CAPSTONE PROJECT

CUSTOMER CHURN PREDICTION

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION:

1.1.Defining problem statement:

The E-commerce industry has experienced rapid growth, driven by digital transformation, increasing internet penetration and the rise of mobile commerce. However, with intense competition and ever-changing customer preferences, retaining customers has become a major challenge. Customer churn, defined as customers who stop purchasing or engaging with an e-commerce platform over a specific period, directly impacts business profitability.

Studies indicate that acquiring a new customer can be more expensive than retaining an existing one, making churn reduction a critical focus for e-commerce businesses.High churn rates lead to revenue loss, increased marketing costs and reduced customer lifetime value.

1.2.Need for the project:

An E-Commerce company provider is facing a lot of competition in the current market and it has become a challenge to retain the existing customers in the current situation. Hence, the company wants to develop a model through which they can do churn prediction of the accounts and provide segmented offers to the potential churners. In this company, account churn is a major thing because 1 account can have multiple customers,hence by losing one account the company might be losing more than one customer.

This predictive capability will enable proactive retention strategies, allowing the business to take targeted actions such as personalized offers, improved customer support and tailored marketing campaigns to reduce churn and enhance customer loyalty.

1.3.Understanding business opportunity:

Customer churn is a critical challenge for businesses, as retaining existing customers is often more cost-effective than acquiring new ones. A predictive churn model presents a significant business opportunity by enabling companies to proactively identify at-risk customers and implement targeted retention strategies. By leveraging machine learning, businesses can:

- Early identification of churn-prone customers allows businesses to offer personalized incentives, loyalty programs or improved services to retain them.

- Reducing churn directly contributes to higher customer lifetime value (CLV) and overall business growth.
- Instead of blanket campaigns, marketing teams can focus resources on customers who need engagement the most.

2.EXPLORATORY DATA ANALYSIS AND BUSINESS IMPLICATIONS

2.1.Non visual understanding of data:

First 5 rows of the dataset

	AccountID	Churn	Tenure	City_Tier	CC_Contacted_LY	Payment	Gender
0	20000	1	4	3	6	Debit Card	Female
1	20001	1	0	1	8	UPI	Male
2	20002	1	0	1	30	Debit Card	Male
3	20003	1	0	3	15	Debit Card	Male
4	20004	1	0	1	12	Credit Card	Male

Table 1: First 5 rows

The dataset has been loaded successfully.The dataset has 11260 rows and 19 columns.

Descriptive Details

	AccountID	Churn	City_Tier	CC_Contacted_LY	Service_Score	CC_Agent_Score	Complain_ly
Count	11260.00	11260.00	11148.00	11158.00	11162.00	11144.00	10903.00
mean	25629.50	0.1683	1.6539	17.8670	2.9025	3.0664	0.2853
std	3250.62	0.3742	0.9150	8.8532	0.7255	1.3797	0.4515
min	20000.00	0.0000	1.0000	4.0000	0.0000	1.0000	0.0000
25%	22814.75	0.0000	1.0000	11.0000	2.0000	2.0000	0.0000
50%	25629.50	0.0000	1.0000	16.0000	3.0000	3.0000	0.0000
75%	28444.25	0.0000	3.0000	23.0000	4.0000	4.0000	1.0000
max	31259.00	1.0000	3.0000	132.0000	5.0000	5.0000	1.0000

Table 2: Descriptive Details

Observation:

- Some columns have missing values, requiring imputation or handling before modeling.

- Around 16.8% of customers have churned, which could indicate class imbalance.
- Some customers have contacted support up to 132 times, possibly indicating dissatisfaction.
- While most customers rated services positively, 28.5% lodged complaints, which could be a key churn predictor.
- CC_Contacted_LY and Complain_ly are right-skewed variables since their mean is greater than the median.

Understanding of variables

#	Column	Non-Null	Count	Dtype
0	AccountID	11260	non-null	int64
1	Churn	11260	non-null	int64
2	Tenure	11042	non-null	float64
3	City_Tier	11148	non-null	float64
4	CC_Contacted_LY	11158	non-null	float64
5	Payment	11151	non-null	object
6	Gender	11152	non-null	object
7	Service_Score	11162	non-null	float64
8	Account_user_count	10816	non-null	float64
9	account_segment	11163	non-null	object
10	CC_Agent_Score	11144	non-null	float64
11	Marital_Status	11048	non-null	object
12	rev_per_month	10469	non-null	float64
13	Complain_ly	10903	non-null	float64
14	rev_growth_yoy	11257	non-null	float64
15	coupon_used_for_payment	11257	non-null	float64
16	Day_Since_CC_connect	10902	non-null	float64
17	cashback	10787	non-null	float64
18	Login_device	11039	non-null	object

Table 3 : Data Info

Observation:

- There are columns with missing values,which needs to be addressed.
- Some columns with object type may actually be numerical but stored as strings . These should be checked and converted if needed.

2.2.Visual understanding of data

Univariate Analysis:

We will analyze the distribution of independent variables.It will help us identify the pattern among the variables and the effects they have on target variable.

First,lets see how the target variable (Churn) is distrubuted.

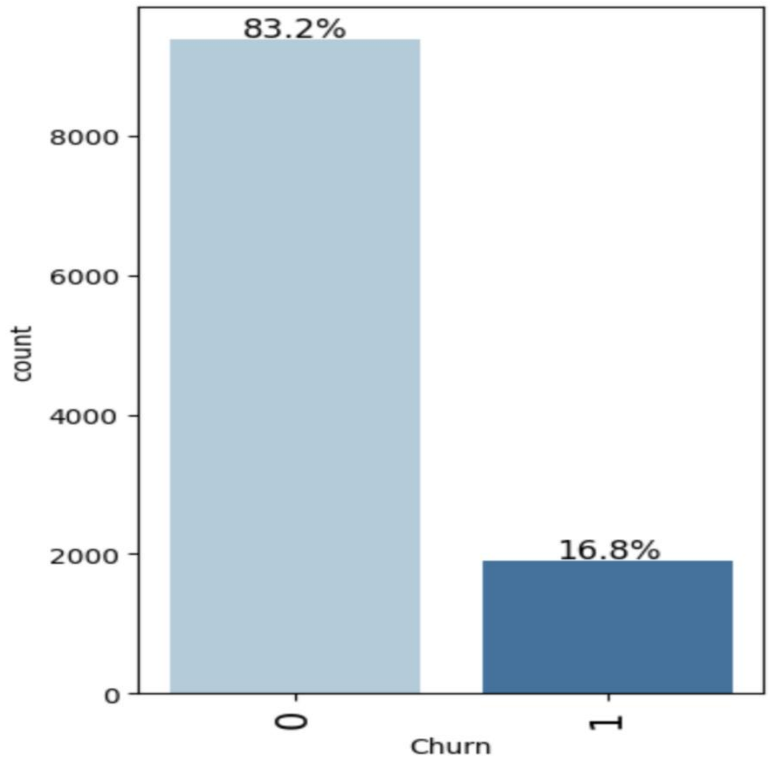

Fig 1: Countplot of the target variable(Churn)

- From the above plot we can see that the target variable is unevenly distributed.The ratio is about 83:17.Which indicates that the dataset is imbalanced.

Observation on Tenure

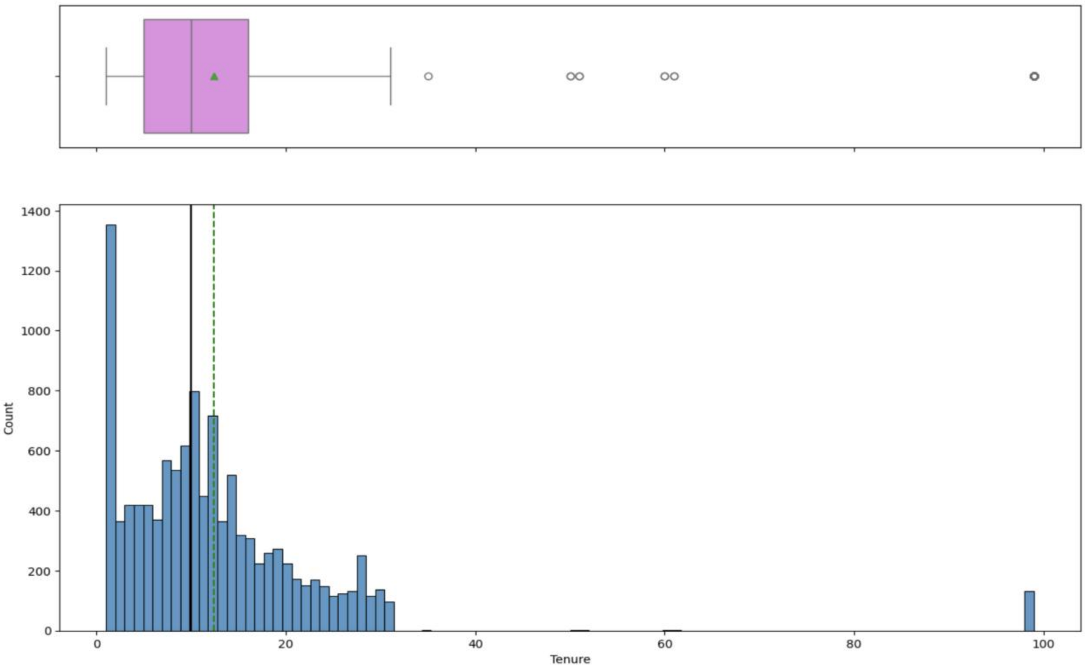

Fig 2: Observation on Tenure

- The distribution is right-skewed,significant percentage of customer hold a lower tenure of 1 month,this could be an indication of new customers.
- The average tenure among the customers is at 12.

Observation on CC_Contacted_LY

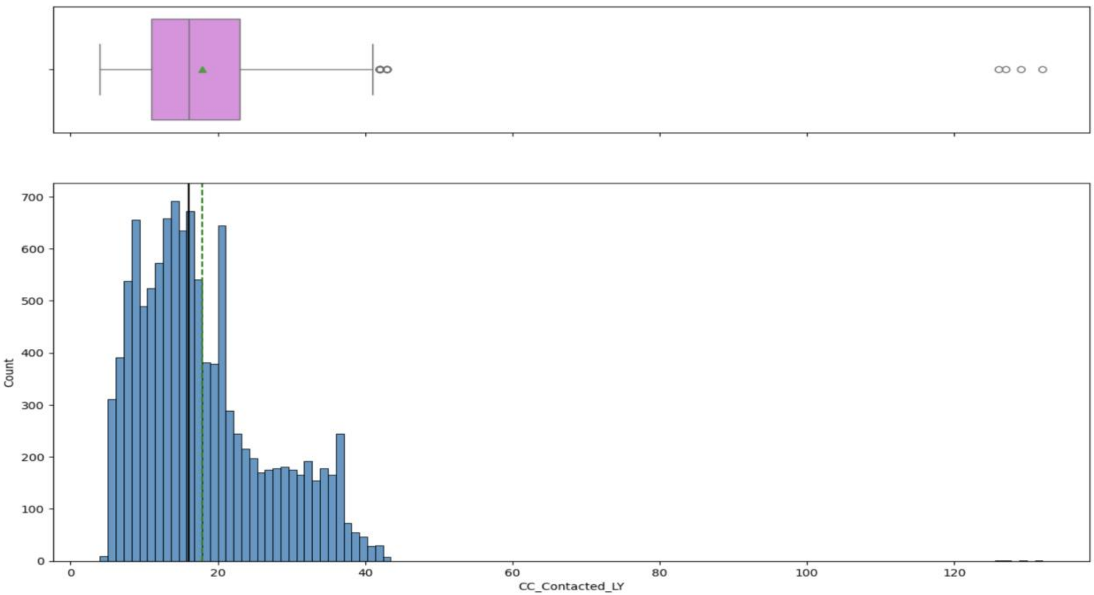

Fig 3: Observation on CC_Contacted_Ly

- Most customers have contacted customer care fewer times, with a peak around 15-20 contacts.
- A few customers have contacted customer care excessively, as indicated by the extreme outliers beyond 60.

Observation on rev_per_month

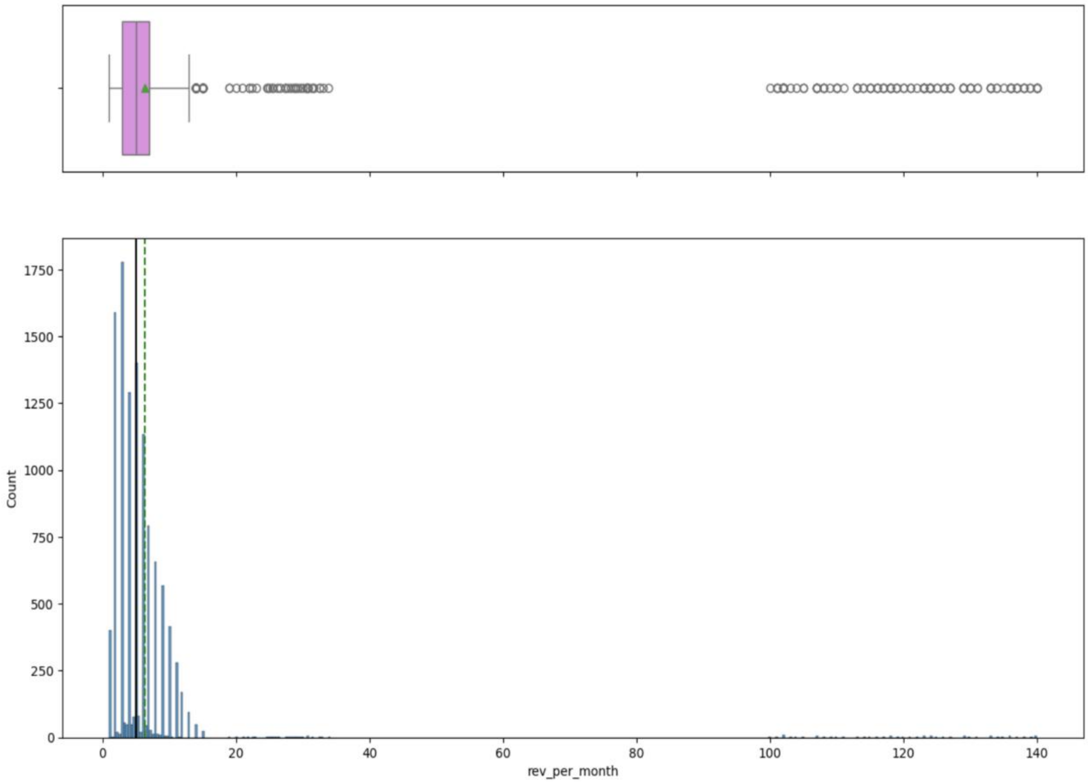

Fig 4: Observation on rev_per_month

- Most accounts generate low monthly revenue, clustered between 1 to 10.
- A few accounts have extremely high revenues, as shown by the significant number of outliers beyond 40.

Observation on rev_growth_yoy

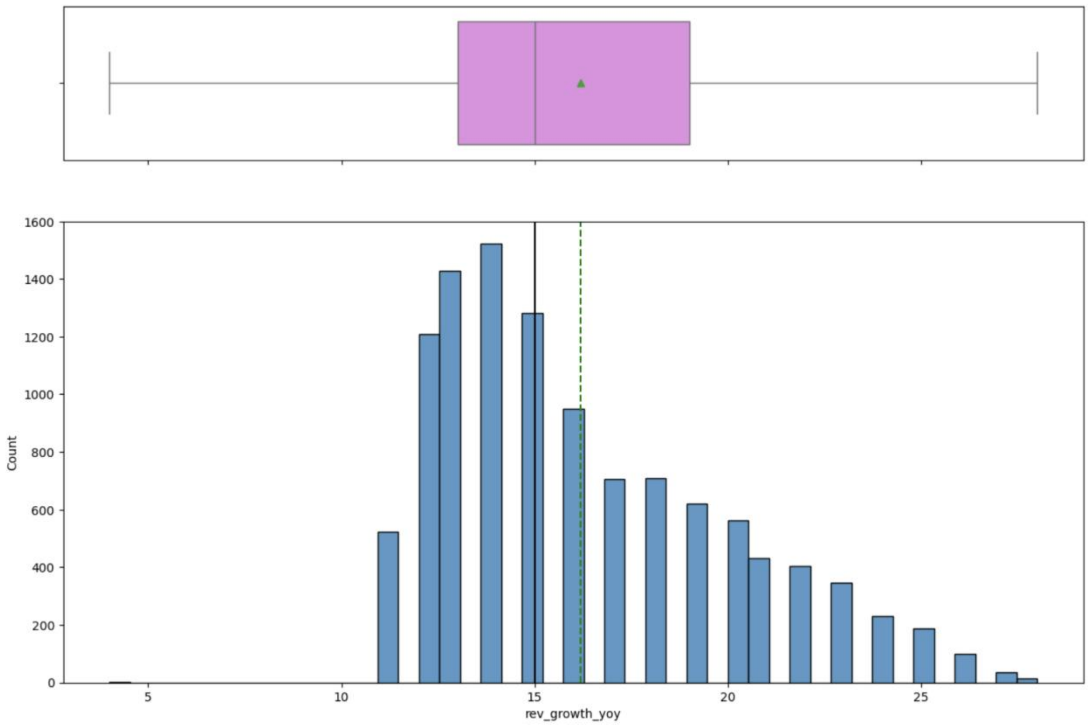

Fig 5: Observation on rev_growth_yoy

- There is no indication of negative values ,this shows existing customers contribution is better compared to previous year.

Observation on Day_Since_CC_connect

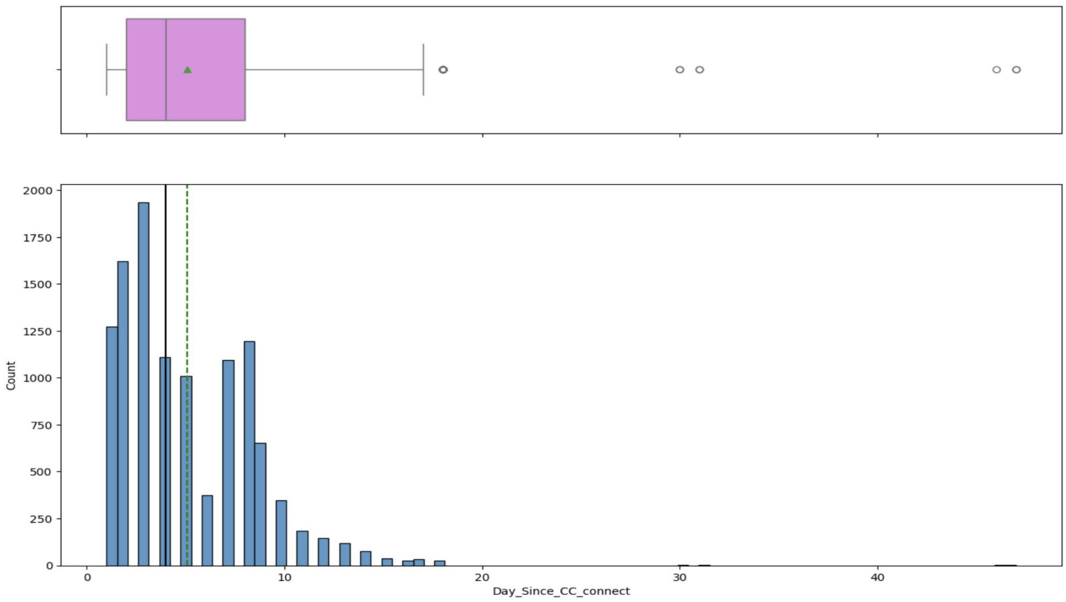

Fig 6: Observation on Day_Since_CC_connect

- There are only few customers who have not contacted the customer care between 1-20 days.

Observation on cashback

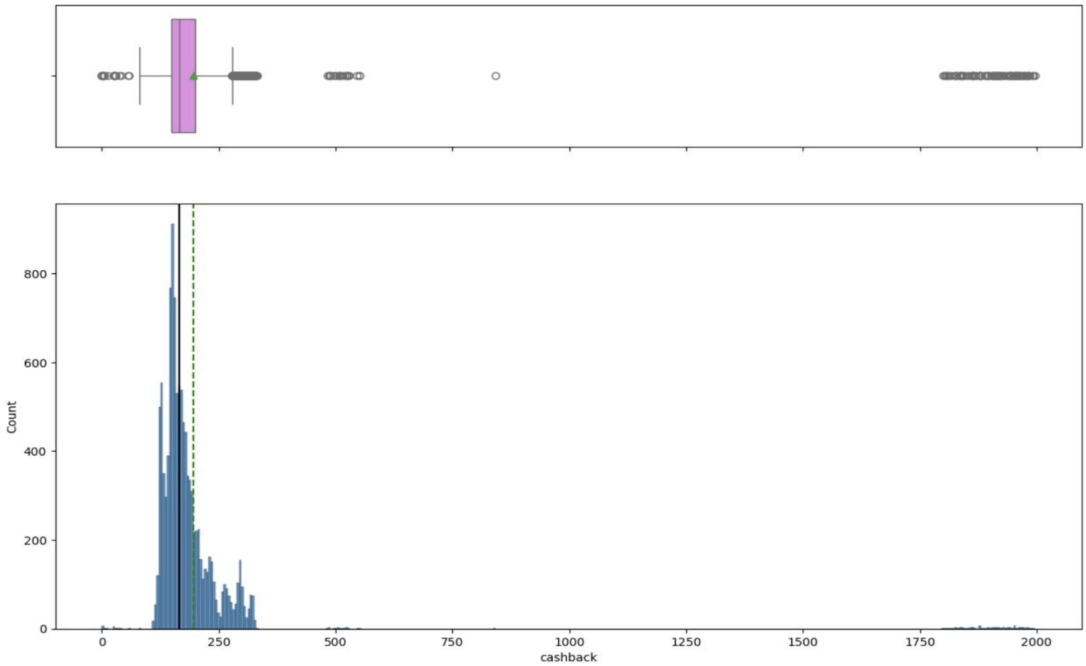

Fig 7: Observation on cashback

- Most customers receive lower cashback amounts, while a few receive significantly high amounts.
- Presence of outliers suggests that a small group of customers might be getting exceptionally high cashback.

Observation on City_Tier

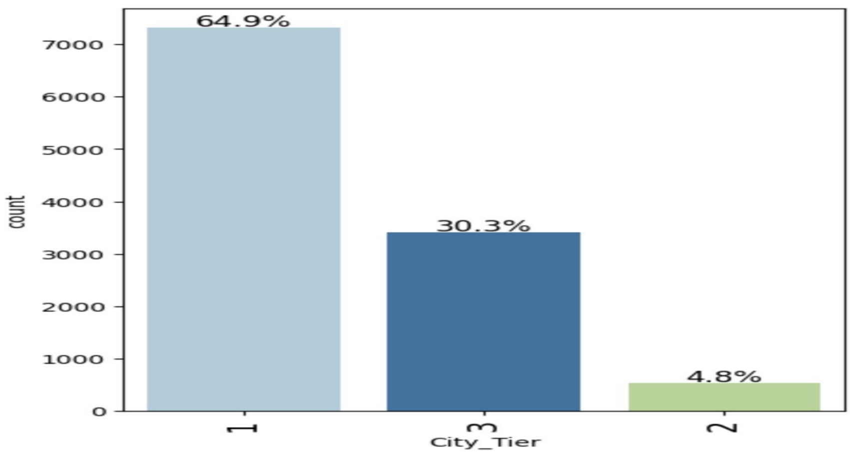

Fig 8: Countplot on City_Tier

- Most customers are from Tier 1 with approximately 65% with only 5% from Tier 2.

Observation on Payment

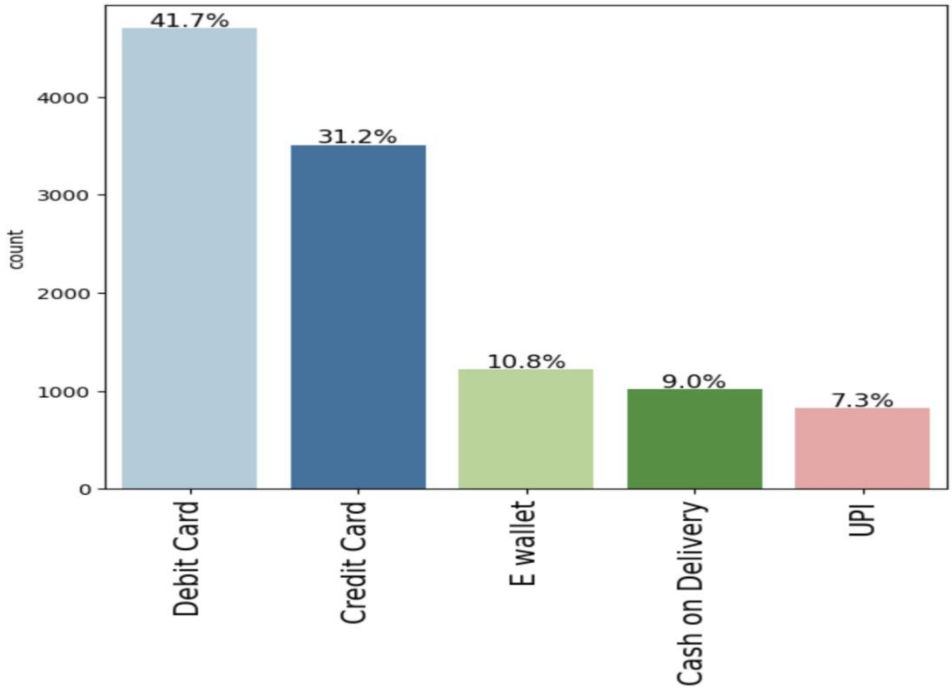

Fig 9: Countplot on Payment

- Approximately 42% of the customer have choosed their mode of payment as Debit card following 32% with Credit card.Customer might feel that card payment as more secured way compared to other payment methods.

Observation on Service_Score

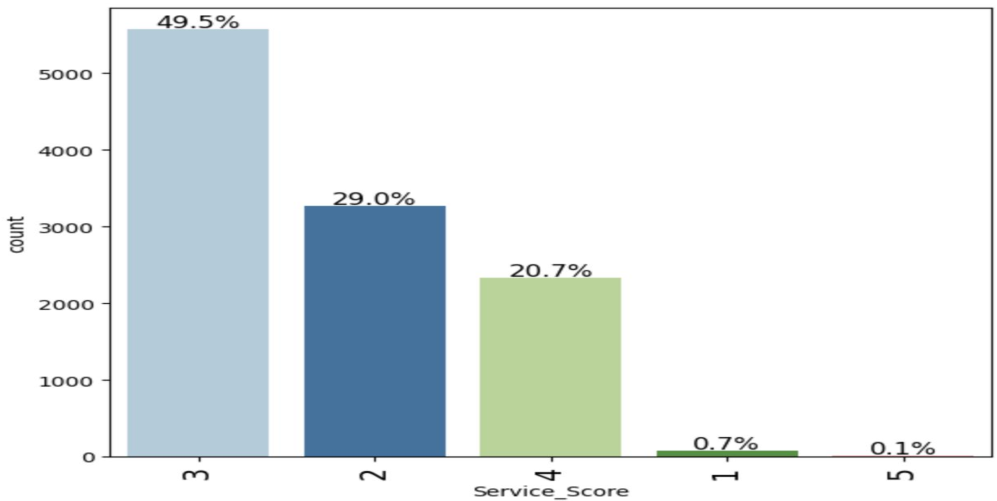

Fig 10: Countplot on Service_Score

- Approximately 21% of the customers have rated the service provided by the company as 4 and 5.

Observation on Account_user_count

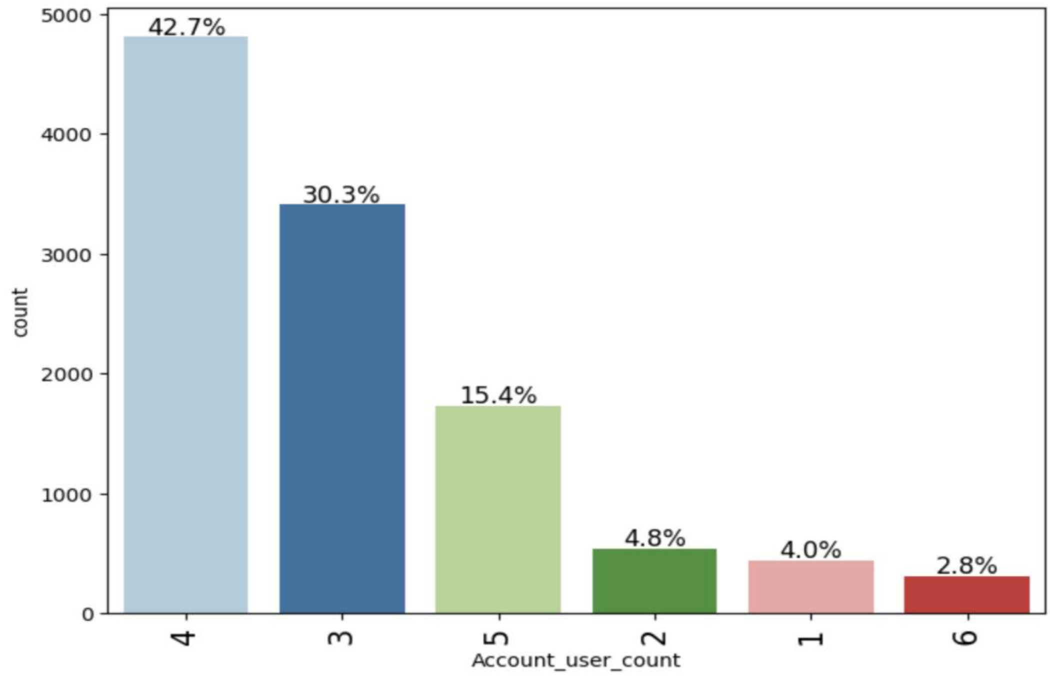

Fig 11: Countplot on Account_user_count

- Only 4% of the customers have not shared their account to any other users.

Observation on account_segment

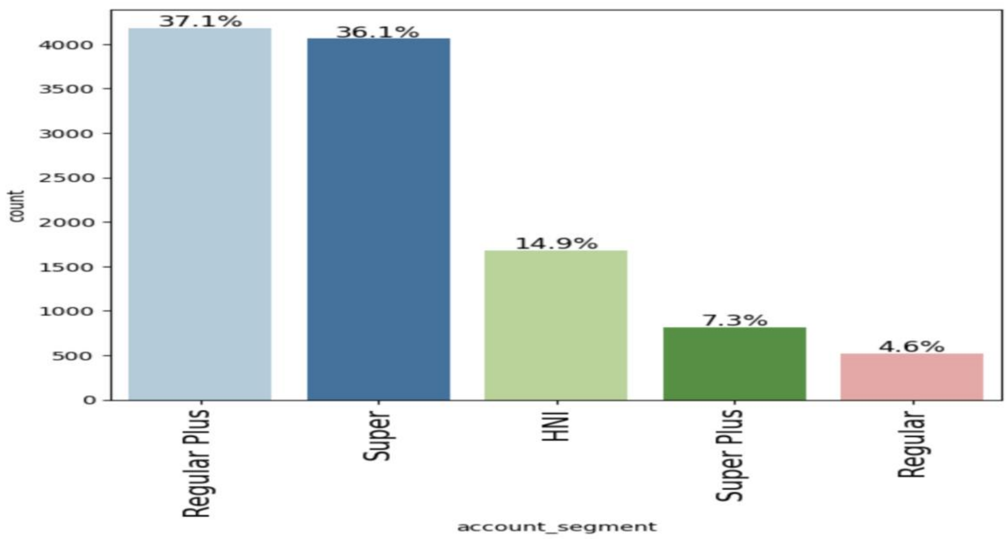

Fig 12: Countplot on account_segment

- Approximately 15% fall under HNI and 4.6% into Regular.With majority of the customer are of Regular Plus segment.

Observation on Martial_Status

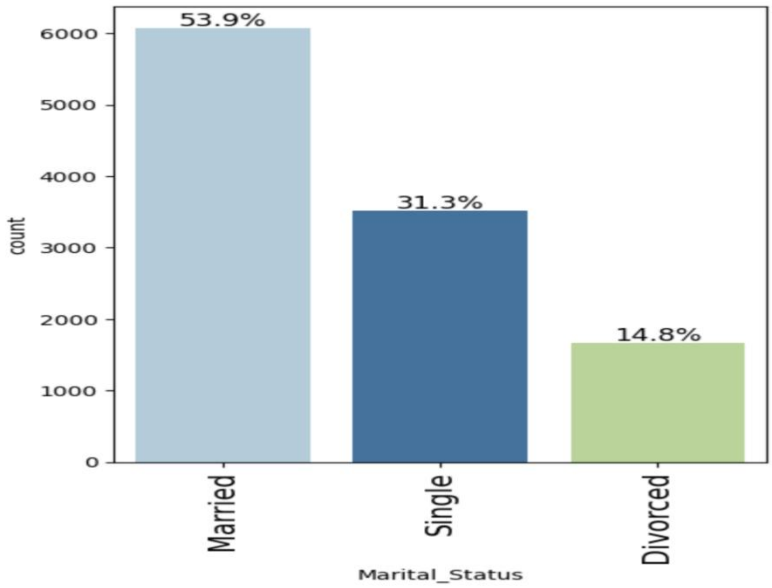

Fig 13: Countplot on Martial_Status

- Approximately 54% of customers are married.

Observation on Complain_ly

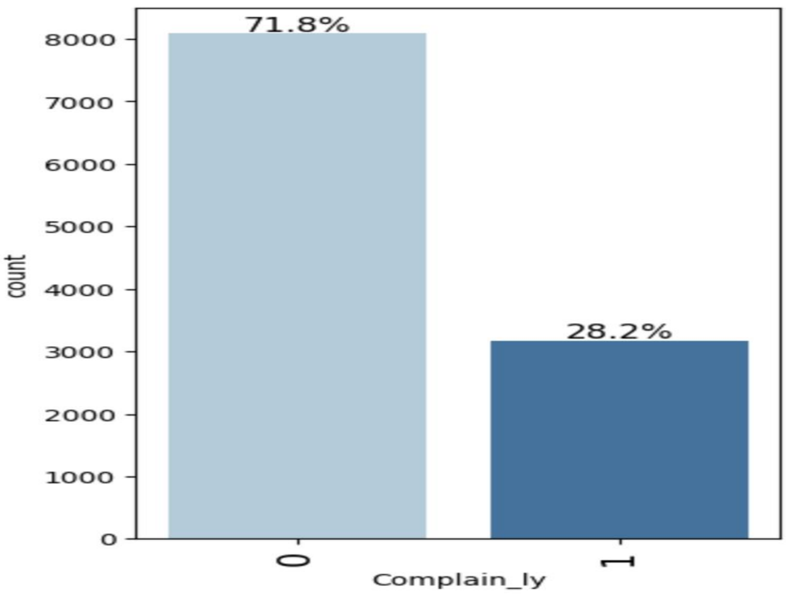

Fig 14: Countplot on Complain_ly

- Approximately 72% of customers have not made any complaints in the last 12 months, which is a positive sign for the company. However, the remaining 28% should be addressed promptly, as resolving their concerns on the first request can enhance customer satisfaction.

Observation on coupon_used_for_payment

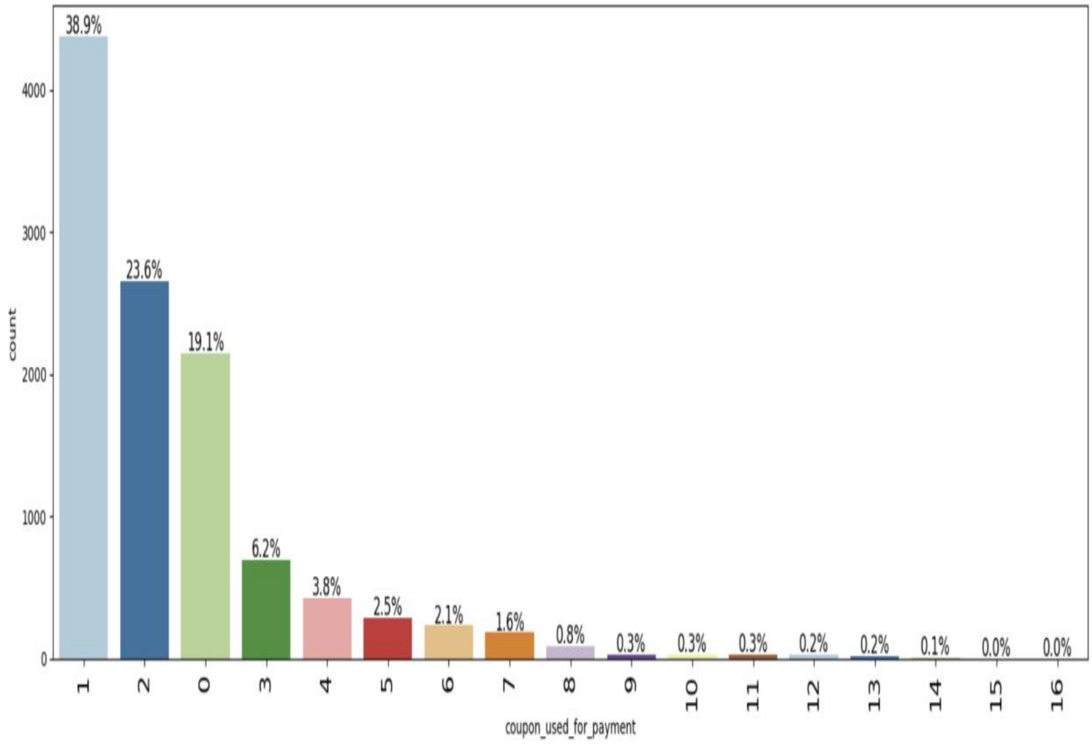

Fig 15: Countplot on coupon_used_for_payment

- Approximately 60% of the customers has used 1 or 2 coupons and 19% have not used coupons.

Observation on Login_device

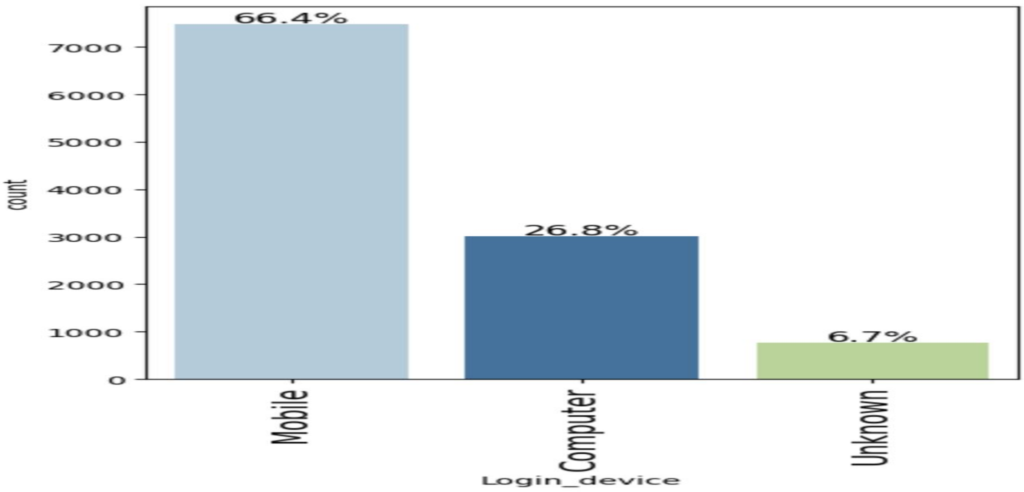

Fig 16: Countplot on Login_device

- It is understandable that most customer have logged in with mobile.
- This gives the fair information how effective cc_agent_score,complain_ly and service_score variables on target varible.Approximately 40% of the customers have churned when they have raised a complain.

Bivariate Analysis:

Understanding Tenure vs Churn

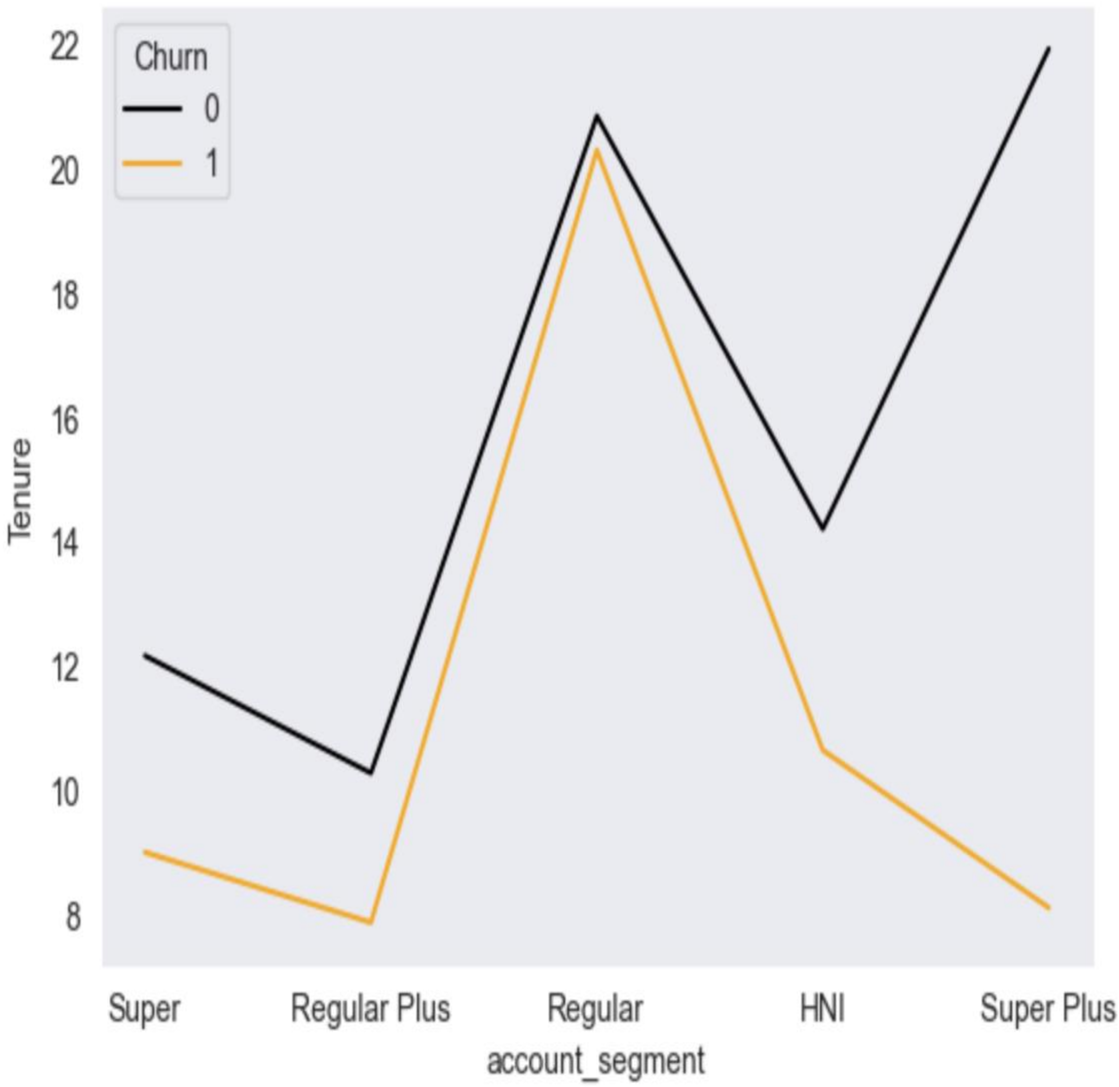

Fig 17:Tenure vs Churn

- Churners have low tenure than non churners.More the tenure ,the customers are less likely to churn.Less the tenure,more likely the customers churn.
- The regular segment customer show highest tenure for churners and non-churners.

Understanding Complain vs Churn

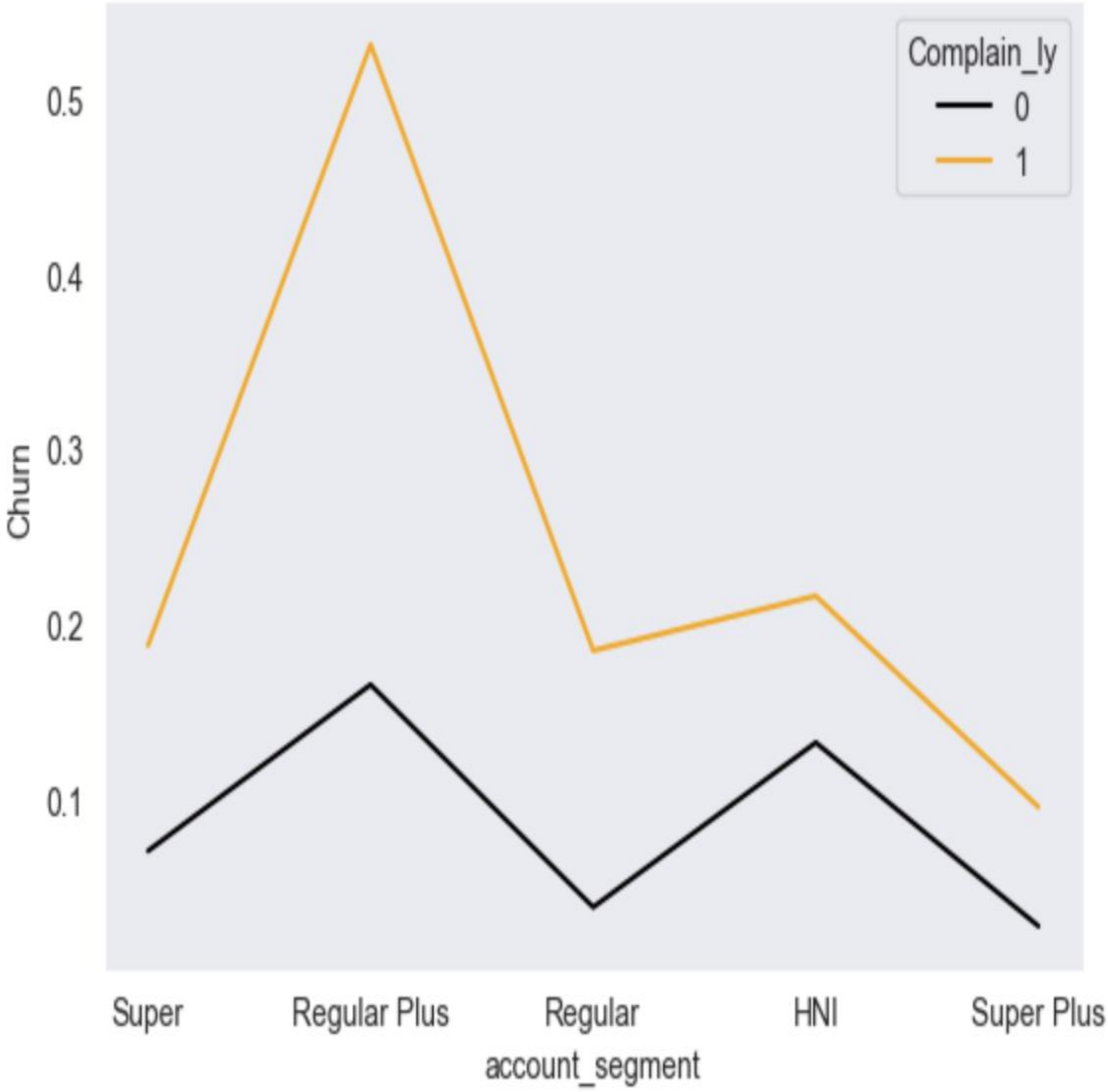

Fig 18:Complain vs Churn

- When complain raised by the customer,more likely the customer churns and when complain is not raised by the customer less likely the customer churns.
- Regular plus customer segment has the highest churn,around 50-55% of the customers in this category has churned.

Multivariate Analysis:

Heatmap

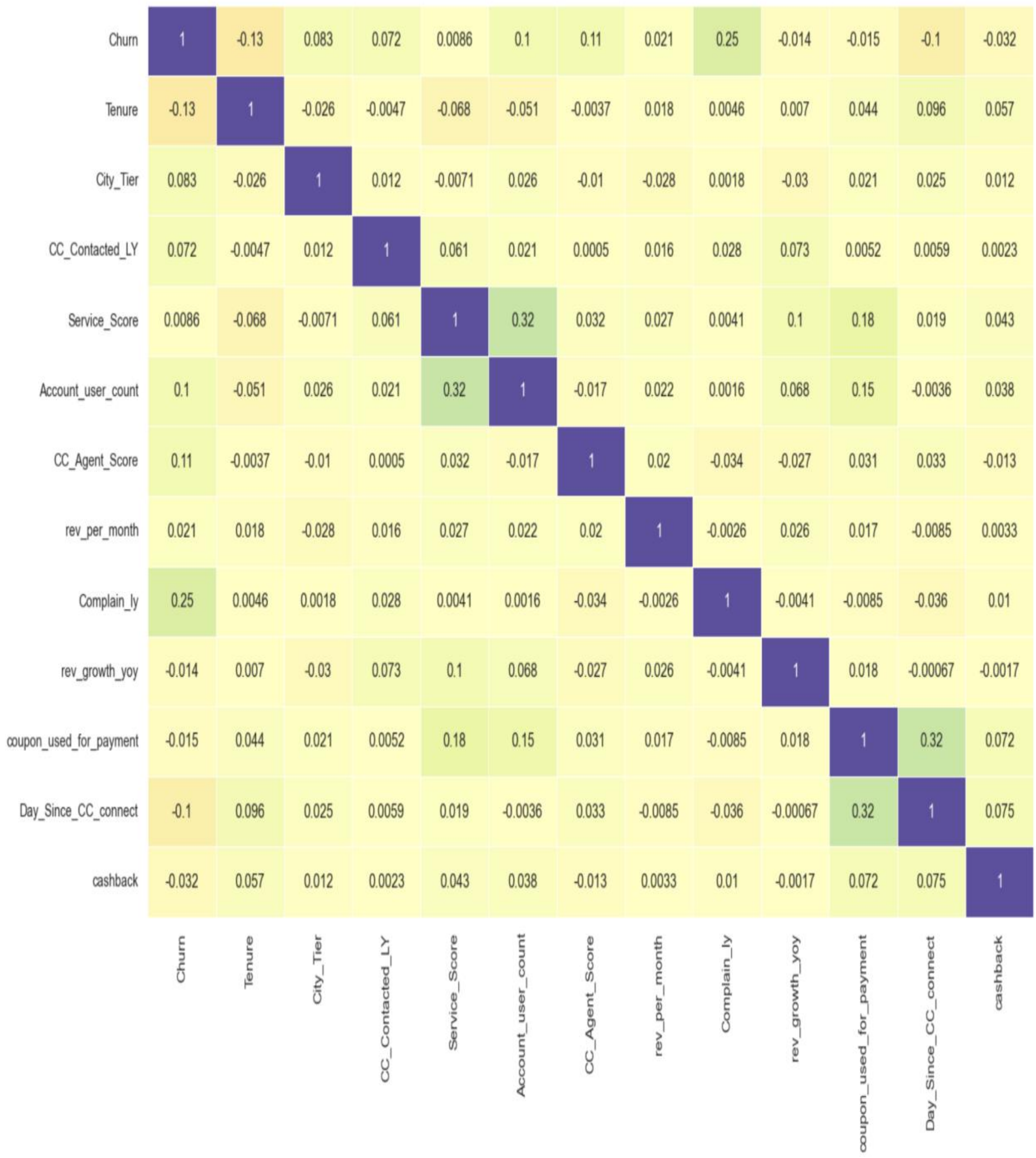

Fig 19: Heatmap

- The are no highly correlated variable.
- The variable churn and Tenure indicate a weak negative correlation.

Pairplot

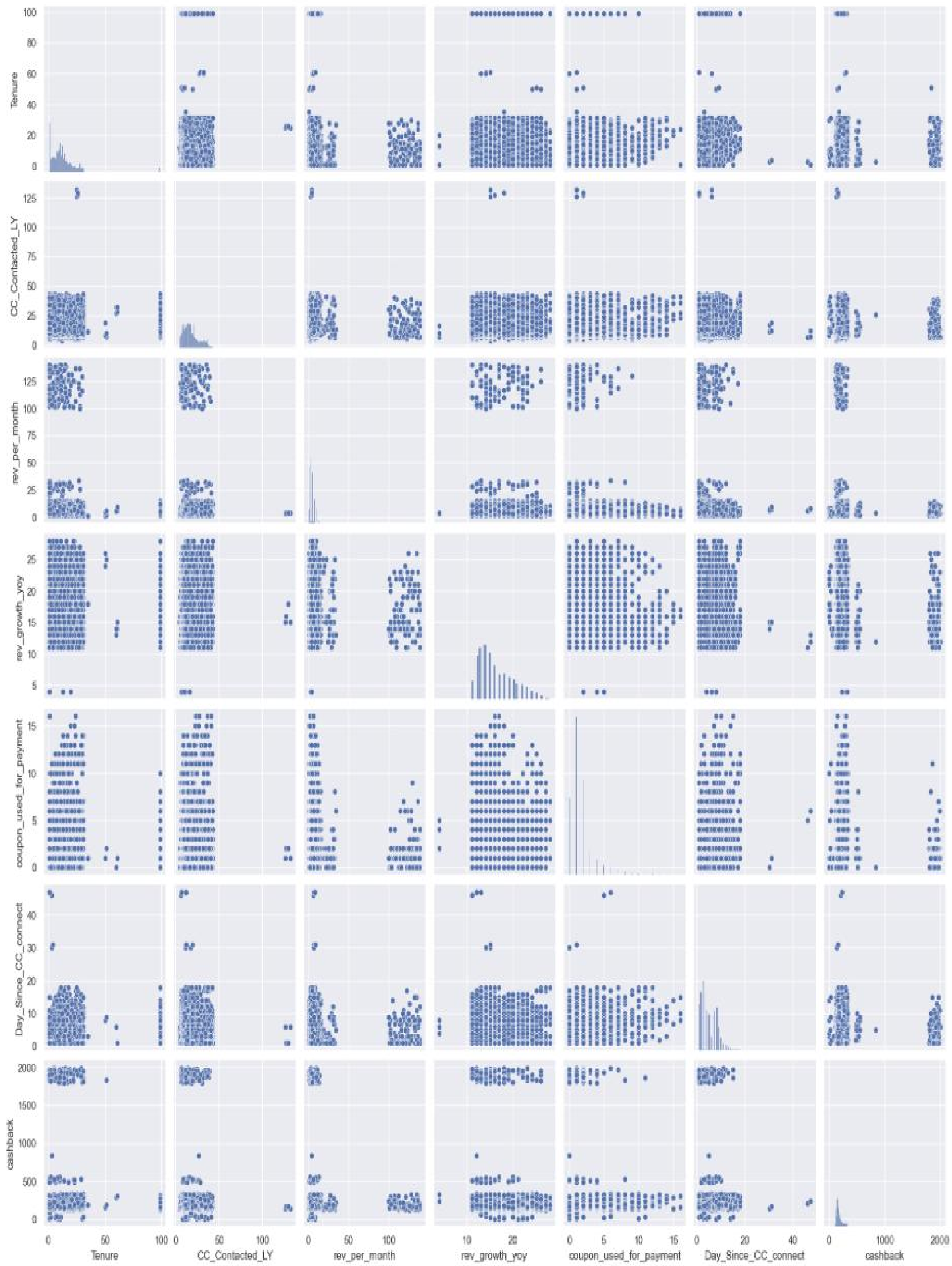

Fig 20: Pairplot

- There is no relationship between any numerical values.

Insights from EDA and its impact on business

- 72% of customers have not raised a complain in the last 12 months, which is a positive sign. However, 40% of customers who raised complaints have churned, highlighting a strong link between dissatisfaction and attrition. Ensuring with first-time resolution of complaints and implementing proactive support strategies such as feedback surveys and personalized problem-solving.
- Most customers are from Tier 1 cities (65%), with only 5% from Tier 2. This suggests a high digital adoption rate and urban market concentration.Consider in expanding services in Tier 2 and Tier 3 cities through targeted marketing campaigns and localized promotions.
- Only 21% of customers have rated the service as 4 or 5 stars, indicating room for improvement in customer experience.Investing in service quality training could improve rating.
- Focus on early engagement strategies, such as personalized welcome offers and proactive customer support, to reduce churn in new customers.
- Most customers contact customer care fewer times, but a small group contacts excessively (outliers beyond 60 times).Priority handling for frequent complainants to improve efficiency.
- Customers with low tenure are less likely to churn and customer with high tenure more likely to churn.
- The Business has to be cautious about the account segment 'Regular 'and 'Regular plus'.

Segmentation made from EDA:

Segment 1: Low Tenure,No complain raised and higher customer care score are less likely to churn.

Segment 2: High Tenure,complain raised and lower customer care score are more likely to churn.

Impact : These insights will certainly help the business stay competitive in the market by making it easier to identify potential churners. With this information, the business can proactively target and retain at-risk customers, improving customer retention and reducing churn.

Identifying and retaining customers at risk of churning has a direct impact on revenue. It protects the business from revenue loss, as losing customers directly affects overall profitability.

3.DATA CLEANING AND PRE-PROCESSING:

3.1.Approach used for identifying and treating missing values and outlier treatment:

Missing value check

Churn	0
Tenure	0
City_Tier	112
CC_Contacted_LY	102
Payment	109
Gender	108
Service_Score	98
Account_user_count	0
account_segment	97
CC_Agent_Score	116
Marital_Status	212
rev_per_month	0
Complain_l12m	357
rev_growth_yoy	0
coupoun_used_for_payment	0
Day_Since_CC_connect	0
cashback_l12m	473
Login_device	221

Table 4: Missing value check

The dataset contains missing values, bad values and anomalies that needed to be handled appropriately.

- The variable Login_device had values represented as &&&&, which were converted to Unknown.
- Variables Tenure, Account_user_count, rev_per_month, rev_growth_yoy, coupon_used_for_payment, and Day_Since_CC_connect contained invalid characters (#, @, +, $, *). These were first converted to NaN and then imputed using the KNN imputer for numerical variables (Tenure, Account_user_count, Day_Since_CC_connect, and coupon_used_for_payment).
- Additionally, Tenure and Day_Since_CC_connect had anomalies, including 0 values, which were logically incorrect (e.g.Tenure cannot be zero and Day_Since_CC_connect cannot indicate zero days since the last contact). These were also imputed using the KNN imputer.
- The variable Service_Score contained 0 values, which were considered invalid and were replaced with 5, as only 0.1% of the values were affected.
- Since the bad values and anomalies were addressed first, the missing values were then imputed.
- Variables(cashback, Complain_ly, CC_Agent_Score, City_Tier, CC_Contacted_LY, Service_Score) were imputed using the KNN imputer.
- For categorical variables, the variable account_segment had inconsistent levels such as "Regular +", which was standardized to "Regular Plus", and "Super +" to "Super Plus".
- The missing values in account_segment were handled by assigning "Regular Plus" to customers from Tier 1 cities and "HNI" to customers from Tier 2 and Tier 3 cities, based on visual analysis.
- The missing values in Login_device were imputed as "Unknown", while Marital_Status was imputed using the most frequent category (mode).
- The KNN imputer was chosen for numerical variables because the proportion of missing values was low and KNN imputation helps preserve relationships within the data while avoiding bias.

Outlier Detection:

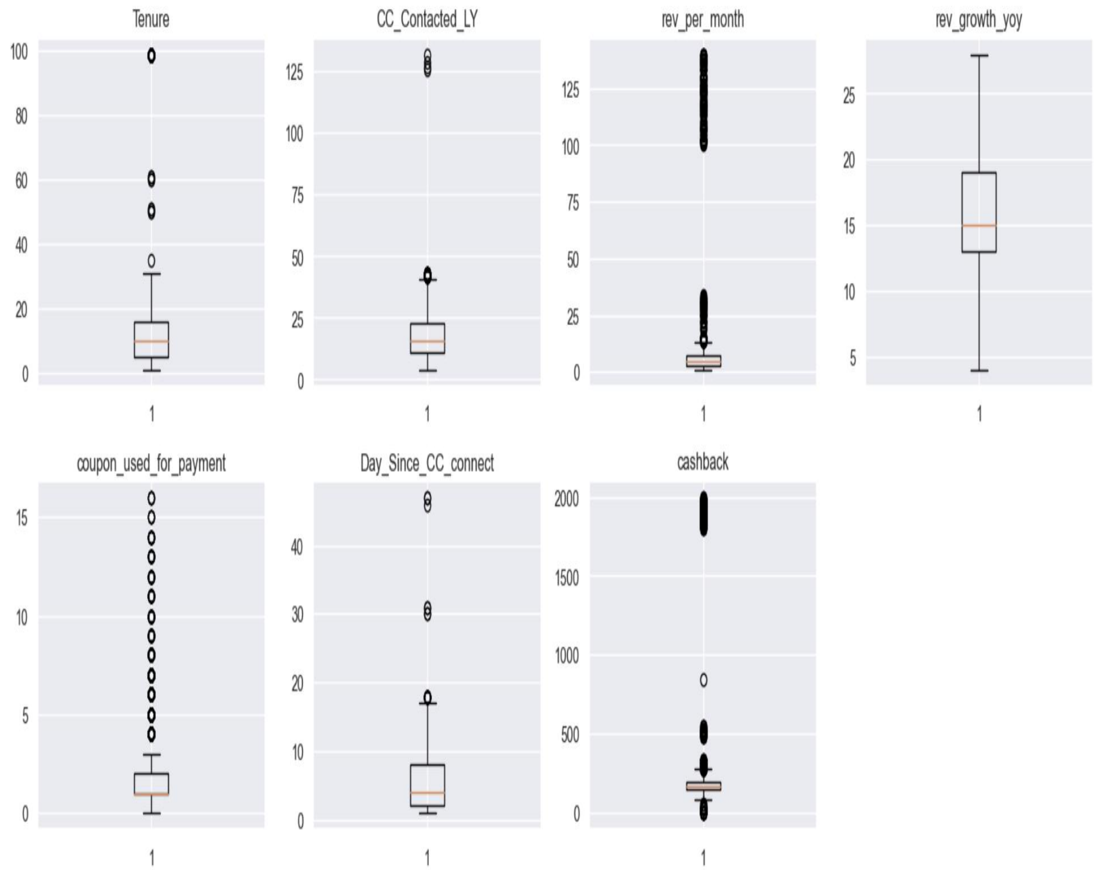

Fig 21: Outlier Detection

- There are outliers found in the data. However,the outliers are not treated as they are proper values.

3.2.Removal of irrelavant variables: Feature Selection

- The dataset includes a variable **AccountID**, which serves as a unique identifier for each customer. While this identifier is essential for tracking individual account, it does not contain predictive information relevant to the model. Since **AccountID** is a purely nominal variable with no inherent relationship to the target variable or other features, it does not contribute to the predictive power of the model. Therefore, it has been dropped from the dataset.
- The variable **Gender** is not significant for the model, as the proportion of categories is nearly equal, which was observed in the dataset. Additionally, a **Chi-square test** was performed, yielding a p-value of 0.0027, which is less than 0.05. This suggests that Gender has a statistically significant association with the target variable. However, due to its low predictive power, it was dropped from the dataset.

4. MODEL BUILDING

MODEL EVALUATION METRIC SELECTION:

Choosing the right evaluation metric is crucial because it directly impacts how a model's performance is measured and optimized. Selecting the wrong metric can lead to **misleading conclusions** and poor business decisions.

Model can make wrong predictions as:

False Positive : Predicting the customer has churned,but in reality the customer has not churned.

False Negative : Predicting the customer has not churned ,but in reality the customer has churned.

Which case is more important?

- **False Positives :**The company may offer unnecessary retention incentives (e.g., discounts, special offers) to a customer who was never going to leave.This leads to extra costs and reduced profit margins.Can result in over-allocation of customer retention efforts, diverting resources from truly at-risk customers.Increased costs but no actual loss of a customer.
- **False Negatives :** The company fails to take action to retain a customer who is actually at risk.Leads to lost revenue from customers who leave.Missed opportunity to retain high-value customers through personalized retention strategies.Direct revenue loss and potential long-term damage (e.g., loss of loyal customers).

Which is more important?

- False Negatives are usually more costly because losing a customer means lost revenue and future business.
- In industries with high customer acquisition costs, retaining an existing customer is cheaper than acquiring a new one—making recall (minimizing false negatives) a key priority.

How to reduce the loss?

- To minimize the losses,we need to reduce the False Positive,from the company prespective both the loss can impact .This can be achieved by maximizing the recall score.

Recall = True Positive(TP) / True Positive (TP) + False Negative(FN)

Hence,building a model that has good F1 Score and good Recall score will be the appropriate one.

4.1.MODEL PERFORMANCE WITH REASON FOR SELECTING THE MODEL

LOGISTIC REGRESSION:

Reason for choosing logistic regression:

Decision Making: This model provides probability of customer churn,making it useful for decisoin making

Interpretability: The model's coefficients indicate how each feature impacts the likelihood of churn.

Baseline Model: Serves as a strong baseline before using more complex models and sets a benchmark for other models.

- We will now perform logistic regression using statsmodels, a Python module that provides functions for the estimation of many statistical models, as well as for conducting statistical tests, and statistical data exploration.
- Using statsmodels, we will be able to check the statistical validity of our model - identify the significant predictors from p-values that we get for each predictor variable.
- A constant term is added to the independent variable matrix to account for the intercept in the linear regression model.

Train-Test Split:

- We want to predict churn status for the customers .Hence,we have created a new dataframe stored in the variable X that contains all variables expect Churn and another series Y that contains only the Churn. Before we proceed to build a model, we have encoded the categorical features.
- We have split the data into training and test sets using a 80:20 ratio. This allows us to build the model on the training data and then evaluate its performance on the test data, ensuring that the model's predictive ability is assessed on unseen data.
- We also ensure that the same combination of data points is used for every model we build, which helps in making a fair comparison between models and finding the best one.

Summary of the model after built:

Logit Regression Results

Dep. Variable:	Churn	No. Observations:	9008
Model:	Logit	Df Residuals:	8983
Method:	MLE	Df Model:	24
Date:	Sun, 16 Mar 2025	Pseudo R-squ.:	0.2402
Time:	08:43:39	Log-Likelihood:	-3111.4
converged:	True	LL-Null:	-4094.9
Covariance Type:	nonrobust	LLR p-value:	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
const	-4.1202	0.313	-13.168	0.000	-4.734	-3.507
Tenure	-0.0330	0.004	-7.808	0.000	-0.041	-0.025
City_Tier	0.4387	0.043	10.230	0.000	0.355	0.523
CC_Contacted_LY	0.0307	0.004	8.482	0.000	0.024	0.038
Service_Score	-0.2072	0.050	-4.177	0.000	-0.304	-0.110
Account_user_count	0.3412	0.036	9.375	0.000	0.270	0.413
CC_Agent_Score	0.2814	0.024	11.567	0.000	0.234	0.329
rev_per_month	0.0090	0.002	3.711	0.000	0.004	0.014
Complain_ly	1.5831	0.067	23.522	0.000	1.451	1.715
rev_growth_yoy	-0.0223	0.009	-2.482	0.013	-0.040	-0.005
coupon_used_for_payment	0.0554	0.019	2.938	0.003	0.018	0.092
Day_Since_CC_connect	-0.0203	0.012	-1.742	0.082	-0.043	0.003
cashback	0.0001	0.000	0.681	0.496	-0.000	0.001
Payment_Credit Card	-0.6487	0.113	-5.732	0.000	-0.871	-0.427
Payment_Debit Card	-0.5278	0.108	-4.890	0.000	-0.739	-0.316
Payment_E wallet	-0.0856	0.142	-0.604	0.546	-0.363	0.192
Payment_UPI	-0.4660	0.152	-3.066	0.002	-0.764	-0.168
account_segment_Regular	-0.1970	0.212	-0.929	0.353	-0.612	0.218
account_segment_Regular Plus	1.0309	0.107	9.594	0.000	0.820	1.241
account_segment_Super	-0.6049	0.108	-5.584	0.000	-0.817	-0.393
account_segment_Super Plus	-1.1305	0.215	-5.248	0.000	-1.553	-0.708
Marital_Status_Married	-0.3228	0.098	-3.283	0.001	-0.516	-0.130
Marital_Status_Single	0.7731	0.099	7.827	0.000	0.580	0.967
Login_device_Mobile	-0.3956	0.072	-5.487	0.000	-0.537	-0.254
Login_device_Unknown	-0.3755	0.137	-2.745	0.006	-0.644	-0.107

Table 5:Logistic Regression Summary

Intrepertaion:

- Negative values of the coefficient show that the probability of a customer churning decreases with the increase of the corresponding attribute value.
- Positive values of the coefficient show that the probability of a customer churning increases with the increase of the corresponding attribute value.
- p-value of a variable indicates if the variable is significant or not. If we consider the significance level to be 0.05 (5%), then any variable with a p-value less than 0.05 would be considered significant.
- Now, we will evaluate the model's performance by examining False Positives, False Negatives and various metrics.

Coefficient Interpretations

- Coefficient of some variables are positive an increase in these will lead to increase in chances of a customer churning.
- Coefficient of features and some levels are negative increase in these will lead to decrease in chances of a customer churning.

Converting coefficients to odds

- The coefficients (βs) of the logistic regression model are in terms of log(odds) and to find the odds, we have to take the exponential of the coefficients. Therefore, oods=exp (βs)
- The percentage change in odds is given as (exp(β)−1)*100

Coefficient interpretations after determining the exponential of the coefficients:

City_Tier: Holding all other features constant a 1 unit change in number of tenure will increase the odds of customer churning by ~1.55 or a ~55.03% increase in customer churning.

Tenure : Holding all other features constant a 1 unit change in tenure will decrease the odds of customer churning by ~0.97 or a ~3.24% decrease in customer churning.

The same applies to all the positive features or level as interpreted for **City_Tier** and to all neagtive features or level as interpreted for **Tenure.**

Training performance:

Accuracy	Recall	Precision	F1
0.860	**0.316**	**0.691**	**0.434**

Table 6:Logistic Regression training performance

Observations:

- We observe that the model's F1 score is approximately 0.43, and the Recall is around 0.31. This indicates that the model has a low Recall score, meaning that out of all the actual churned customers, only 31% were correctly predicted as churned.
- We can observe that this model is not performing well using the above performance metrics . The F1 score and recall score for training data is low. So, we will try building other models on the same data.

KNN:

Reason for choosing KNN:

Non-parametric & No Assumptions: Doesn't assume a distribution for the data, so it can adapt to weird shapes in minority class regions.

- We will now perform KNN using sklearn,we have created a new dataframe stored in the variable X1 that contains all variables expect Churn and another series Y1 that contains only the Churn.
- KNN doesn't assume an underlying data distribution, making it flexible for churn prediction.It classifies a new customer based on similar past customers, which is useful when churn behavior depends on patterns in past data.
- We scale the data using Standardization,which could make the KNN model easier to calculate the distance and produce better results. Same train and test split,which helps in comparing which model performs better and selection of model.
- In order to optimize our model, it's essential to experiment with different values of k to find the most suitable fit for our data. We can commence this process by setting k equal to 3 and gradually exploring other values to assess their impact on the model's performance.
- We'll only consider odd values of K as the classification will be done based on majority voting. KNN is performed using the model **knn_3** with K=3 and its performance is as follows.

Training performance:

Accuracy	**Recall**	**Precision**	**F1**
0.978	**0.915**	**0.955**	**0.935**

Table 7:Knn Training Performance

Test performance:

Accuracy	**Recall**	**Precision**	**F1**
0.926	**0.707**	**0.822**	**0.760**

Table 8:Knn Test Performance

Observations:

- We observe that the model is overfitting with low recall test score.The model has a low Recall score, meaning that out of all the actual churned customers, only 70% were correctly predicted as churned.
- We can observe that this model is not performing well using the above performance metrics . The F1 score and recall score are overfitting. So, we will try building other models on the same data.

NAIVE BAYES:

Reason for choosing Naive Bayes:

Better for Imbalanced Classes:CNB modifies the way it computes weights to downplay dominant (majority) classes.

- We will now perform Naive Baye's using sklearn,we have used the same dataframe stored in the variable **X2** that contains all variables expect **Churn** and same series **Y2** that contains only the **Churn.**
- Same train and test split,which helps in comparing which model performs better and selection of model.
- **Complement Naive Baye's** is performed because it estimates probabilities using the complement of that class, making it more robust to imbalanced data

Its performance is as follows.

Training performance:

Accuracy	**Recall**	**Precision**	**F1**
0.616	**0.721**	**0.266**	**0.388**

Table 9:Naive Bayes Training Performance

Observations:

- We can observe that this model is not performing well using the above performance metrics . All the metric score for training data is low. So, we will try building other models on the same data.
- The model was correctly able to predict 72.1% of positive class,but had poor precision score,where out of predicted positive only 26.6% were actually positive.

DECISION TREE:

Reason for choosing Decision Tree:

Interpretability: We can visualize the tree, and each decision/path clearly shows how the model arrived at a prediction.Great for non-technical stakeholders to understand decisions.

Complex interactions and non-linearity: Each feature may not directly impact the target variable,but interaction with other variable influences the prediction.

Imbalanced dataset: In case of imbalanced dataset the tree based model perform better ,because it has parameter through which imbalance can be managed.

- We will now perform decision tree which is a strong choice for churn prediction due to their interpretability, ability to handle mixed data types, and robustness to feature interactions.
- Same train and test split,which helps in comparing which model performs better and selection of model.

Let us see the performance of the decision tree.

Training performance:

Accuracy	Recall	Precision	F1
1.000	1.000	1.000	1.000

Table 10:Decision Tree Training Performance

Test performance:

Accuracy	Recall	Precision	F1
0.932	0.796	0.791	0.794

Table 11:Decision Tree Test Performance

Observation:

- We can observe that this model is not performing well using the above performance metrics . There is a clear indication that the model is performing good at training and performance goes down at test.This shows the model performance is overfitting.So, we will try building other models on the same data.

RANDOM FOREST:

Reason for choosing Random Forest:

Complex interactions and non-linearity: Each feature may not directly impact the target variable,but interaction with other variable influences the prediction.

Reduces overfitting and generalize better: Random Forest reduces overfitting and generalizes better than a single decision tree because it uses bootstrap aggregation (bagging) and random feature selection, which adds diversity among trees and improves robustness.

Imbalanced dataset: In case of imbalanced dataset the tree based model perform better ,because it has parameter through which imbalance can be managed.

- Random Forest is a powerful model for customer churn prediction due to its ability to handle high-dimensional data, non-linearity, feature interactions, and imbalanced datasets.
- Same train and test split,which helps in comparing which model performs better and selection of model.

Let us see the performance of the random tree.

Training performance:

Accuracy	Recall	Precision	F1
1.000	**1.000**	**1.000**	**1.000**

Table 12:Random Forest Training Performance

Test performance:

Accuracy	Recall	Precision	F1
0.953	**0.734**	**0.978**	**0.839**

Table 13:Random Forest Test Performance

Observation:

- We can observe that this model is not performing well using the above performance metrics . There is a clear indication when comparing the train and test perfromance the recall and F1 metric are overfitting.
- Maximizing Recall helps reduce Type II errors (False Negatives), while maximizing Precision helps reduce Type I errors (False Positives). A low F1 and Recall score on the test set indicate that the model struggles to correctly classify positive instances. So, we will try building other models on the same data.

BAGGING CLASSIFIER:

Reason for choosing Bagging Classifier:

Reduces variance: Bagging averages multiple models, less sensitive to data imbalance noise.

- Bagging (Bootstrap Aggregating) is an ensemble learning technique that reduces variance, mitigates overfitting, and improves model stability by combining multiple base models.
- Same train and test split,which helps in comparing which model performs better and selection of model.

Let us see the performance of the Bagging classifier.

Training performance:

Accuracy	Recall	Precision	F1
0.996	0.976	0.999	0.987

Table 14:Bagging classifier Training Performance

Test performance:

Accuracy	Recall	Precision	F1
0.947	0.710	0.957	0.815

Table 15:Bagging classifier Test Performance

Observation:

- We can observe that this model is not performing well using the above performance metrics . There is a clear indication when comparing the train and test perfromance the recall and F1 metric are overfitting.
- The model has performed better in training data with recall score of 0.976,but the test data did not generalize ,which performed with the recall score of 0.710,the model were able to predict only 71% of positive case correctly.
- Which indicates that this model is not better for churn prediction,because Recall is the important metric for the model,better recall score indicates the positive class (customer churned) is correctly predicted as churned.So, we will try building other models on the same data.

ADA BOOST CLASSIFIER:

Reason for choosing Ada Boost Classifier:

Adaptive learning: Focuses on mistakes made previously
Boosts weak learners: Improves even poor base models

- AdaBoost (Adaptive Boosting) is an ensemble learning technique that combines multiple weak learners to create a strong classifier. It works by giving more weight to misclassified instances in each iteration, forcing the model to focus on harder-to-classify cases.
- Same train and test split,which helps in comparing which model performs better and selection of model.

Let us see the performance of the Ada boost classifier.

Training performance:

Accuracy	Recall	Precision	F1
0.877	**0.423**	**0.739**	**0.538**

Table 16:Ada Boost Training Performance

Test performance:

Accuracy	Recall	Precision	F1
0.885	**0.425**	**0.775**	**0.549**

Table 17:Ada Boost Test Performance

Observation:

- We can observe that this model is not performing well using the above performance metrics . There is a clear indication that the model score in train and test are low for recall and Precision.
- Which indicates the model has not predicted the customer churn correctly ,when the customer has actually churned and the predicted as not churned ,since this score is less ,the performance of the model is not good.WIth the recall score of 0.425 which indicates only 42.5% positive case were predicted correctly which is low.So, we will try building other models on the same data.

GRADIENT BOOST CLASSIFIER:

Reason for choosing Ada Boost Classifier:

Reduces Bais and gives predcitive Accuracy : Gradient Boosting (GB Boost) is a powerful ensemble learning technique that builds models sequentially, correcting the errors of previous models using gradient descent.Since GB Boost learns from mistakes iteratively, it is effective in handling class imbalance.

- Same train and test split,which helps in comparing which model performs better and selection of model.Below are the performance

Training performance:

Accuracy	Recall	Precision	F1
0.903	0.520	0.848	0.644

Table 18:Gradient Boost Classifier Training Performance

Test performance:

Accuracy	Recall	Precision	F1
0.896	0.489	0.805	0.609

Table 19:Gradient Boost Classifier Test Performance

Observation:

- We can observe that this model is not performing well using the above performance metrics . There is a clear indication that the model score in train and test are low for recall and Precision and F1 score.So, we will try building other models on the same data.
- With recall score of 0.489 for the test data is low,this shows the model can predict only 48.9% of positive class correctly from test data.

EXTREME GRADIENT BOOST CLASSIFIER:

- XGB Includes L1 (Lasso) and L2 (Ridge) regularization, reducing overfitting compared to standard Gradient Boosting.Prunes the trees using maximum depth and minimum child weight to prevent overly complex models.
- Same train and test split,which helps in comparing which model performs better and selection of model.

Let us see the performance of the extreme gradient boost classifier.

Training performance:

Accuracy	Recall	Precision	F1
0.999	0.993	0.999	0.996

Table 20:Extreme Gradient Boost Classifier Training Performance

Test performance:

Accuracy	Recall	Precision	F1
0.957	0.785	0.948	0.859

Table 21:Extreme Gradient Boost Classifier Test Performance

Observation:

- We can observe that this model is performing well ,but the recall scores good for training data but low at test data,which indicates overfitting,and does not predict the positive class correctly,the model is able to predict 78% of positive class correclty which low comapred to other metric.
- Predicting the postive class correctly is important for business,which helps the business in retaining the at risk customer,who are about to churn.Hence, this model can be tuned with hyperparameters which may improve the recall score which is important for the business.

Since, we have build few models,now lets us try to tune the model and check the performance.

4.2.MODEL PERFORMANCE IMPROVEMENT:

LOGISITIC REGRESSION :

Dealing with multi-collinearity

We detect multicollinearity using the Variation Inflation Factor (VIF).The purpose of this anlaysis dictates which thershold to use.

const	84.852
Tenure	1.097
City_Tier	1.451
CC_Contacted_LY	1.020
Service_Score	1.165
Account_user_count	1.136
CC_Agent_Score	1.013
rev_per_month	1.007
Complain_ly	1.005
rev_growth_yoy	1.026
coupon_used_for_payment	1.201
Day_Since_CC_connect	1.240
cashback	1.080
Payment_Credit Card	3.042
Payment_Debit Card	3.257
Payment_E wallet	2.299

Payment_UPI	1.678
account_segment_Regular	1.311
account_segment_Regular Plus	2.577
account_segment_Super	2.273
account_segment_Super Plus	1.437
Marital_Status_Married	2.146
Marital_Status_Single	2.160
Login_device_Mobile	1.186
Login_device_Unknown	1.176

Table 22:VIF score

VIF score indicates that there is no multicollinearity between the variables.Hence,no features are dropped.

Model performance measurement ROC-AUC

Let's see if the f1_score and recall can be improved further by changing the model threshold,by default thershold is 0.5,let us try changing it.

First, we will check the ROC curve, compute the area under the ROC curve (ROC-AUC), and then use it to find the optimal threshold with help of **Youden's J statistic.**

Next, we will check the Precision-Recall curve to find the right balance between precision and recall to get the thershold where the f1 score will be maximum.

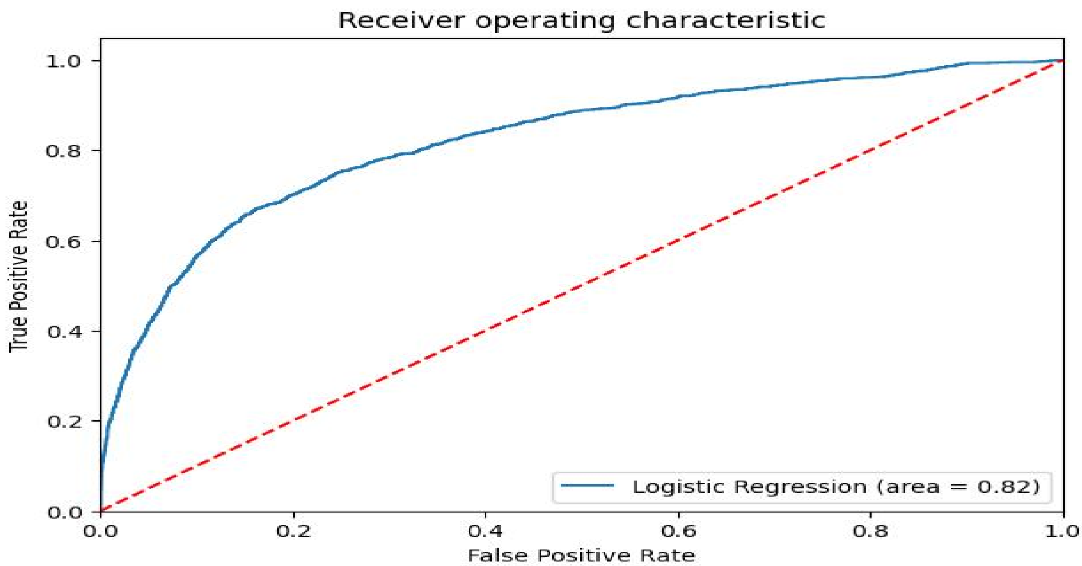

Fig 22: ROC-AUC Curve

Optimal threshold using AUC-ROC curve:

The optimal cut off would be where tpr is high and fpr is low.TPR is nothing but the true positive rate and FPR is false positive rate.

TPR=TP/TP+FN and **FPR=FP/FP+TN**

The optimal threshold determined was 0.2373,now lets check the performance at the optimal thershold value.

Training performance:

Accuracy	**Recall**	**Precision**	**F1**
0.810	**0.668**	**0.458**	**0.544**

Table 23:Logisitic Regression optimal thershold Training Performance

Observation:

- With thershold at 0.23 the performance of the model is poor at the training.Recall score of 66.8% and F1 score at 54.4% indicates low performance at predicting churned customers and predicted churned customers has not actually churned which is bad indication for the model.

Lets us see whether we can improve our scores further by plotting precision and recall curve by determing new thershold where precision and recall intercepts.

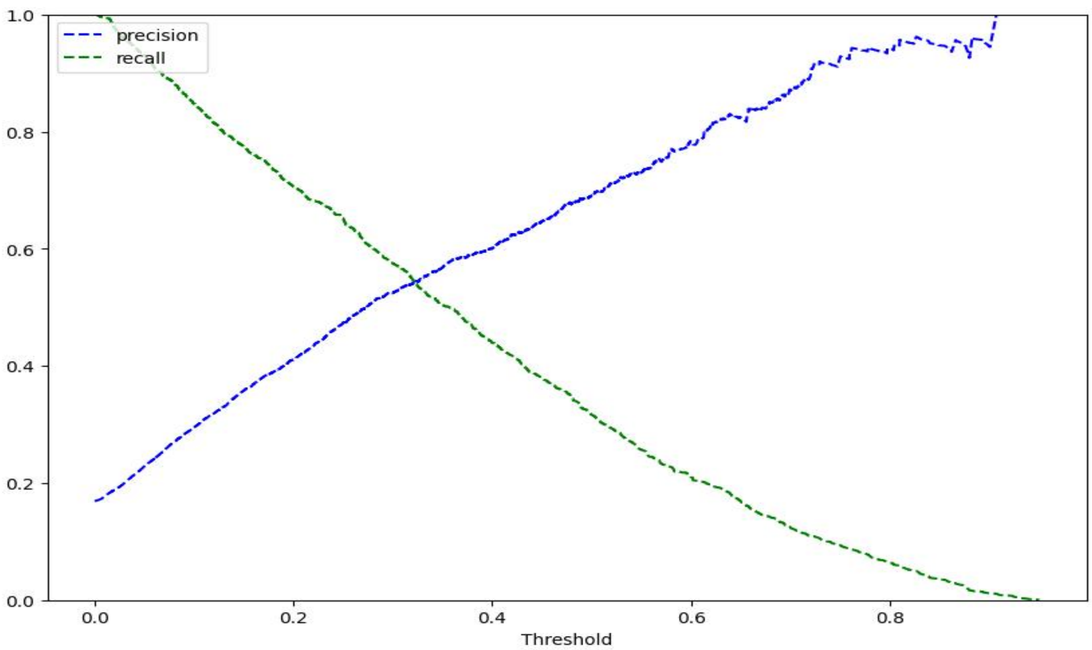

Fig 23: Precision-Recall Curve

Thershold at 0.3 where precision and recall intercepts,lets see how the scores stands at 0.3.

Training performance:

Accuracy	Recall	Precision	F1
0.840	0.575	0.525	0.549

Table 24:Logisitic Regression optimal thershold Training Performance

Observation:

- There at no improvement of recall and f1 score at training,which clearly indciates the model performance will be low at test as well.Hence the model performance at thershold 0.3,where recall score of 57.5% and F1 of 54.9% indicates a poor performance.

KNN :

Let's run the KNN with no of neighbours to be at 1,3,5..19 and find the optimal number of neighbours from the above list using the recall score.

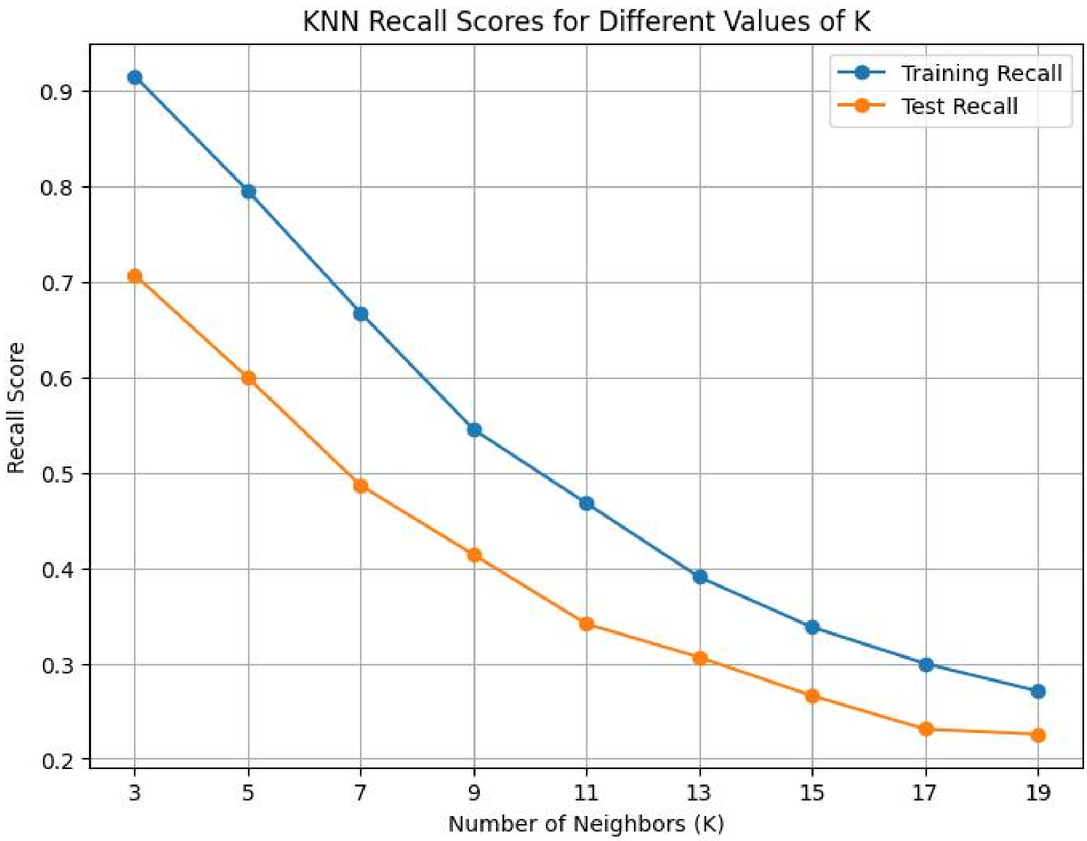

Fig 24: KNN Recall Scores for different values of K

Let's run the KNN with no of neighbours to be 1,3,5..19 and find the optimal number of neighbours from the above list using the F1 score.

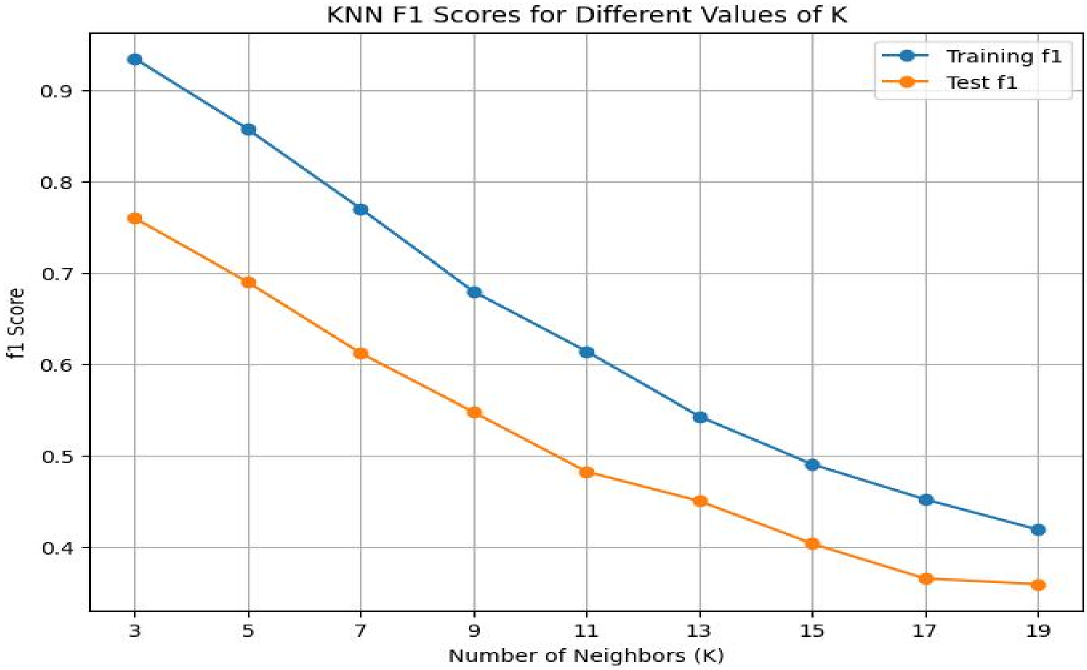

Fig 25: KNN F1 Scores for different values of K

- The recall scores and F1 scores for both training and test sets are highest when k=3. This suggests that with k=3, the model is better at identifying positive and negative instances in both the training and test data compared to other values of k.
- As the value of k increases beyond 3, the recall scores and F1 scores tend to decrease for both training and test sets. This indicates a potential risk of the model not being able to identify the underlying patterns in the data.
- Therefore, based on the performance indicater at various K, k=3 appears to be the most suitable choice for balancing model performance between capturing positive
- instances effectively and generalizing well to new data.With K=3,which has been done at the model building stage and observations were made.

DECISION TREEE :

WITH CLASS_WEIGHTS:

- Since,the dataset is imbalanced, the model tends to favor the majority class.
- class_weight='balanced' adjusts the weights inversely proportional to class frequencies, ensuring the model gives equal importance to both classes.

- Helps improve recall for the minority class (detecting churned customers).
- Tuning the decision tree with parameter class_weight='balanced' ,with that lets us see the performance.

Training performance:

Accuracy	**Recall**	**Precision**	**F1**
1.000	**1.000**	**1.000**	**1.000**

Table 25:Decision Tree Classifier with class weight Training Performance

Test performance:

Accuracy	**Recall**	**Precision**	**F1**
0.931	**0.755**	**0.814**	**0.784**

Table 26:Decision Tree Classifier with class weight Test Performance

Observation:

- The performance of decisoin tree reamins the same even after weights were adjusted by giving equal importance to both the class.The performance shows recall with 75.5% and F1 of 78.4% shows decent prediction of customer churning correctly.

PRE-PRUNING:

- Pre-pruning (also called early stopping) prevents a Decision Tree from growing too deep, reducing the risk of overfitting while improving generalization.
- Pre-pruning uses parameters which helps the model in preventing the tree growing deep.Below are the parameter which helps in addressing overfitting by preventing the tree growing deep.

DecisionTreeClassifier(class_weight='balanced',max_depth=6,max_leaf_nodes=50,min_samples_split=10,random_state=1)

- We have used Grid search tuning technique that attempts to compute the optimum values of hyperparameters.Through grid search CV,which by default perfroms K-fold cross validation technique we have attained the optimum values for each parameter which is shown above.With,that lets us check the performance of the model.

Training performance:

Accuracy	**Recall**	**Precision**	**F1**
0.789	**0.795**	**0.445**	**0.571**

Table 27:Decision Tree Classifier Pre-Pruning Training Performance

Observation:

- The model with pre-purning had performed poorly at training ,with 79.5% as recall,Precision of 44.5% and F1 of 57.1% are low ,which is not good for our case.Eventhough model were able to predit 79.5% of the positive class correctly from training data,the model has not performed better,where the model predicted as positive and only 44.5% were actually positive.

POST-PRUNING:

- Post-pruning (also called cost-complexity pruning) trains the full tree first and then prunes unnecessary branches, improving generalization and reducing overfitting.
- The DecisionTreeClassifier provides parameters such as min_samples_leaf and max_depth to prevent a tree from overfitting. Cost-complexity pruning provides another option to control the size of a tree. In DecisionTreeClassifier, this pruning technique is parameterized by the cost-complexity parameter, ccp_alpha. Greater values of ccp_alphaincrease the number of nodes pruned.
- Here, we only show the effect of ccp_alpha on regularizing the trees and how to choose a ccp_alpha based on validation scores.With ccp_alpha = 0.00011 attained through grid search,we will check the performance of model.

Training performance:

Accuracy	**Recall**	**Precision**	**F1**
0.993	**1.000**	**0.958**	**0.979**

Table 28:Decision Tree Classifier Post-Pruning Training Performance

Test performance:

Accuracy	**Recall**	**Precision**	**F1**
0.925	**0.777**	**0.771**	**0.774**

Table 29:Decision Tree Classifier Post-Pruning Test Performance

Observation:

- The model clearly indicates overfitting.The recall score of 77.7% is low as per the business requirement of predicting churn correctly. which needs to be addressed.

RANDOM FOREST:

- Tuning a RandomForestClassifier involves optimizing hyperparameters to enhance the model's performance on the given dataset.
- Key hyperparameters include.The number of trees in the forest,the maximum depth of each tree,the number of features considered for splitting at each node, and the minimum number of samples required to split a node.
- The goal is to improve the model's ability to learn from all classes effectively while avoiding overfitting. Careful tuning is necessary to ensure the model generalizes well and does not become overly complex.
- Cross-validation is used throughout the tuning process to evaluate the model's performance and ensure that the selected hyperparameters lead to a model that performs well on unseen, real-world data.
- After hyperparameter tuning, the performance of the RandomForestClassifier on both the training and test datasets is thoroughly analyzed to confirm that the model makes accurate predictions across all classes.

Training performance:

Accuracy	**Recall**	**Precision**	**F1**
0.998	**1.000**	**0.988**	**0.994**

Table 30:Random Forest Classifier Tuned Training Performance

Training performance:

Accuracy	**Recall**	**Precision**	**F1**
0.956	**0.793**	**0.928**	**0.855**

Table 31:Random Forest Classifier Tuned Test Performance

Observation:

- Model has performed well on training,but failed to perform on test with 79.3% of recall score which is overfitting,recall score is important in our case,the model has not equiped well on test for recall scores,where the model had predicted only 79% of churned customers correctly.

3.5.BAGGING CLASSIFIER:

- Tuning a BaggingClassifier involves optimizing hyperparameters to enhance the model's performance on the given dataset.

- Key hyperparameters include.Class_weights,the number of base estimators (weak learners),the maximum number of samples drawn for training each estimator,the maximum number of features considered for training each estimator, and Whether bootstrapping is used for sample selection.
- The goal is to improve the model's ability to learn effectively from the data while reducing variance and avoiding overfitting. Careful tuning ensures that individual estimators contribute meaningfully without excessive complexity.
- Cross-validation is used throughout the tuning process to evaluate the model’s performance and ensure that the selected hyperparameters lead to a model that generalizes well on unseen, real-world data.
- After hyperparameter tuning, the performance of the BaggingClassifier on both the training and test datasets is thoroughly analyzed to confirm that the model makes accurate and stable predictions.

Training performance:

Accuracy	**Recall**	**Precision**	**F1**
0.258	**0.999**	**0.186**	**0.313**

Table 32:Bagging Classifier Tuned Training Performance

Test performance:

Accuracy	**Recall**	**Precision**	**F1**
0.251	**0.997**	**0.180**	**0.306**

Table 33:Bagging Classifier Tuned Test Performance

Observation:

- The model demonstrates extremely high recall (close to 1), indicating that it correctly identifies almost all positive cases. However, the precision is very low (around 0.18–0.19), suggesting a high number of false positives. This imbalance results in a low F1-score (around 0.31), reflecting poor overall model performance.These results suggest that the model is heavily biased toward predicting positives.

GRADIENT BOOSTING CLASSIFIER:

- Tuning a GradientBoostingClassifier involves optimizing hyperparameters to enhance the model's performance on the given dataset.
- Key hyperparameters include.The number of boosting stages (n_estimators),the learning rate (learning_rate),The maximum depth of

each tree (max_depth), and the fraction of samples used for fitting individual base learners (subsample).

- The goal is to improve the model's ability to learn complex patterns while minimizing bias and variance. Careful tuning ensures that the boosting process effectively reduces errors without overfitting.
- Cross-validation is used throughout the tuning process to evaluate the model's performance and ensure that the selected hyperparameters lead to a model that generalizes well on unseen, real-world data.
- After hyperparameter tuning, the performance of the GradientBoostingClassifier on both the training and test datasets is thoroughly analyzed to confirm that the model makes accurate and stable predictions.

Training performance:

Accuracy	Recall	Precision	F1
1.000	1.000	1.000	1.000

Table 34:Gradient Boost Classifier Tuned Training Performance

Test performance:

Accuracy	Recall	Precision	F1
0.932	0.796	0.791	0.794

Table 35:Gradient Boost Classifier Tuned Test Performance

Observation:

- The training set results show high performance, with an accuracy, recall, precision, and F1-score of 1.000, indicating that the model has perfectly learned the training data. However, the test set results show a drop in performance, with an accuracy of 0.932, recall of 0.796, precision of 0.791, and an F1-score of 0.794.
- This suggests that while the model generalizes well, there is some overfitting, as seen in the difference between training and test performance. The recall and precision values on the test set indicate that the model makes some misclassifications, but overall, it maintains a good balance between identifying positive cases and minimizing false positives.

EXTREME GRADIENT BOOSTING CLASSIFIER:

- Tuning an XGBClassifier involves optimizing hyperparameters to enhance the model's performance on the given dataset.
- Key hyperparameters include.The number of boosting rounds (n_estimators),the learning rate (learning_rate),the maximum depth of each tree (max_depth),the fraction of samples used for training each tree (subsample),the fraction of features used for each split (colsample_bytree),the L1 regularization term (alpha), and the L2 regularization term (lambda).
- The goal is to improve the model's ability to learn complex patterns while minimizing bias and variance. Careful tuning ensures that boosting effectively reduces errors without overfitting. Regularization parameters (alpha and lambda) help control model complexity and prevent excessive fitting to noise.
- Randomized cross-validation, along with stratified cross-validation, is used for imbalanced datasets to efficiently search for the best hyperparameters while ensuring that the model maintains robust performance across different class distributions. Stratified cross-validation ensures that each fold maintains the same proportion of classes, leading to a more reliable evaluation. Additionally, cross-validation is applied throughout the tuning process to evaluate the model's performance and ensure that the selected hyperparameters lead to a model that generalizes well on unseen, real-world data.
- After hyperparameter tuning, the performance of the XGBClassifier on both the training and test datasets is thoroughly analyzed to confirm that the model makes accurate and stable predictions.

Training performance:

Accuracy	**Recall**	**Precision**	**F1**
1.000	**1.000**	**1.000**	**1.000**

Table 36:Extreme Gradient Boost Classifier Tuned Training Performance

Test performance:

Accuracy	**Recall**	**Precision**	**F1**
0.977	**0.914**	**0.947**	**0.930**

Table 37:Extreme Gradient Boost Classifier Tuned Test Performance

Observation:

- The model achieves good performance on the training set, with an accuracy, recall, precision, and F1 score of 1.00. On the test set, it maintains strong generalization with an accuracy of 0.977, recall of 0.914, precision of 0.947, and an F1 score of 0.930. These results confirm that this is the best-performing model, effectively balancing precision and recall while minimizing errors, making it well-suited for our problem.
- A recall of 0.914 on the test set means that 91.4% of customers who actually churned were correctly identified by the model. This is crucial in a churn prediction scenario, as missing a true churn case (false negative) could result in lost customers without proactive retention efforts. The high recall ensures that most at-risk customers are flagged, allowing the business to take necessary interventions to reduce churn.

5.MODEL COMPARISION AND FINAL MODEL SELECTION WITH MODEL VALIDATION

Model Comparision

- Based on the evaluation results of the improved models for customer churn prediction, it is evident that the models demonstrate enhancements in performance metrics compared to their default counterparts.
- We can analyze this with the help of the performance comparison data below.

Training Performance comparison:

Metric	**KNN**	**Decision Tree**	**Decision Tree Post Prunining**	**Random Forest**	**RF with class weight**	**Tuned XGB**
Accuracy	0.978	1.000	0.993	1.000	1.000	1.000
Recall	0.915	1.000	1.000	1.000	1.000	1.000
Precision	0.955	1.000	0.958	1.000	1.000	1.000
F1	0.935	1.000	0.979	1.000	1.000	1.000

Table 38:Training Performance Comparision

Test Performance comparison:

Metric	**KNN**	**Decision Tree**	**Decision Tree Post Prunining**	**Random Forest**	**RF with class weight**	**Tuned XGB**
Accuracy	0.926	0.932	0.925	0.953	0.956	**0.977**
Recall	0.707	0.796	0.777	0.734	0.742	**0.914**
Precision	0.822	0.791	0.771	0.978	0.993	**0.947**
F1	0.760	0.794	0.774	0.839	0.849	**0.930**

Table 39:Test Performance Comparision

Final Model Selection:

- After conducting hyperparameter optimization for all the models, it was found that the XGBoost (XGB) classifier showed good Recall score for customer churn prediction.Which was able to predict 91.4% of the positive cases correctly.
- XGBoost holds several advantages over other models, which are listed below
- **Handling Imbalanced Data:** XGBoost allows fine-tuning of class weights (scale_pos_weight), making it well-suited for datasets where churned customers are a minority class.
- **Regularization to Prevent Overfitting:** XGBoost incorporates L1 (alpha) and L2 (lambda) regularization, which helps control model complexity and reduces overfitting, leading to better generalization.
- **Boosting Mechanism for Higher Accuracy:** Unlike bagging-based models like Random Forest, XGBoost uses gradient boosting, where each new tree corrects the errors of the previous one, leading to superior predictive performance.
- **Computational Efficiency:** XGBoost is optimized for speed using parallel and distributed computing, making it significantly faster than traditional gradient boosting methods.
- **Feature Importance and Interpretability:** XGBoost provides built-in feature importance scores, helping businesses identify key drivers of customer churn and take proactive measures.
- Due to these advantages, the XGBoost Classifier is the most suitable model for predicting customer churn in this scenario.

Class wise model Performance of optimal model:

Training data:

	precision	**recall**	**f1-score**	**support**
0	1.000	1.000	1.000	7484
1	1.000	1.000	1.000	1524
accuracy			1.000	9008
macro avg	1.000	1.000	1.000	9008
weighted avg	1.000	1.000	1.000	9008

Table 40:Class wise model Performance of optimal model - Train data

Test data:

	precision	**recall**	**f1-score**	**support**
0	0.983	0.990	0.986	1880
1	0.947	0.914	0.930	372
accuracy			0.977	2252
macro avg	0.965	0.952	0.958	2252
weighted avg	0.977	0.977	0.977	2252

Table 41:Class wise model Performance of optimal model - Test data

- The model performs better in predicting the positive case.This helps business to understand the at risk customer who are about to churn in advance and perform strategies in retaining the customer.
- This XGB model was choosen as the best model for the business,as it meets the business requirement,which had predicted 91.4% of positive class correctly.Out of 372 test cases which belong to positive class,340 were correctly predicted as positive.

Performance Measurement of best model with model validation:

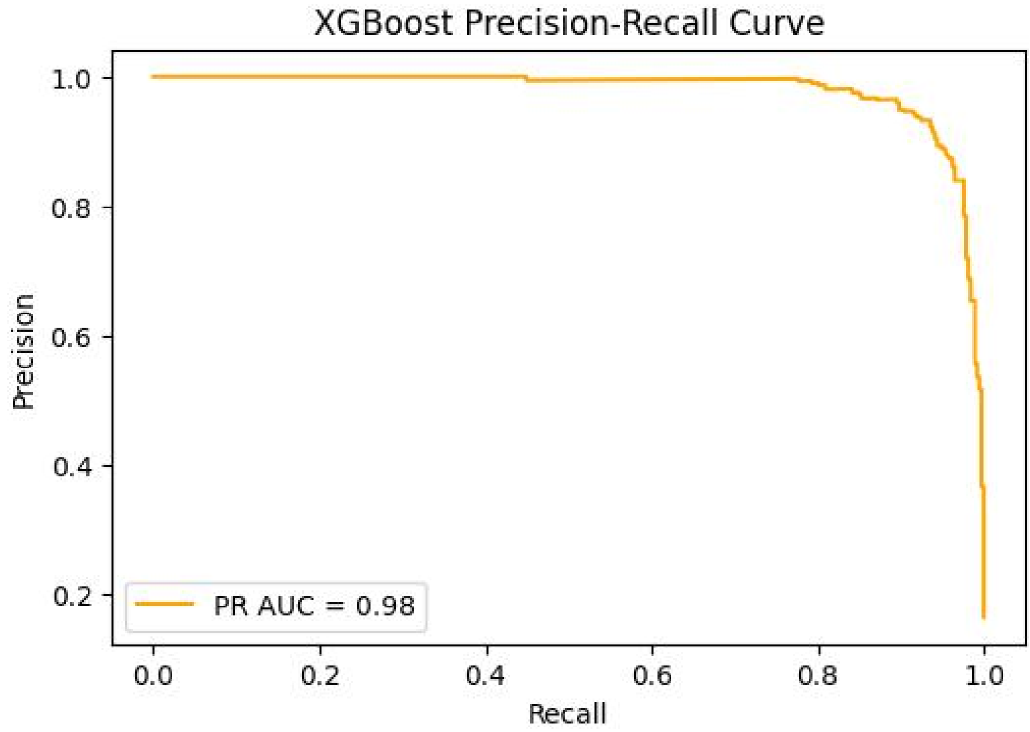

Fig 26:Precision-Recall Curve

With PR AUC = 0.98,

- This is an excellent PR AUC close to 1.0 indicates that the model maintains high precision and high recall, especially important in imbalanced classification problems.
- PR AUC is a better metric than ROC AUC when the positive class is less .When we are more concerned with false positives or false negatives.
- Flat Line near Precision = 1.0 for low Recall .The curve starts at precision ≈ 1, meaning the first few predictions (with highest confidence) are extremely accurate.These are likely very high-quality positive predictions.
- As the model tries to capture more positives (higher recall), the precision slowly drops, which is normal.This shows a good balance,the model retains high precision up to a recall of ~0.9.
- XGB model is selected as the best model on basis of recall score.As the business is more concerned about predicting the churn customer.Good recall score indicates the model is able to predict the churn customers correctly.

For the tuned XGBClassifier, the most important features utilized in identifying the target variable, i.e., Churn, are represented below:

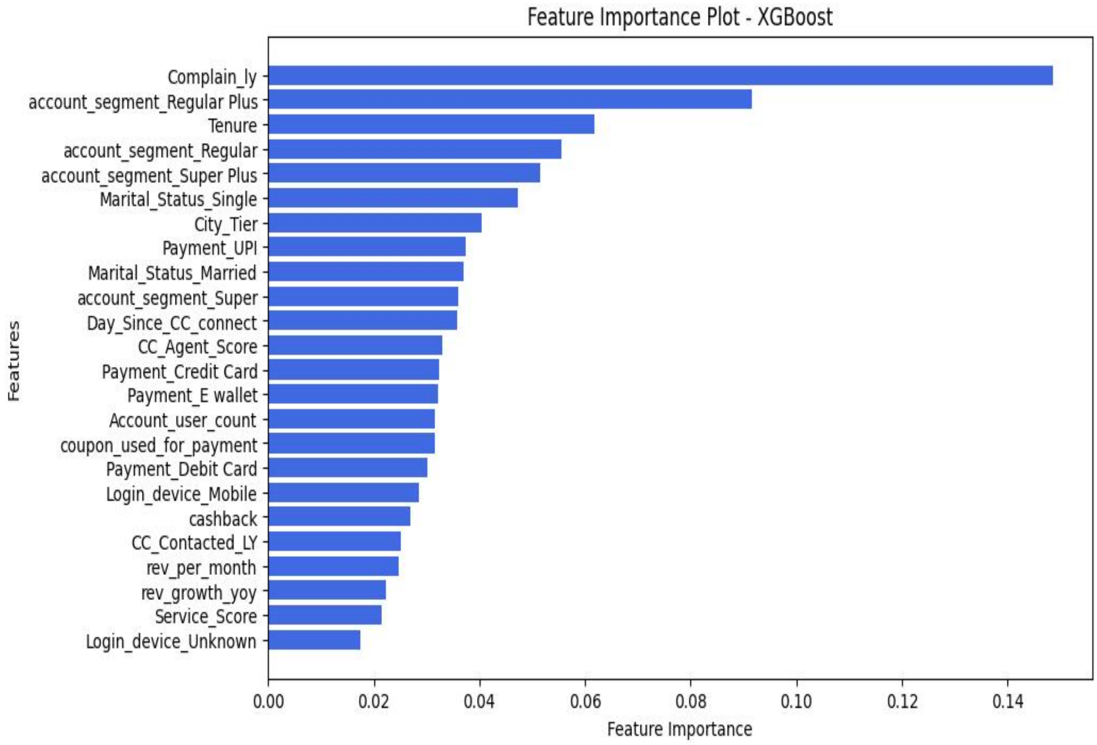

Fig 27: Feature Importance - Optimal model

The top 5 attributes affecting the model's predictions are:

1. Complain_ly 2.account_segment_Regular Plus 3.Tenure 4.account_segment_Regular 5.Marital_Status_Single

Although other attributes may influence the model's performance, the above-given attributes are more likely to be the primary deciding factors in determining whether a customer will churn or not.**This feature importance plot gives us the contribution of each feature in model building.**

6.FINAL INTERPRETATION AND RECOMMENDATION:

Actionable insights and recommendation are given as per the final model,by understaing the feature importance.

Actionable Insights:

- When the customer have raised any complain,the chances are higher that the customer will churn.
- Customers in the Regular and Regular Plus account segments show a higher likelihood of churn.Customers in these segments who have also raised complaints are at even higher risk of leaving.

- More tenure the likelihood of customer churning is less,less the tenure the customer churning is more.
- If the martial status is single ,chances of customer churning is higher.

Customer Complain: When the customer have raised any complain,the chances are higher that the customer will churn.Complaint frequency and severity can serve as early churn indicators.

Account segment Regular and Regular Plus: Customers in the Regular and Regular Plus account segments show a higher likelihood of churn.These segments might be more price-conscious and easily influenced by promotions or lower-cost alternatives from competitors.Customers in these segments who have also raised complaints are at even higher risk of leaving.

Tenure: More tenure the likelihood of customer churning is less,less the tenure the customer churning is more.Higher tenure often indicates a positive past experience or a strong relationship with the brand, leading to increased retention.Customers who have been with the company for a long time tend to develop loyalty and familiarity with the services.

Martial Status Single: If the martial status is single ,chances of customer churning is higher.They may be more flexible in exploring alternative options, especially if they find better pricing or service offerings elsewhere.Single customers might have different spending patterns compared to married individuals or those with dependents.

Recommendations:

- Implementing a priority resolution system for customers who have raised complaints.Providing real-time tracking of complaint resolution status to improve transparency.
- Compensation for customers with delayed resolution,like discounts for purchase above certain value.
- Retaining the regular and regular plus segments through flash sales,refer a friend discount.
- Retaining short-tenured customers with extented return period,offers on weeks before birthday,sign-up bonus like free shipping on first three purchase.
- Single customers are more likely to churn due to price sensitivity and product availability issues.Frequent stock unavailability or delayed shipping can drive them to competitors.

- Business can implement back-in-stock alerts, flexible payment options,faster shipping services and providing training to customer care employees.

1. Customers Who Have Raised Complaints

- Proactive Resolution: Addressing the complaints quickly and effectively, offering personalized solutions to resolve issues before they escalate.
- Follow-up Mechanisms: Implement follow-up calls or emails to ensure customer satisfaction after issue resolution.
- Loyalty Incentives: Offer discounts, service upgrades, or exclusive perks to customers who faced issues to rebuild trust.

2. Customers in the Regular and Regular Plus Account Segments

- Value-Added Services: Introduce bundled benefits or loyalty rewards to make these segments feel valued without significantly increasing costs.
- Personalized Promotions: Offer exclusive discounts or retention incentives tailored to their preferences.
- Proactive Engagement: Regularly communicate with these customers via targeted messaging to highlight benefits and prevent churn.

3. Customers with Short Tenure (New Customers)

- Onboarding Experience: Improve the first few months of interaction with guided onboarding and personalized engagement.
- Early Engagement Offers: Provide exclusive early-stage discounts or rewards to encourage retention.
- Proactive Check-ins: Assign customer service representatives to monitor and check in with new customers to address concerns early.

4. Single Customers

- Customized Plans: Offer flexible or commitment-free plans that cater to their lifestyle.
- Personalized Engagement: Market services that align with their interests, such as convenience-driven benefits or exclusive experiences.
- Social Engagement: Leverage community-building initiatives, such as networking events or referral bonuses, to increase emotional connection with the brand.