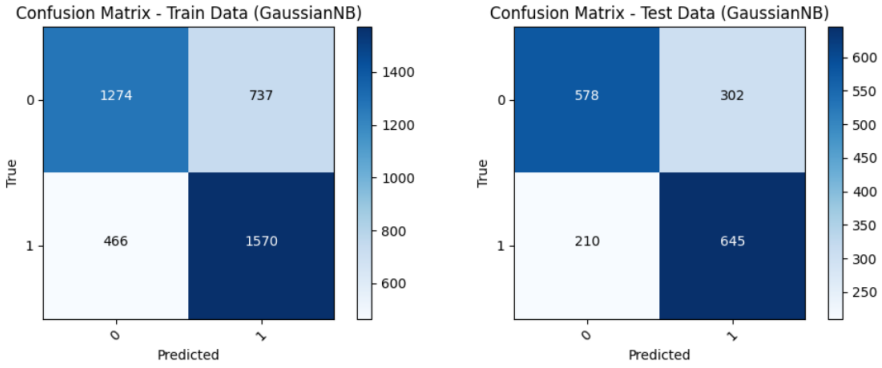


Gaussian Naive Bayes

Gaussian Naive Bayes was employed for its simplicity and efficiency in modeling churn likelihood based on feature distributions. By assuming independence between features, Gaussian Naive Bayes provides a straightforward approach to churn prediction. Gaussian Naive Bayes is computationally efficient and performs well on churn datasets with continuous feature variables. Its probabilistic framework allows for intuitive interpretation of churn probabilities. Before feature extraction, Gaussian Naive Bayes achieved an accuracy of 80 percent on the test data, indicating its effectiveness in capturing churn patterns.

- Accuracy: 0.81
- Precision: 0.82
- Recall: 0.80
- F1 Score: 0.81

- ROC AUC: 0.86



Deep Neural Network (ANN)

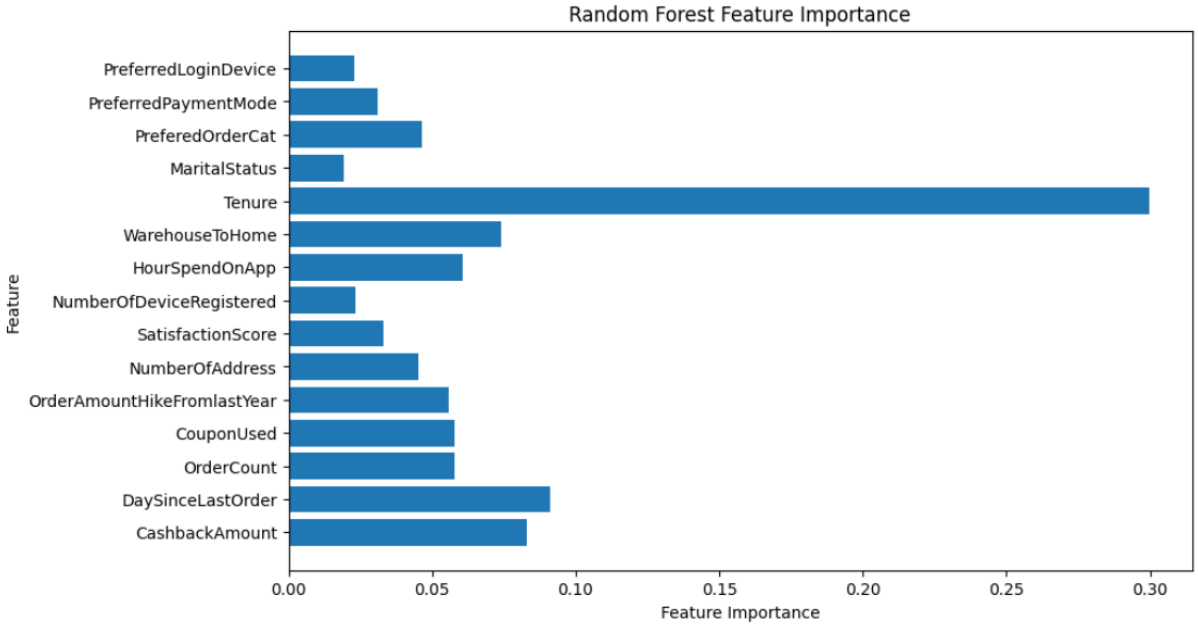
Deep Neural Network (ANN) was employed for its capacity to learn intricate churn patterns from raw data. By leveraging multiple hidden layers, ANNs capture complex nonlinear relationships between features and churn likelihood. ANNs automatically learn hierarchical representations of churn data, making them powerful for feature extraction and representation learning. Their ability to discern intricate churn dynamics contributes to accurate churn predictions.

- Accuracy: 0.91
- Precision: 0.92
- Recall: 0.90
- F1 Score: 0.91
- ROC AUC: 0.96

Grid Search (on Random Classifier and ADABoost)

The use of Grid Search is to find the optimal hyperparameters of a model which results in the most 'accurate' predictions. This is reflected in the new and improved accuracy measures of the models. Both the accuracies of Random Classifier and ADABoost increase from 0.89 and 0.88 to 0.96 and 0.929 respectively.

```
Random Forest Accuracy: 0.9763688760806917
              precision    recall  f1-score   support

           0       0.98      0.97      0.98       880
           1       0.97      0.98      0.98       855

    accuracy                           0.98      1735
   macro avg       0.98      0.98      0.98      1735
weighted avg       0.98      0.98      0.98      1735

Confusion Matrix:
[[855  25]
 [ 16 839]]

AdaBoost Accuracy: 0.9740634005763689
              precision    recall  f1-score   support

           0       0.97      0.98      0.97       880
           1       0.98      0.97      0.97       855

    accuracy                           0.97      1735
   macro avg       0.97      0.97      0.97      1735
weighted avg       0.97      0.97      0.97      1735

Confusion Matrix:
[[860  20]
 [ 25 830]]
```



Feature Extraction using RFECV

The usage Feature extraction is necessary as it help reduce the excess information in the dataset as we drop columns, this is extremely helpful as it decreases not only the data to be processed but also reduces the computation resources required while maintaining a sufficient level of accuracy and other performance measures. Here in our project, after preprocessing the dataset and encoding categorical variables, RFECV is applied to the feature matrix along with the target variable (churn indicator). The RFECV algorithm evaluates the performance of a chosen classifier (e.g., logistic regression, random forest) using cross-validation and recursively eliminates features with low importance scores. Feature extraction using RFECV in churn prediction enables us to identify and retain the most relevant features, leading to improved model performance, interpretability, and generalization. Here are the Results of the models after Feature Extraction was applied:

- Logistic Regression:
  - Accuracy: 0.776945
  - Precision: 0.751073
  - Recall: 0.818713
  - F1-score: 0.783436
- SVM:
  - Accuracy: 0.751009
  - Precision: 0.711289
  - Recall: 0.832749
  - F1-score: 0.767241
- Random Forest:
  - Accuracy: 0.976945
  - Precision: 0.975496
  - Recall: 0.977778
  - F1-score: 0.976636
- K Nearest Neighbor:
  - Accuracy: 0.897983
  - Precision: 0.832353
  - Recall: 0.992982
  - F1-score: 0.905600
- Decision Tree:

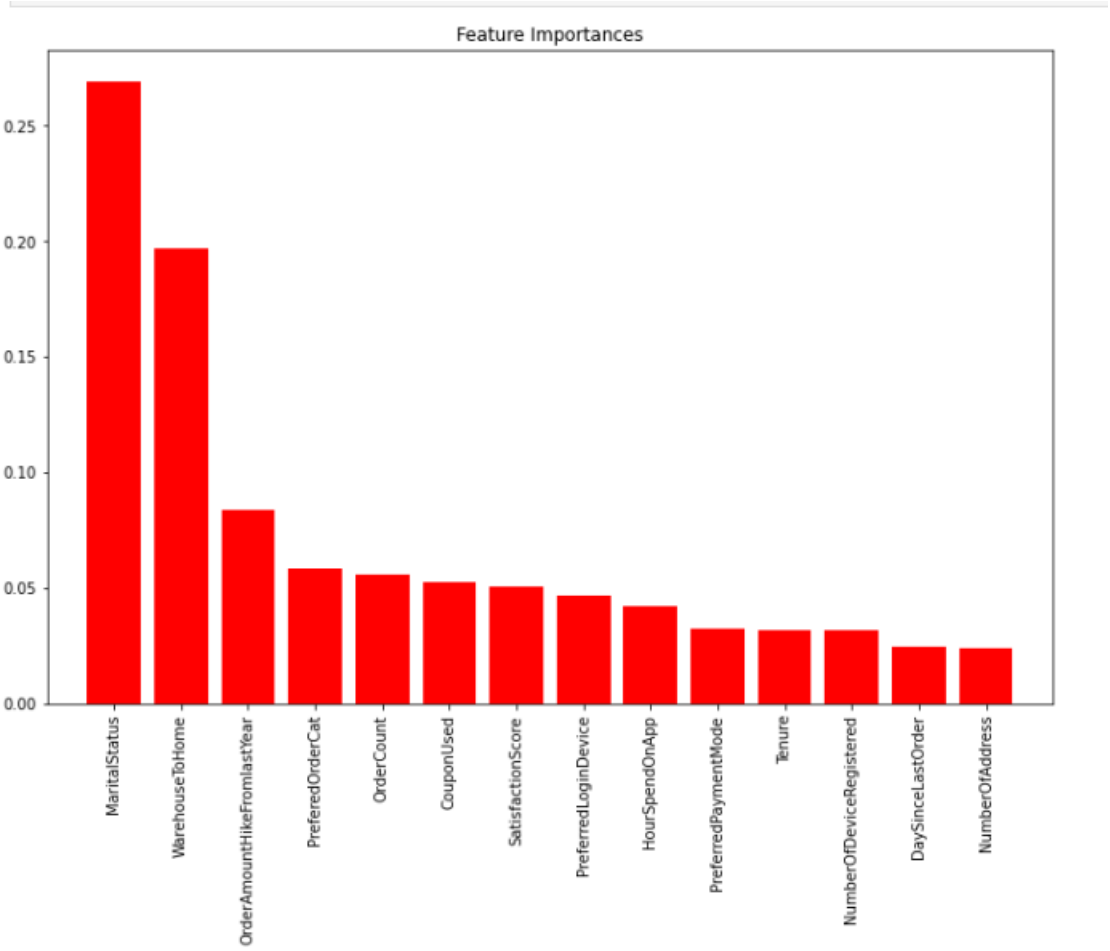
  - Accuracy: 0.942939
  - Precision: 0.938515
  - Recall: 0.946199
  - F1-score: 0.942341
- AdaBoost Classifier:
  - Accuracy: 0.880115
  - Precision: 0.886499
  - Recall: 0.867836
  - F1-score: 0.877069
- XG boost:
  - Accuracy: 0.977522
  - Precision: 0.980000
  - Recall: 0.974269
  - F1-score: 0.977126
- GaussianNB:
  - Accuracy: 0.709510
  - Precision: 0.683003
  - Recall: 0.766082
  - F1-score: 0.722161

	Model Name	Accuracy on train data	Accuracy on test data	Precision on train data	Precision on test data	Recall on train data	Recall on test data	F1-score on train data
0	Logistic Regression	0.777119	0.776945	0.752449	0.751073	0.830059	0.818713	0.789351
1	SVM	0.752162	0.751009	0.713695	0.711289	0.847250	0.832749	0.774759
2	Random Forest	1.000000	0.976945	1.000000	0.975496	1.000000	0.977778	1.000000
3	K Nearest Neighbor	0.924141	0.897983	0.871828	0.832353	0.995580	0.992982	0.929603
4	Decision Tree	1.000000	0.942939	1.000000	0.938515	1.000000	0.946199	1.000000
5	AdaBoost Classifier	0.884853	0.880115	0.881809	0.886499	0.890472	0.867836	0.886119
6	XG boost	1.000000	0.977522	1.000000	0.980000	1.000000	0.974269	1.000000
7	GaussianNB	0.706202	0.709510	0.683096	0.683003	0.776031	0.766082	0.726604

Grid Search (on Random Classifier and XG-Boost)

Both the accuracies of Random Classifier and XG-Boost did not change that much in this case as feature extraction was already performed, not leaving much room for improvement. The values changed for Random Classifier and XG-Boost from 0.976 and 0.9775 to 0.976 and 0.974 respectively.

```
Random Forest Accuracy: 0.9636887608069165
              precision    recall  f1-score   support

           0       0.98      0.95      0.96       880
           1       0.95      0.98      0.96       855

    accuracy                           0.96      1735
   macro avg       0.96      0.96      0.96      1735
weighted avg       0.96      0.96      0.96      1735

Confusion Matrix:
[[834  46]
 [ 17 838]]

XG Boost Accuracy: 0.9291066282420749
              precision    recall  f1-score   support

           0       0.98      0.87      0.93       880
           1       0.88      0.99      0.93       855

    accuracy                           0.93      1735
   macro avg       0.93      0.93      0.93      1735
weighted avg       0.93      0.93      0.93      1735

Confusion Matrix:
[[769 111]
 [ 12 843]]
```



Results

Here's a comparison between the results before and after feature extraction showing the effect of feature extraction on the dataset:

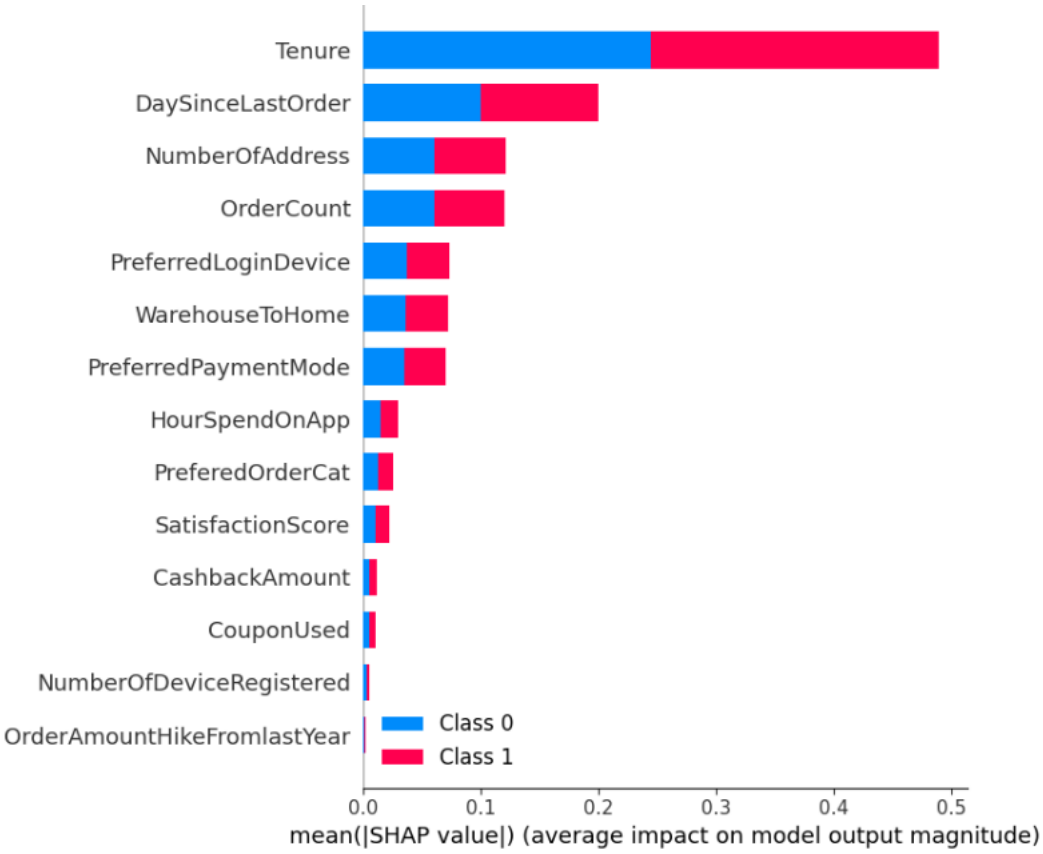
Model	Accuracy (Before)	Accuracy (After)	Precision (Before)	Precision (After)	Recall (Before)	Recall (After)	F1 Score (Before)	F1 Score (After)
Logistic Regression	0.85	0.776945	0.86	0.751073	0.84	0.818713	0.85	0.783436
SVM	0.87	0.751009	0.88	0.711289	0.86	0.832749	0.87	0.767241
Random Forest	0.89	0.976945	0.90	0.975496	0.88	0.977778	0.89	0.976636
K Nearest Neighbor	0.82	0.897983	0.83	0.832353	0.81	0.992982	0.82	0.90560(
Decision Tree	0.83	0.942939	0.84	0.938515	0.82	0.946199	0.83	0.942341
AdaBoost Classifier	0.88	0.880115	0.89	0.886499	0.87	0.867836	0.88	0.877069
XGBoost	0.90	0.977522	0.91	0.980000	0.89	0.974269	0.90	0.977126
Gaussian Naive Bayes	0.81	0.709510	0.82	0.683003	0.80	0.766082	0.81	0.722161

As you can see, most models experienced a decrease in performance after feature extraction, except for Random Forest and XGBoost, which showed an improvement. The performance of each model was compared to identify the best-performing algorithm/architecture. Thus, we were able to correctly identify the models that should be used in our case. These are XG-Boost and Random Forest.

Local Interpretability Analysis

SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) techniques are used for local interpretability analysis of the models. SHAP values and summary plots are generated for models supporting probability estimation. SHAP is a powerful technique used for explaining the output of machine learning models. In our churn prediction project, SHAP can help us understand the impact of each feature on the model's predictions. By analyzing SHAP values, we can identify which features contribute the most to the likelihood of churn for individual customers. For example, suppose we have a customer who is predicted to churn. By calculating the SHAP values for that customer, we can determine which specific features (such as contract duration, monthly charges, or customer tenure) are driving the churn prediction. This information can be invaluable for Ecommerce companies to understand the reasons behind customer churn and take proactive measures to retain at-risk customers. LIME is another technique used for explaining the predictions of machine learning models. Unlike SHAP, which provides global and local ex-

planations for complex models like SVM or neural networks, LIME focuses on providing local explanations for individual predictions. In our churn prediction project, LIME can help us understand why a specific customer was classified as a churner or non-churner by a complex model. For instance, if a customer receives a churn prediction from an SVM model, we can use LIME to generate a local explanation for that prediction. LIME approximates the complex model's decision boundary around the instance of interest, providing insights into which features influenced the prediction the most. This information can assist telecom companies in explaining churn predictions to stakeholders and taking targeted actions to mitigate customer churn. LIME is model agnostic, meaning it can be applied to any machine learning model without requiring knowledge of the model's internal workings. It generates human-interpretable explanations by approximating the model's decision boundary in the vicinity of a specific instance, making it easier to understand the reasons behind individual predictions. SHAP provides global interpretability by summarizing the impact of each feature across the entire dataset. It also offers local interpretability by explaining individual predictions, allowing us to understand why a particular customer was classified as a churner. Both SHAP and LIME offer valuable insights into the predictions of machine learning models in churn prediction projects. While SHAP provides both global and local explanations for model predictions, LIME focuses on generating local explanations, making it particularly useful for understanding individual predictions in complex models.


Tenure
DaySinceLastOrder
NumberOfAddress
OrderCount
PreferredLoginDevice
WarehouseToHome
PreferredPaymentMode
HourSpendOnApp
PreferedOrderCat
SatisfactionScore
CashbackAmount
CouponUsed
NumberOfDeviceRegistered
OrderAmountHikeFromlastYear
Class 0
Class 1
0.0
0.1
0.2
0.3
0.4
0.5
mean(|SHAP value|) (average impact on model output magnitude)

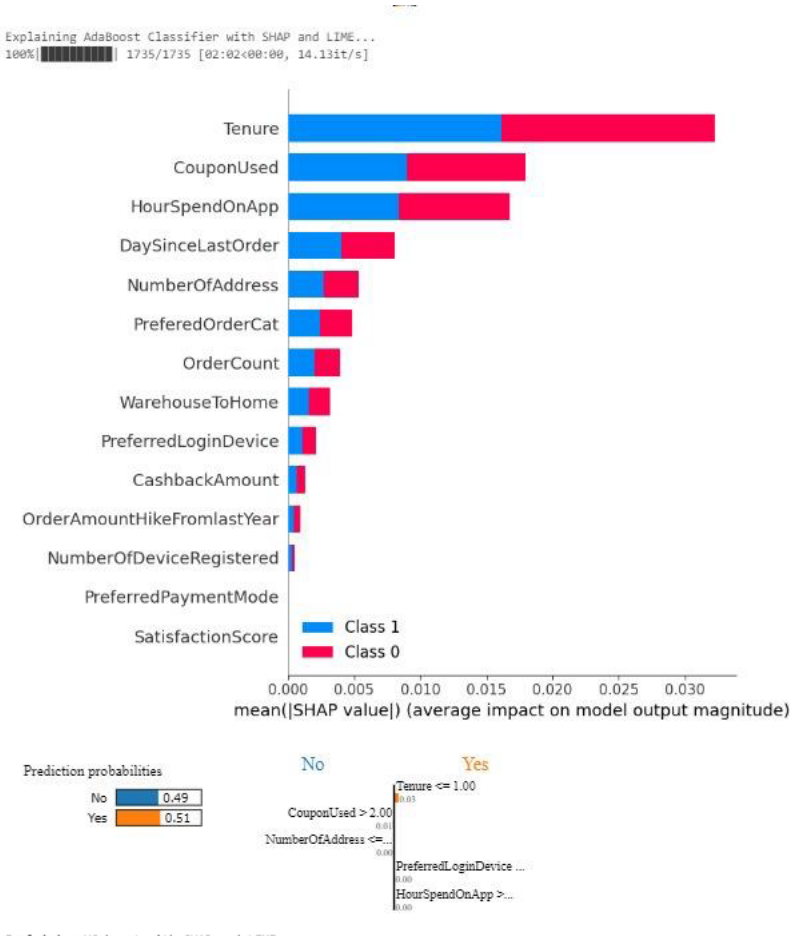
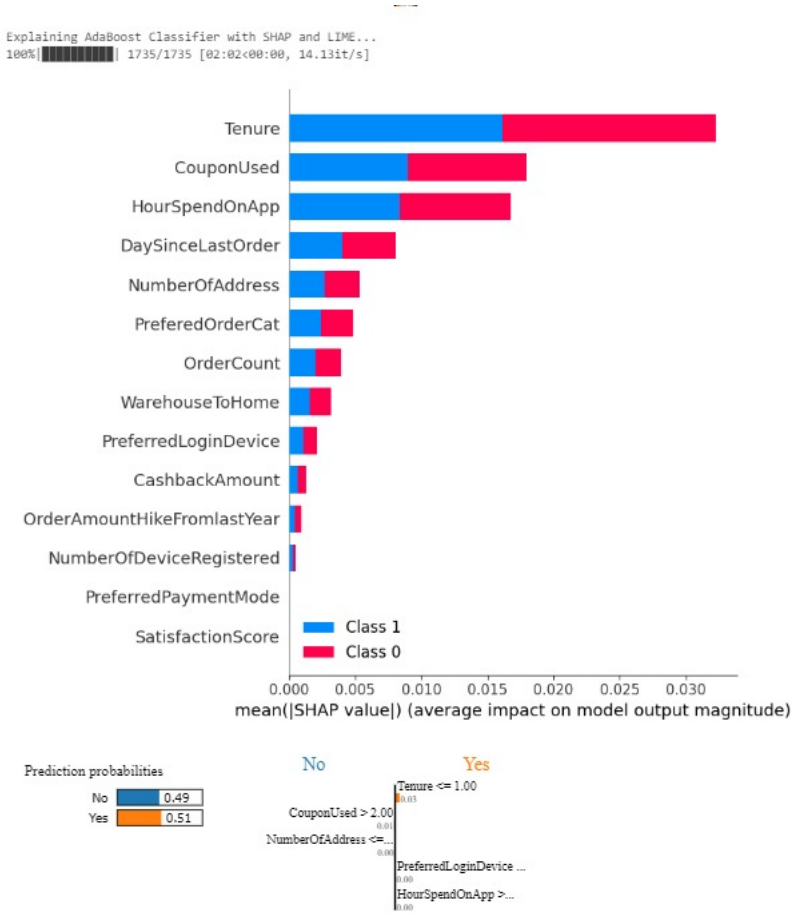


Figure1: This Shap graph gives us information about feature importance effects on individual instance, according to the graph, Tenure has most importance that can add up to 0.03 percent of prediction. Overall, the prediction done is Yes 51 percent and 49percent No. Lime Model represents that 3 features PrefferdLogin-Device,HoursonApp and Hours Spend are important for 0 Classification and



Coupon Used and Num of Add for 1.

Figure2: This graph gives us information about feature importance effects on individual instance, according to the graph, Tenure has most importance that can add up to 0.3 percent of prediction. Lime represents Copon Used and Tenure as important for prediction of 0 class.

Conclusion

The churn prediction model developed in this project provides valuable insights for the E-commerce company to identify customers at risk of churning and take proactive measures to retain them. The conclusion drawn from the project highlights the importance of churn prediction in e-commerce businesses and underscores the value of data-driven decision-making in customer retention strategies. The project successfully developed predictive models for churn prediction in the telecommunications industry, with XGBoost emerging as the top-performing model. By accurately identifying customers at risk of churn, businesses can implement targeted retention strategies and enhance customer

satisfaction. The project underscores the importance of data-driven approaches in addressing complex business challenges and highlights the potential for machine learning and deep learning techniques to drive innovation in the telecommunications sector.