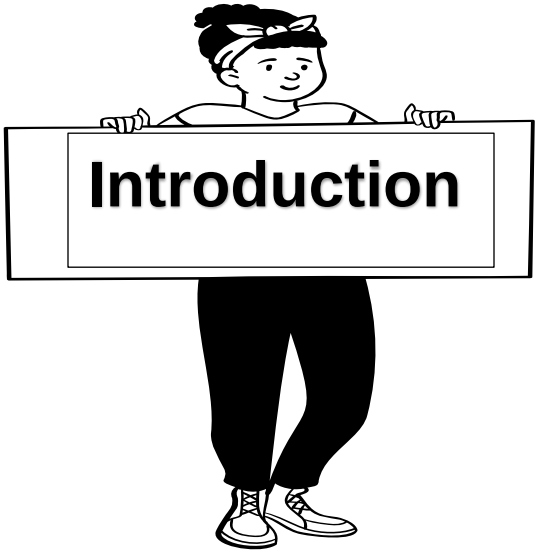
BUSINESS REPORT

Customer Churn

By: **Govind Singh Rawat**

Contents



Executive Summary

Customer churn is a significant issue affecting businesses across various industries. The primary objective of this project is to develop a predictive model to identify potential churners and create segmented offers to retain them. By leveraging historical customer data, we aim to enhance customer satisfaction, maximize Customer Lifetime Value (CLV), and maintain a competitive edge in the market.

Understanding the Challenge

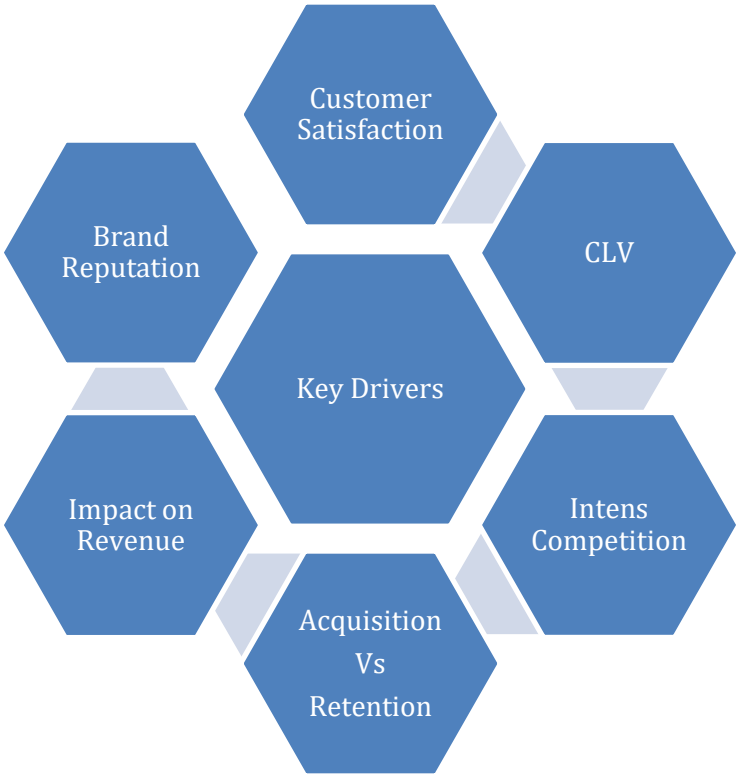
The primary business problem addressed in this project is customer retention. High churn rates can severely impact a company's revenue and brand reputation. The project focuses on developing a churn prediction model to proactively identify customers at risk of leaving and implement targeted retention strategies.

Objectives

- Develop a predictive model for customer churn.
- Provide segmented offers to potential churners.
- Enhance customer satisfaction and loyalty.
- Maximize Customer Lifetime Value (CLV).
- Maintain a competitive position in the market.

Need for the Project

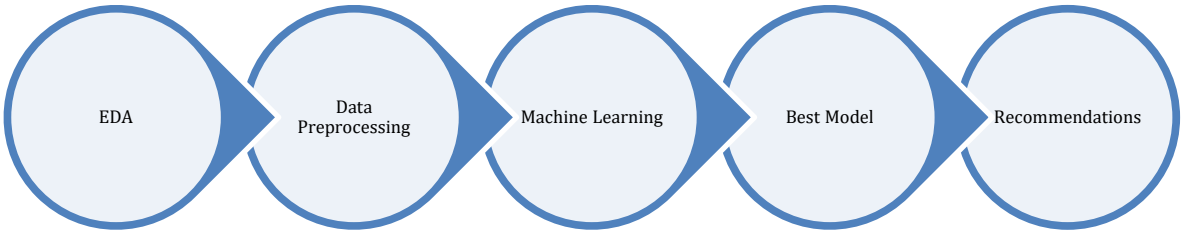
Key Drivers



1. Customer Satisfaction: Improving the overall customer experience to reduce churn.
2. Maximizing CLV: Retaining existing customers is more cost-effective than acquiring new ones.
3. Intense Competition: The competitive market necessitates proactive measures to retain customers.
4. Cost of Acquisition vs. Retention: The cost of acquiring new customers is higher compared to retaining existing ones.
5. Impact on Revenue and Brand Reputation: High churn rates can negatively affect both revenue and brand image.

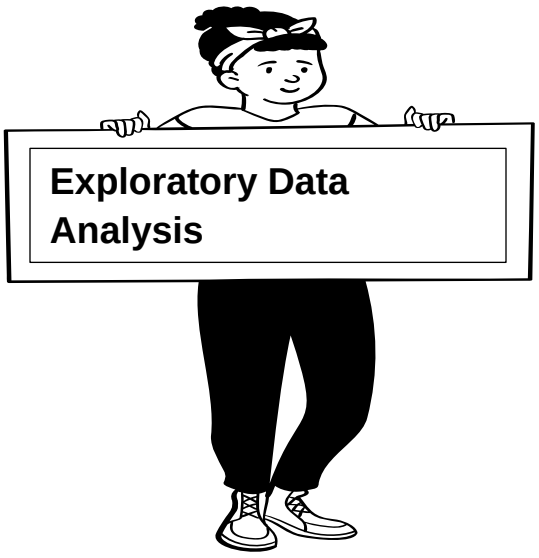
Project Scope

Approach



This project employs a supervised learning classification approach to predict customer churn. The process involves:

- Analyzing historical customer data.
- Data preprocessing and cleaning.
- Applying machine learning algorithms.
- Evaluating models and selecting the best one.
- Providing actionable recommendations based on the analysis.



The dataset contains historical data about the customers. The data appears to be collected over a period, as indicated by variables such as 'rev_growth_yoy' (year-over-

year revenue growth) and 'Complain_ly' (complaint lodged in the last year). The exact period covered by the dataset is not specified, but it likely spans at least one year based on the presence of yearly metrics.

While the specific details of data collection methodology are not provided, we can infer that the dataset represents a snapshot of customer data collected over a period, possibly through various channels and methods within the company's operations.

Visual Inspection:

The dataset contains 11260 observations and 19 columns including 1 unique identifier (AccountID).

```
Data columns (total 19 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   AccountID                11260 non-null  int64
 1   Churn                    11260 non-null  int64
 2   Tenure                   11158 non-null  object
 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       11148 non-null  object
 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            11158 non-null  object
 13  Complain_ly              10903 non-null  float64
 14  rev_growth_yoy           11260 non-null  object
 15  coupon_used_for_payment  11260 non-null  object
 16  Day_Since_CC_connect     10903 non-null  object
 17  cashback                 10789 non-null  object
 18  Login_device             11039 non-null  object
```

Impurities have been observed in the dataset. Lot of columns contain null values and some of them have special characters too.

When we check for the number of null values per columns

AccountID	0
Churn	0
Tenure	102
City_Tier	112
CC_Contacted_LY	102
Payment	109
Gender	108
Service_Score	98
Account_user_count	112
account_segment	97
CC_Agent_Score	116
Marital_Status	212
rev_per_month	102
Complain_ly	357
rev_growth_yoy	0
coupon_used_for_payment	0
Day_Since_CC_connect	357
cashback	471
Login_device	221

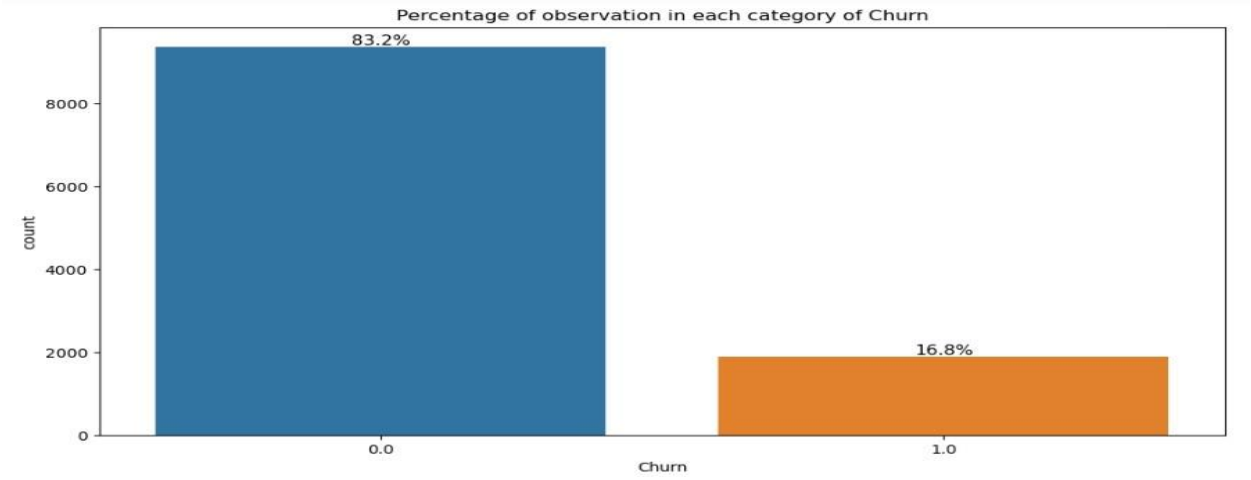
These missing values should be imputed. And we are going to impute them in the data cleaning phase.

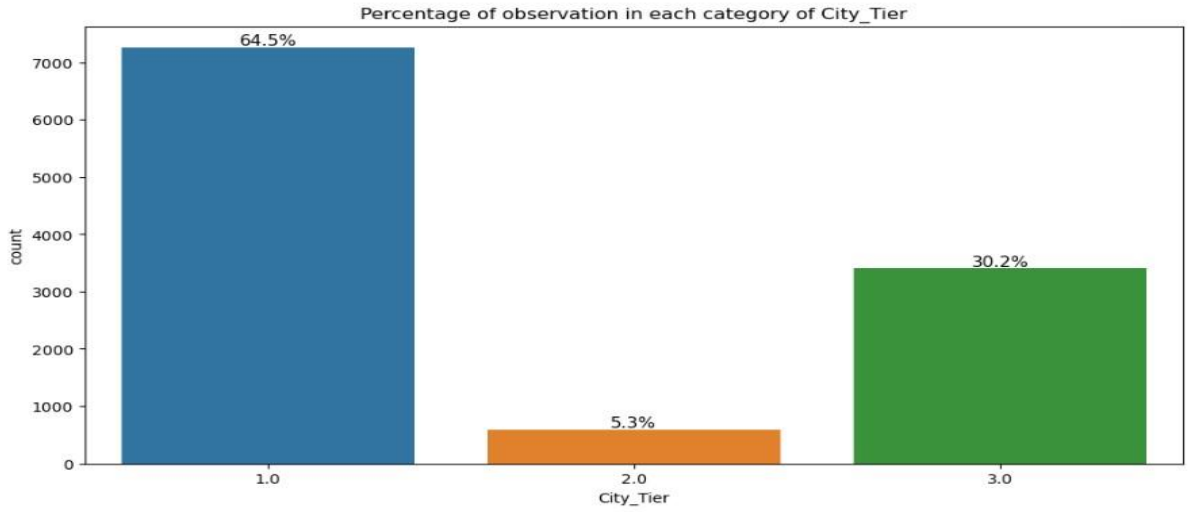
No duplicate observations are present in the dataset.

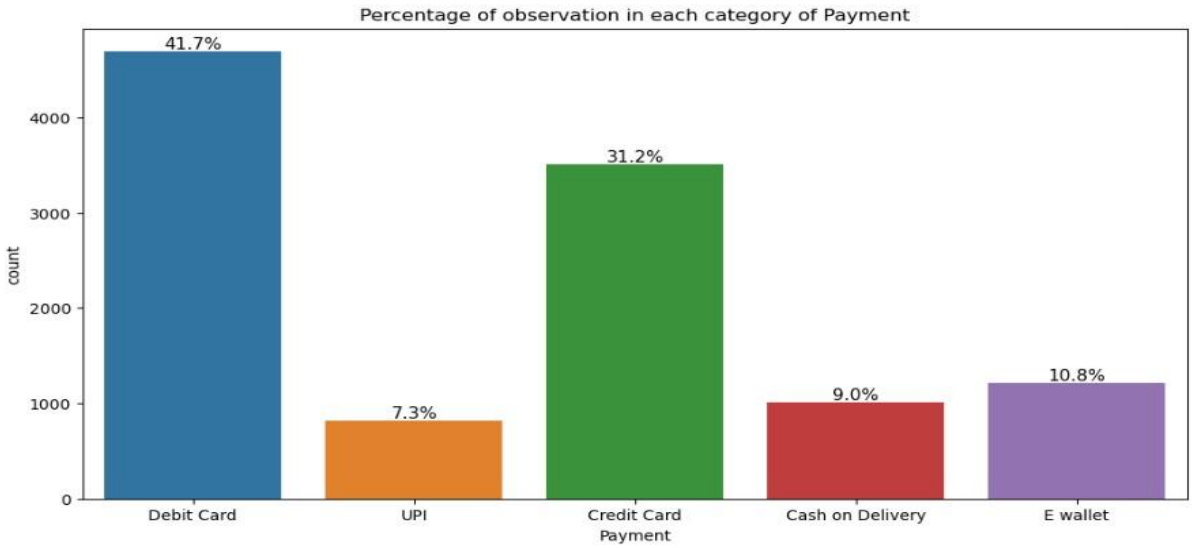
Univariate Analysis:

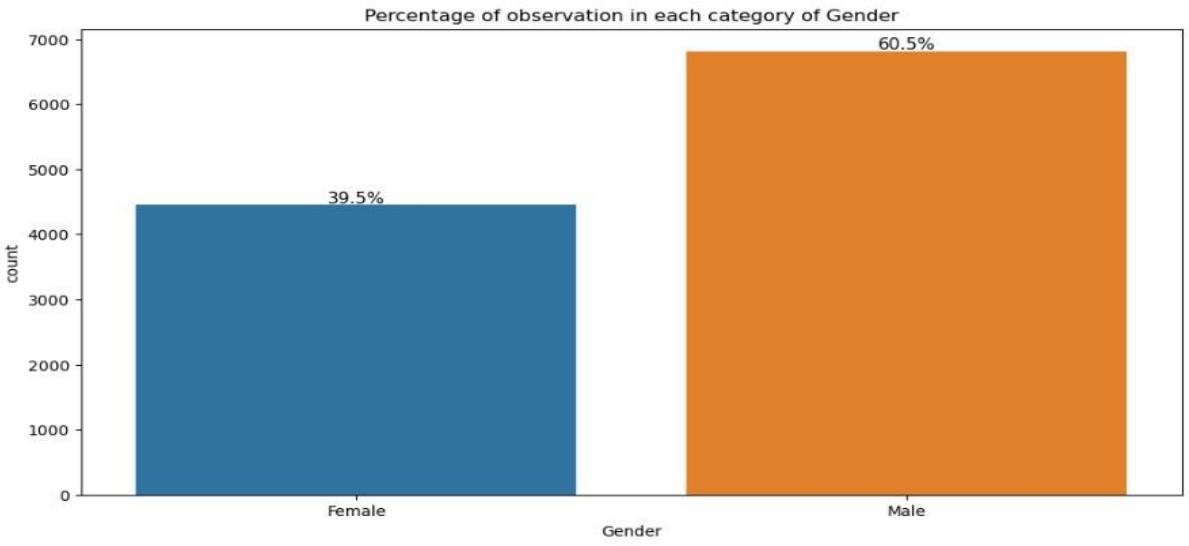
Let's look at each of the variable individually first to understand the data:

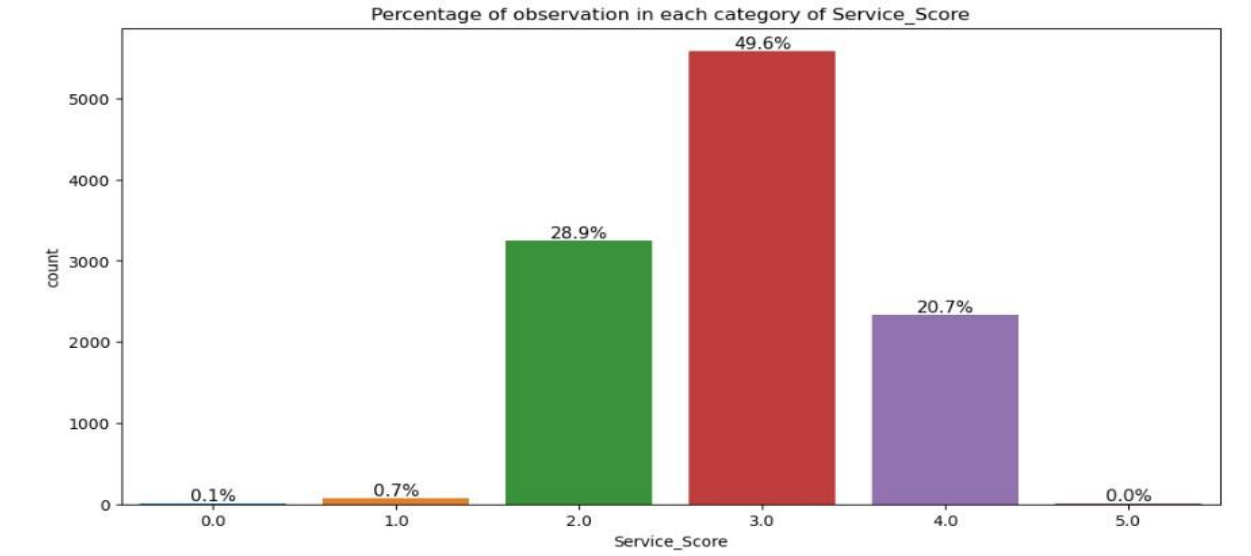
For variables that are categorical in nature, count plot chats have been used. And with the help of "For Loop" charts are plotted.

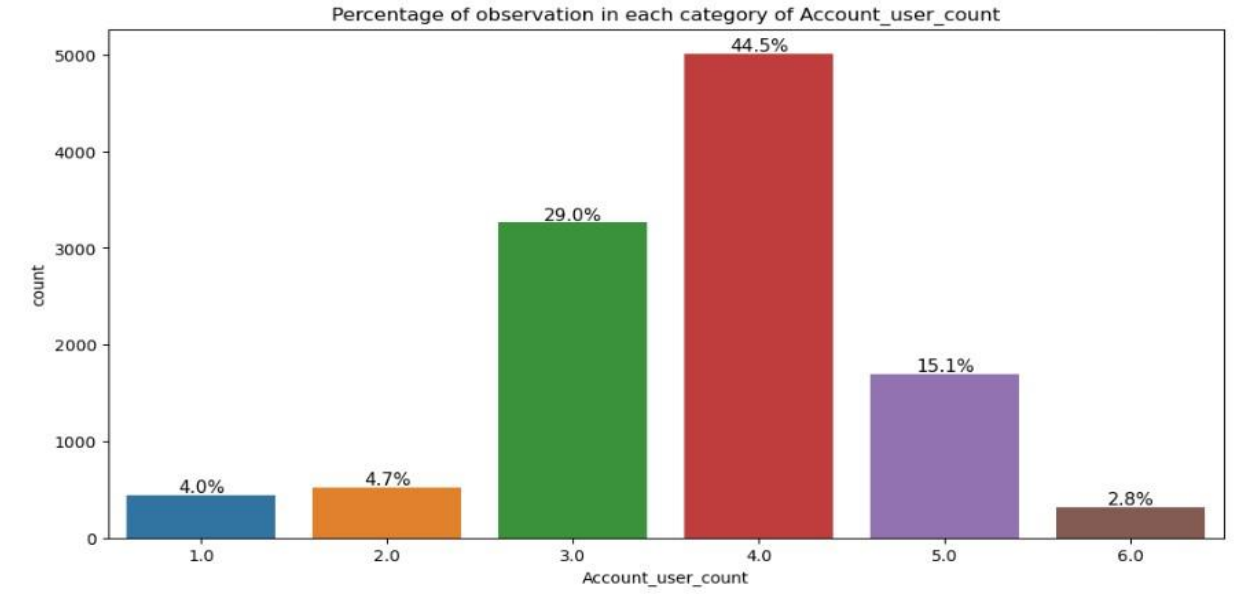


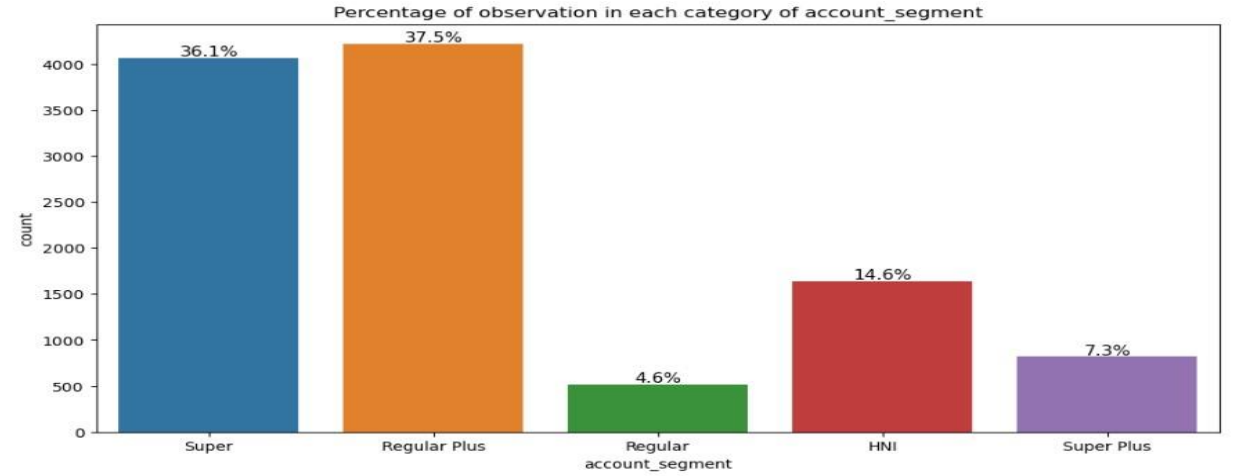

Percentage of observation in each category of City_Tier
64.5%
30.2%
5.3%
count
City_Tier
1.0
2.0
3.0

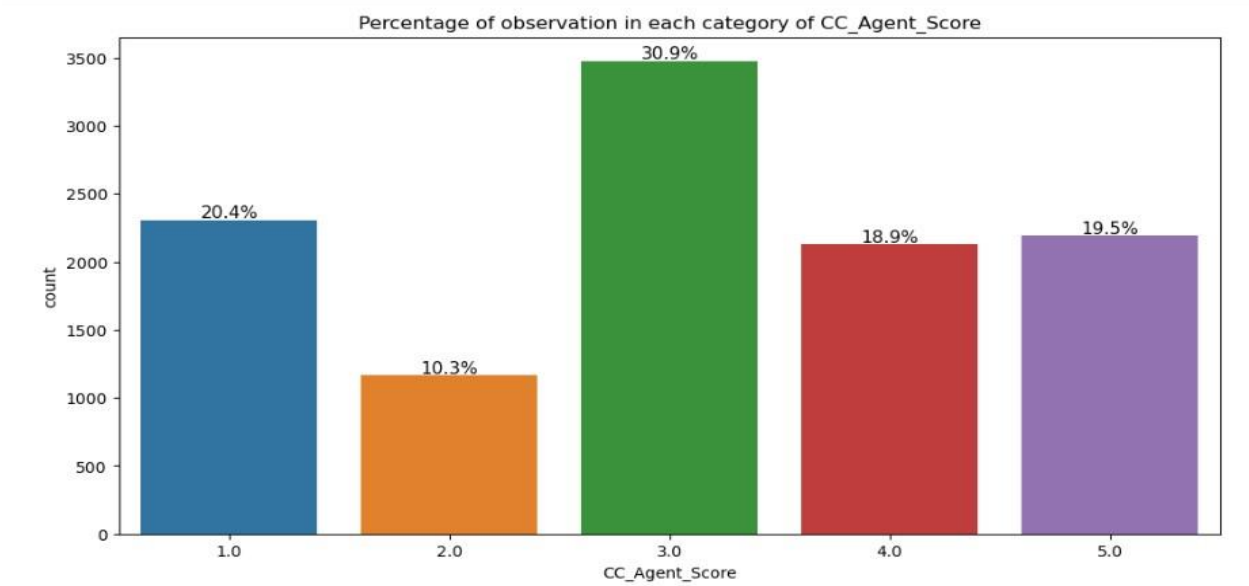

Percentage of observation in each category of Payment
41.7%
31.2%
7.3%
9.0%
10.8%
count
Payment
Debit Card
UPI
Credit Card
Cash on Delivery
E wallet

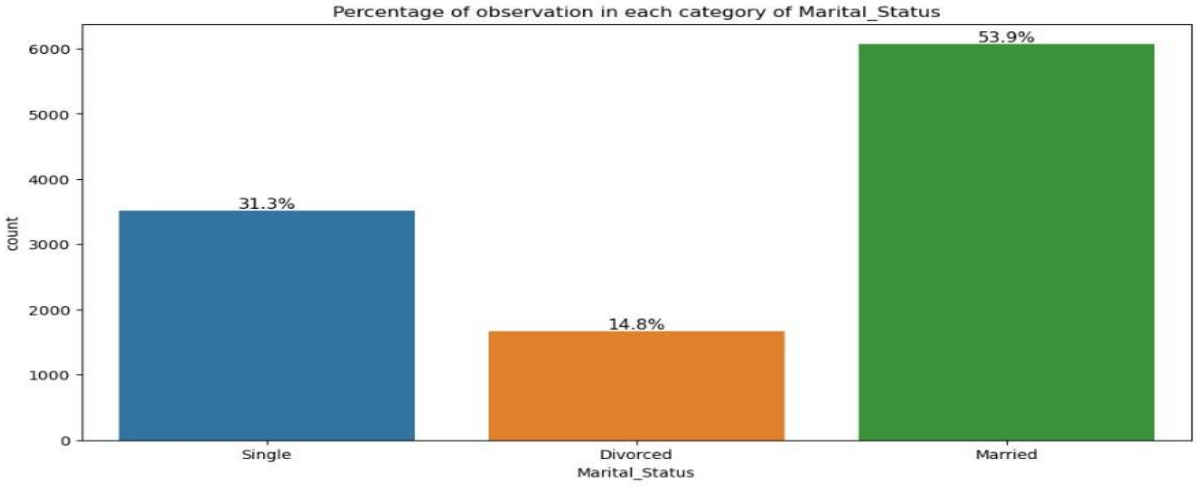

Percentage of observation in each category of Gender
60.5%
39.5%
count
Gender
Female
Male

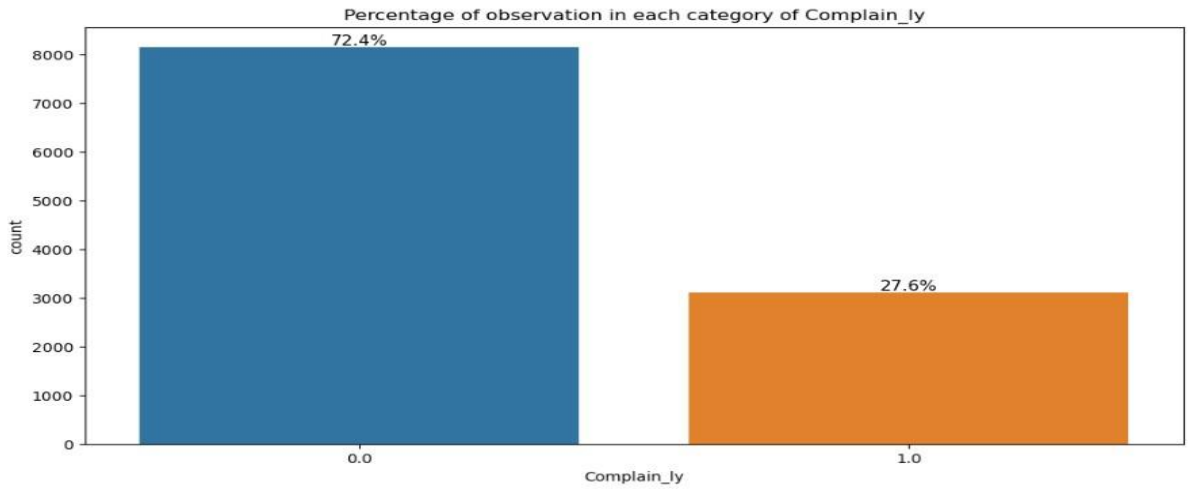

Percentage of observation in each category of Service_Score
count
Service_Score
0.1%
0.7%
28.9%
49.6%
20.7%
0.0%
0.0
1.0
2.0
3.0
4.0
5.0

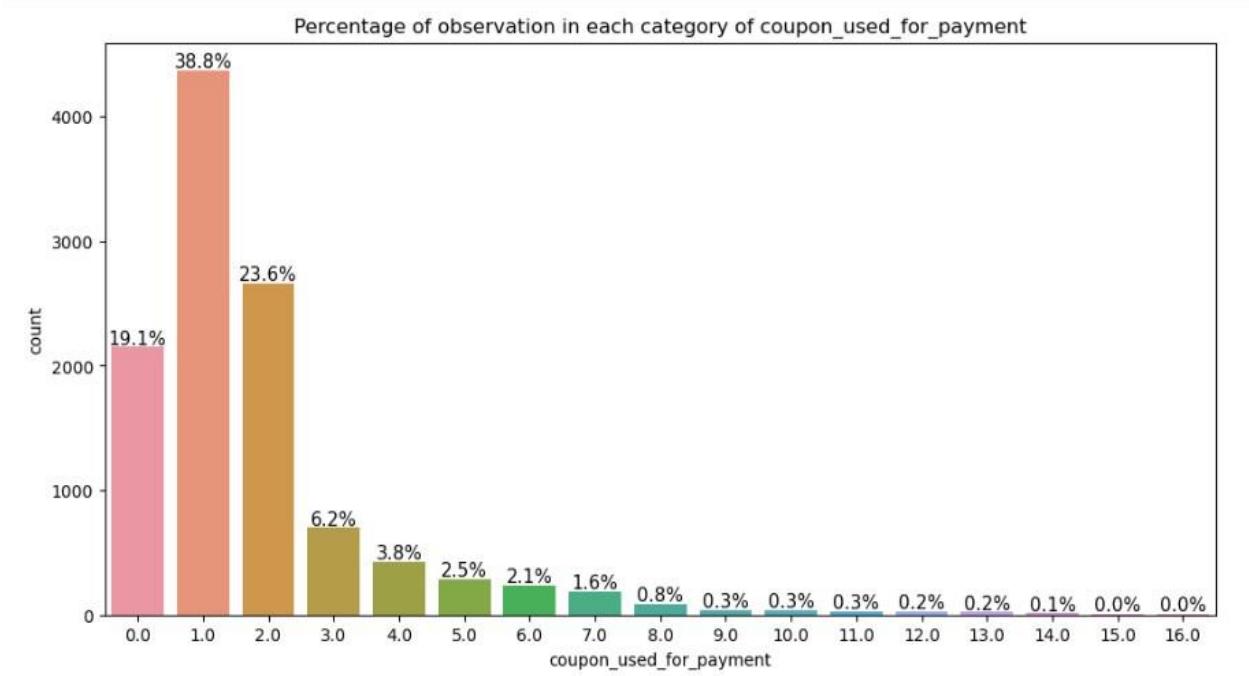

Percentage of observation in each category of Account_user_count
count
Account_user_count
4.0%
4.7%
29.0%
44.5%
15.1%
2.8%
1.0
2.0
3.0
4.0
5.0
6.0

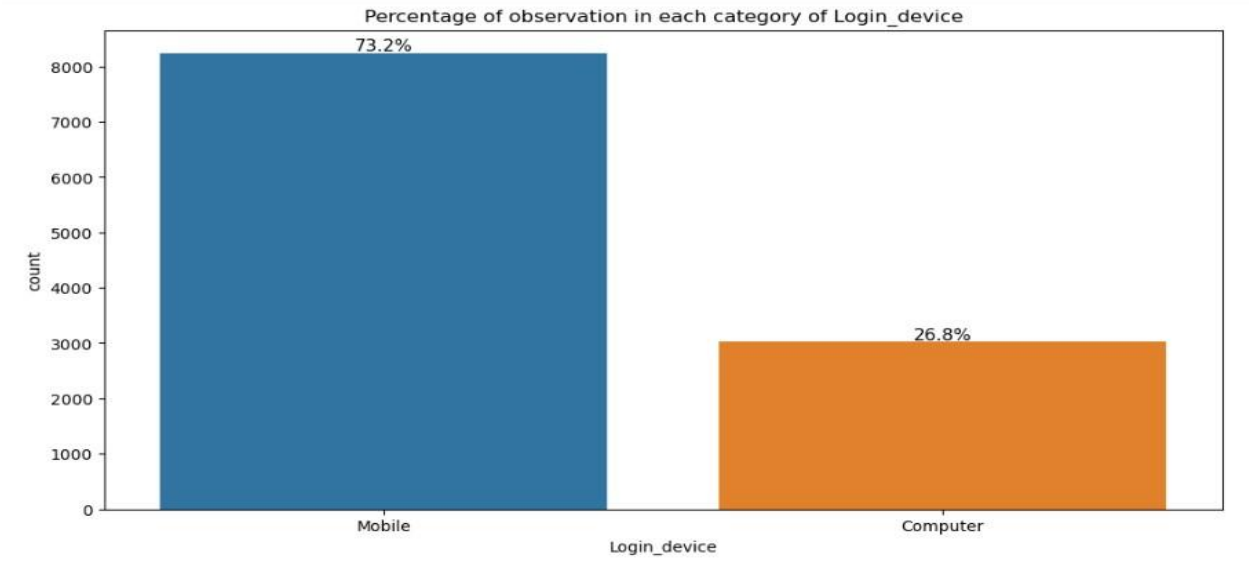

Percentage of observation in each category of account_segment
count
account_segment
36.1%
37.5%
4.6%
14.6%
7.3%
Super
Regular Plus
Regular
HNI
Super Plus


Percentage of observation in each category of CC_Agent_Score
count
CC_Agent_Score
20.4%
10.3%
30.9%
18.9%
19.5%
1.0
2.0
3.0
4.0
5.0

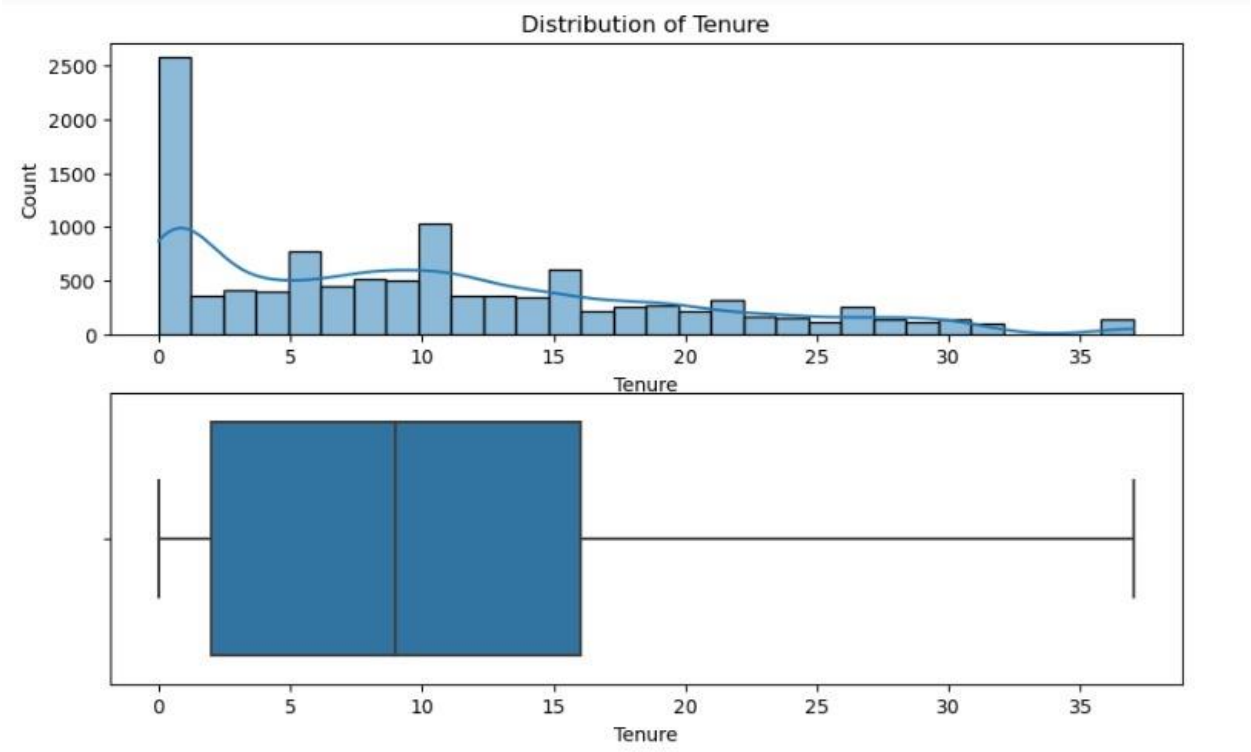

Percentage of observation in each category of Marital_Status
count
Marital_Status
31.3%
14.8%
53.9%
Single
Divorced
Married

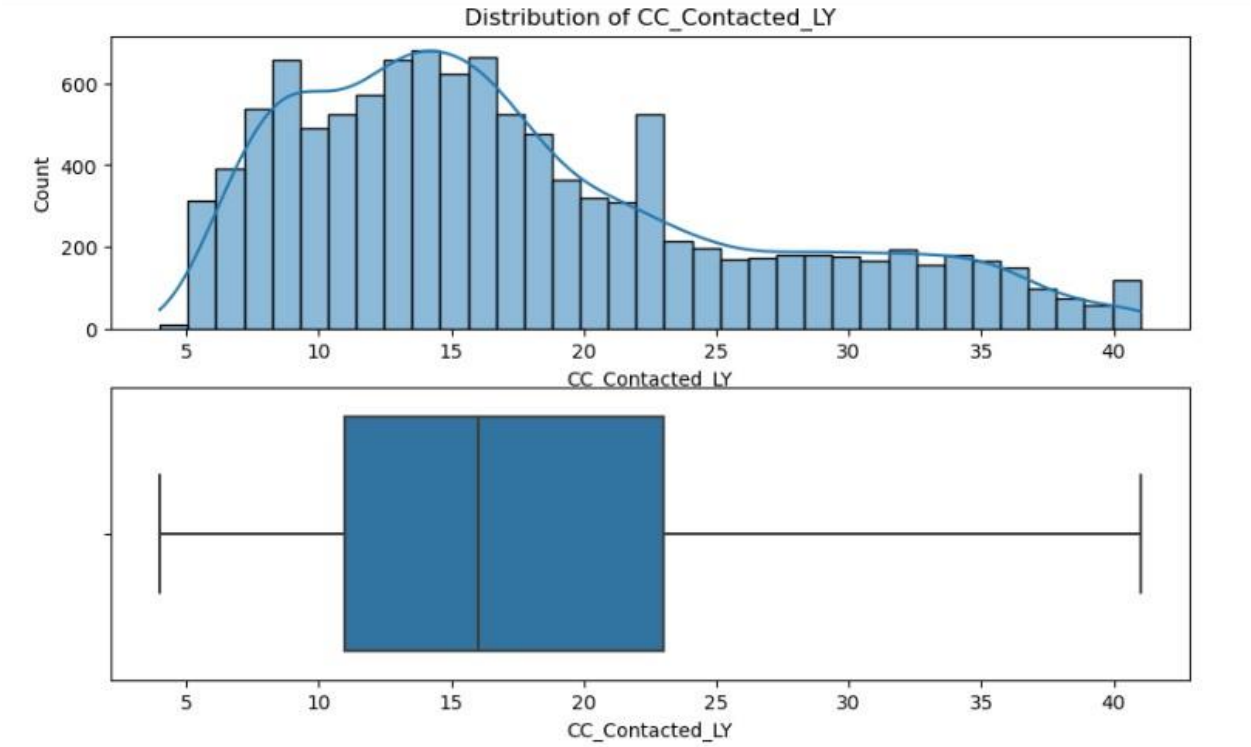

Percentage of observation in each category of Complain_ly
count
Complain_ly
72.4%
27.6%
0.0
1.0

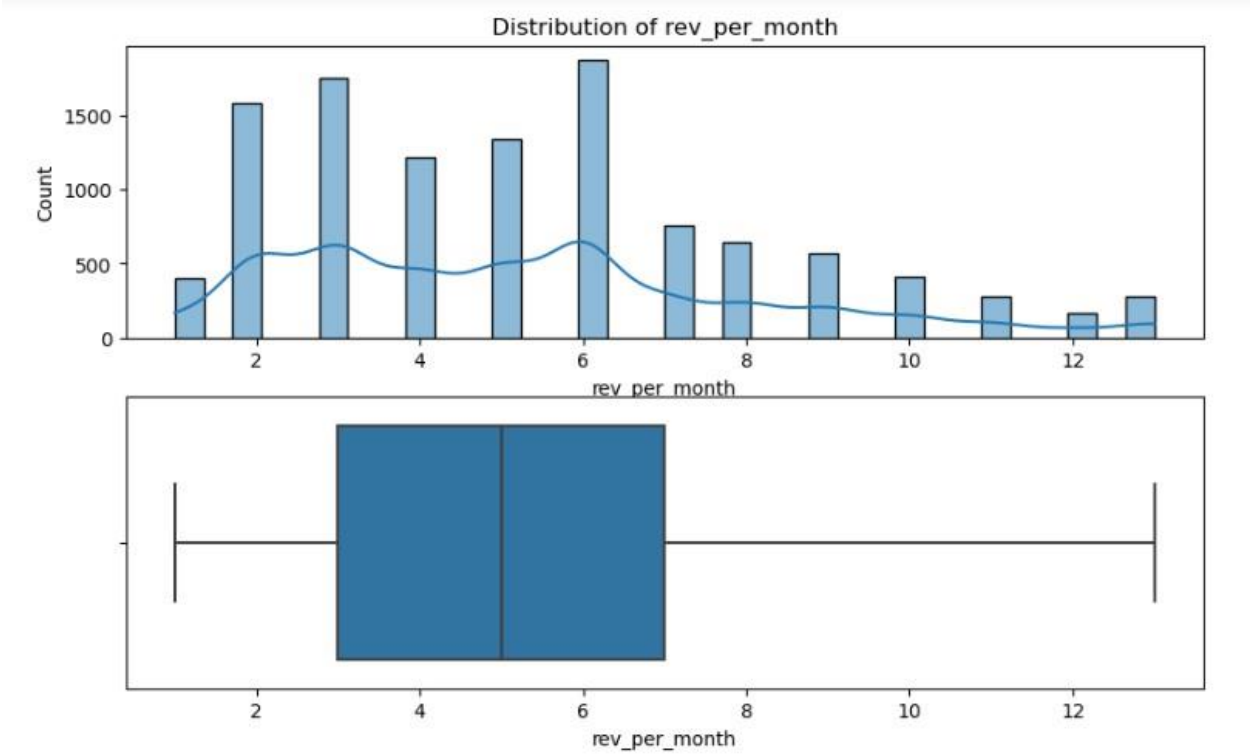


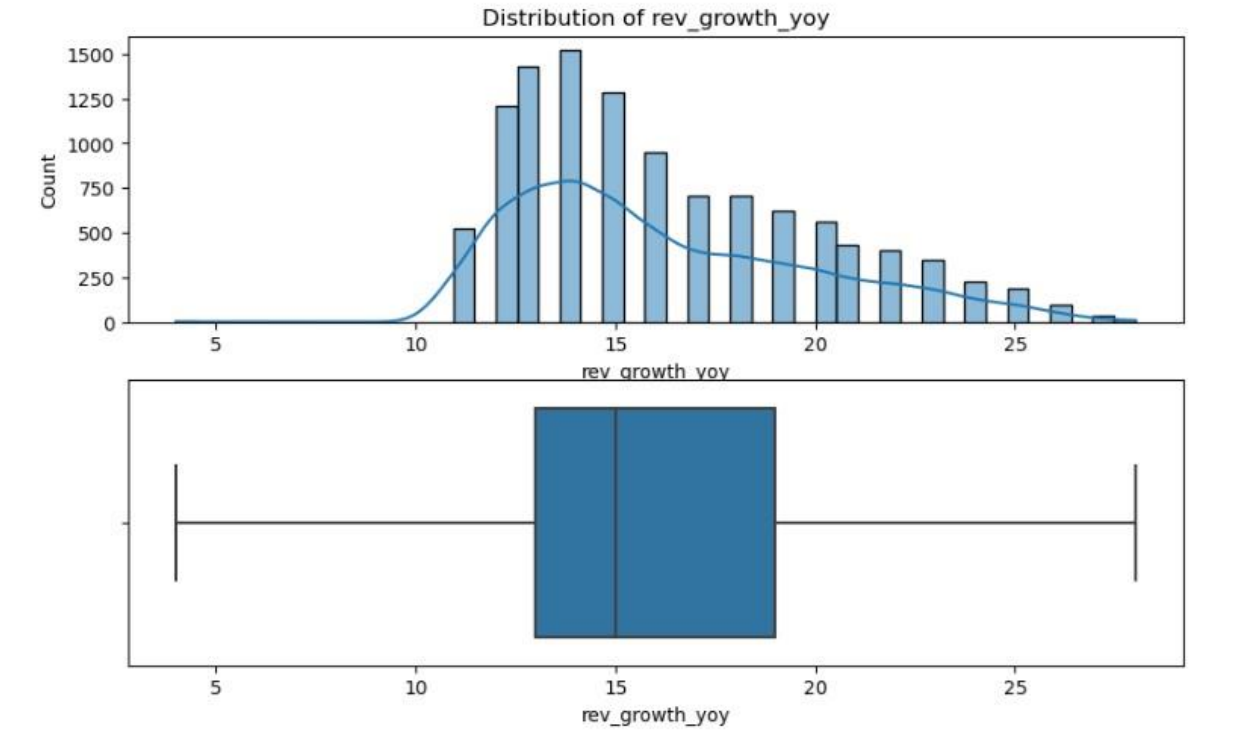


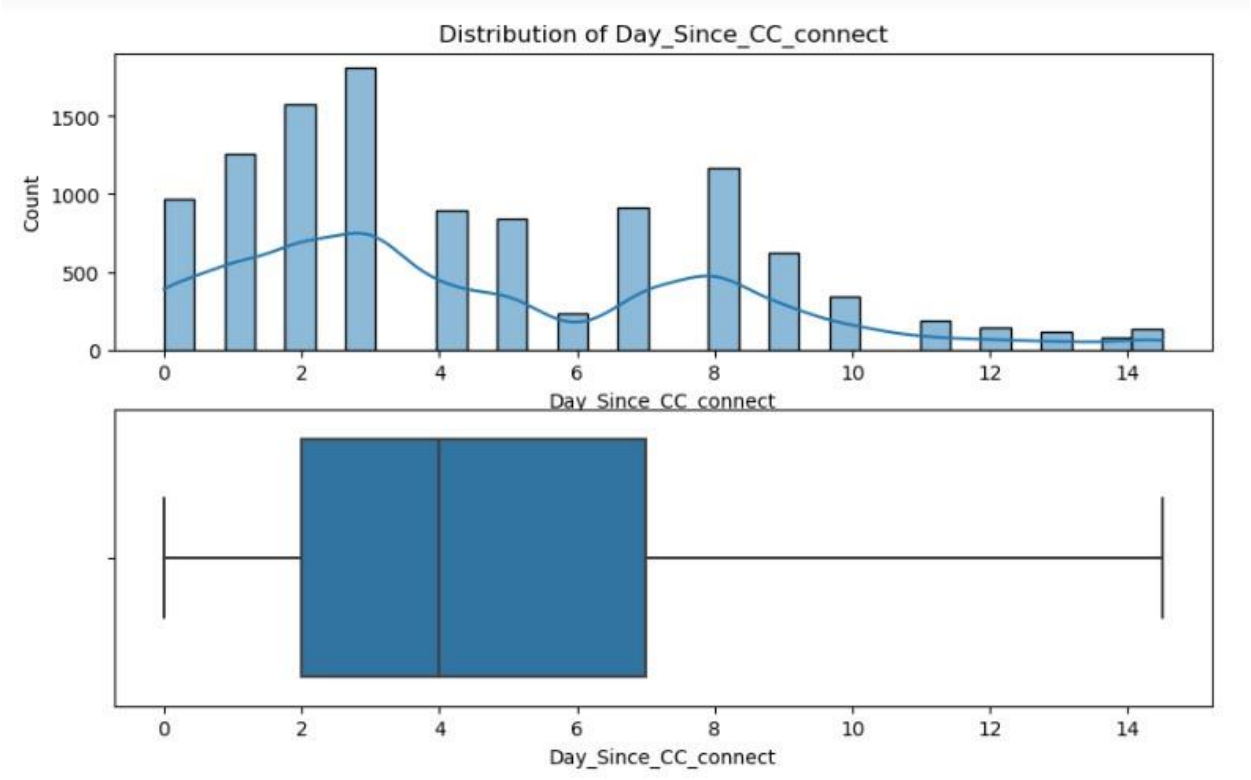
For variables which are continuous in nature are histogram and Boxplot charts are used to graphically represent the distribution and 5-point summary. Again, a For Loop is used to plot the charts.

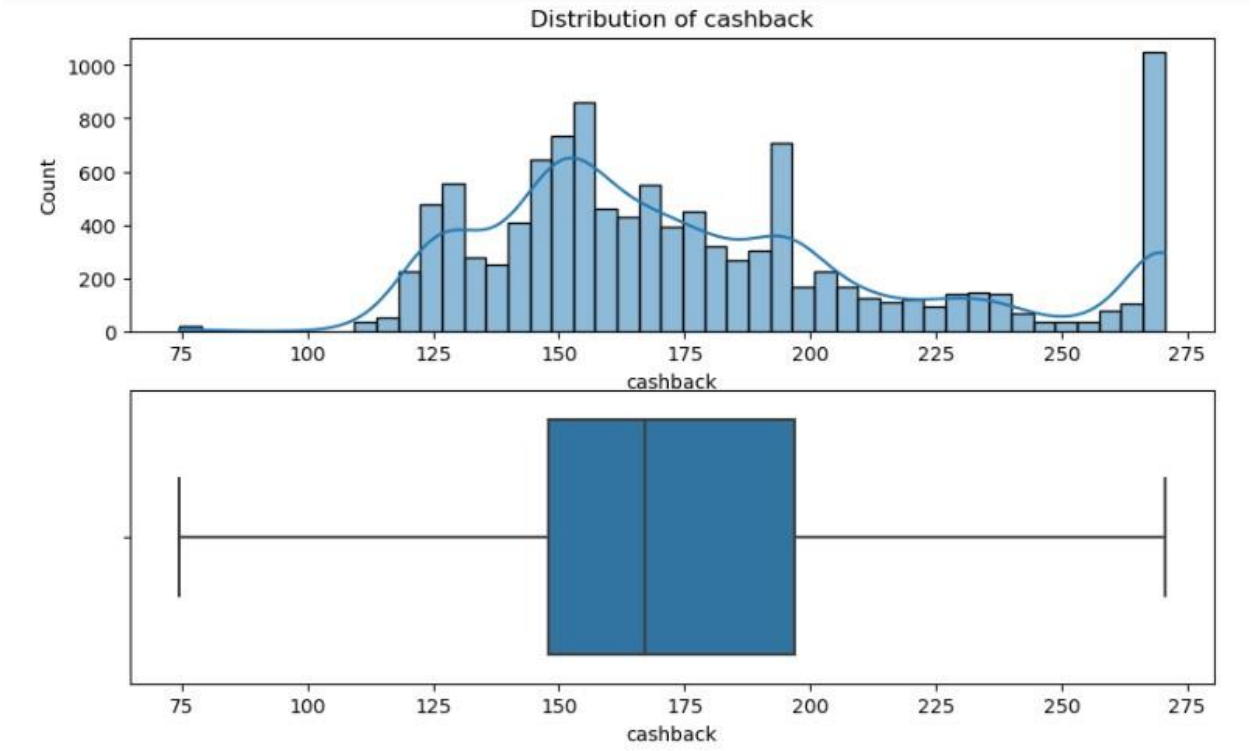

Distribution of Tenure
Count
Tenure
Tenure


Distribution of CC_Contacted_LY
Count
CC_Contacted_LY
CC_Contacted_LY


Distribution of rev_per_month
Count
1500
1000
500
0
2
4
6
8
10
12
rev_per_month
rev_per_month


Distribution of rev_growth_yoy
Count
1500
1250
1000
750
500
250
0
5
10
15
20
25
rev_growth_yoy
rev_growth_yoy


Distribution of Day_Since_CC_connect
Count
1500
1000
500
0
0
2
4
6
8
10
12
14
Day_Since_CC_connect
0
2
4
6
8
10
12
14
Day_Since_CC_connect

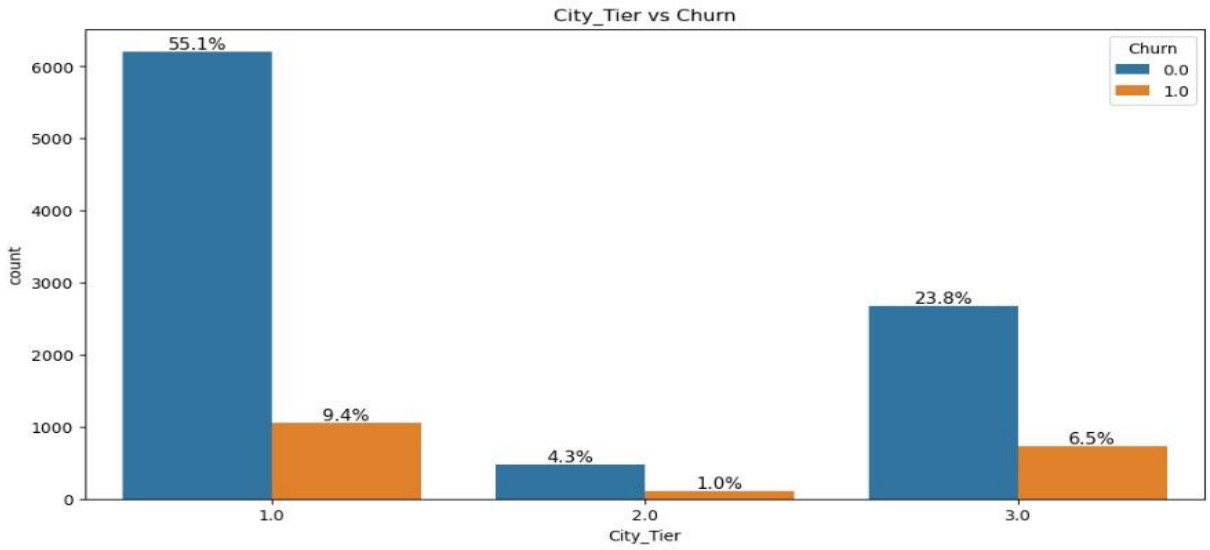

Distribution of cashback
Count
1000
800
600
400
200
0
75
100
125
150
175
200
225
250
275
cashback
75
100
125
150
175
200
225
250
275
cashback

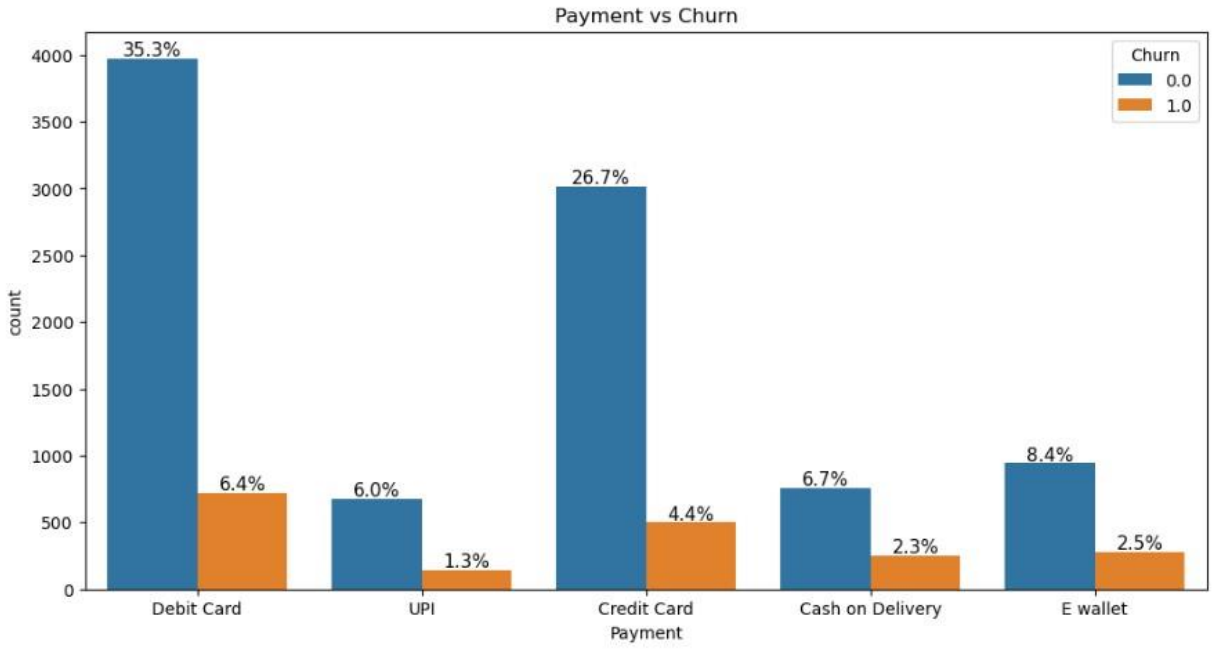
Insights from univariate analysis:

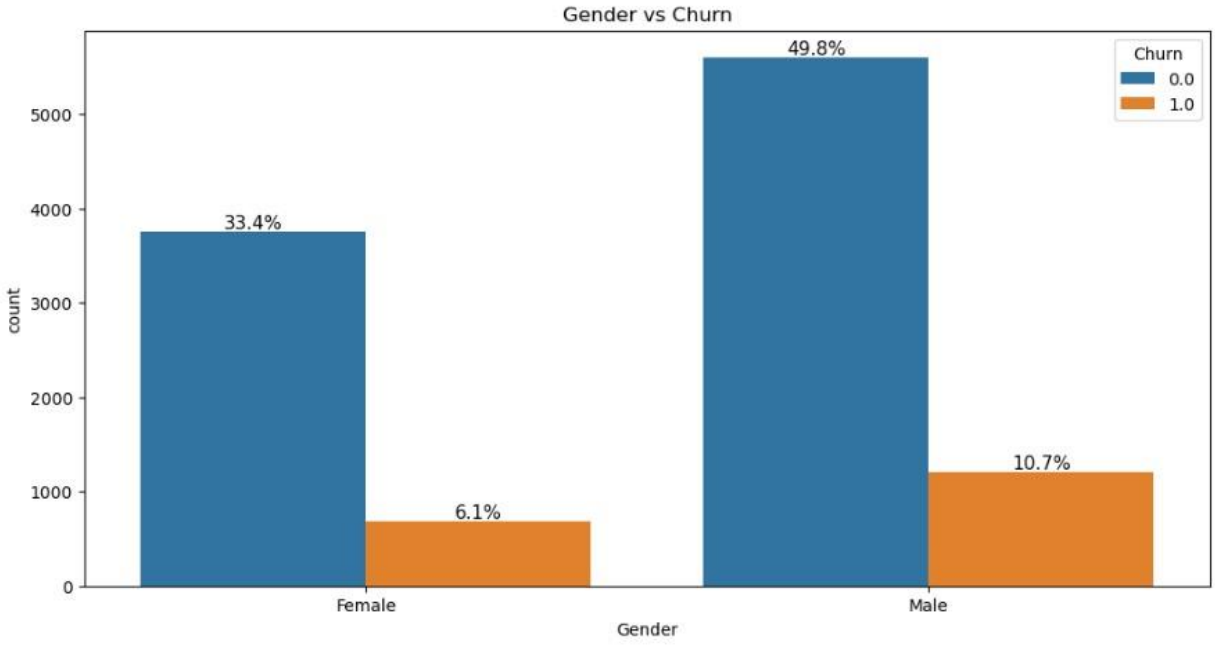
- Only 17% of the customers of the total are churned.
- Most of the population is from tier 1 cities followed by tier 3 cities. Only about 5% are from tier 2 cities which shows a great opportunity of expansion.
- About 70% of the customers prefer to pay the bills through credit/debit cards.
- Majority of customers are male (60.5%).
- 50% rated average in service score. 29% rated below average and only 21% rated above average.
- 96% of the customers have 1 or more users associated with the same account.
- Super and regular plus are the most preferred segments by users. Almost 73% of the customers are in these 2 categories.
- 30% of customers rated support agents below average. This needs to be taken into consideration. We need to work on training and quality of the agents.
- Only 27.6% of the customers raised any complaint in the last 1 year.
- 42% of customers paid their bills through coupons more than once. Coupons are popular and can be used for promoting campaigns and products.
- Mobile is the preferred device for most of the customers.

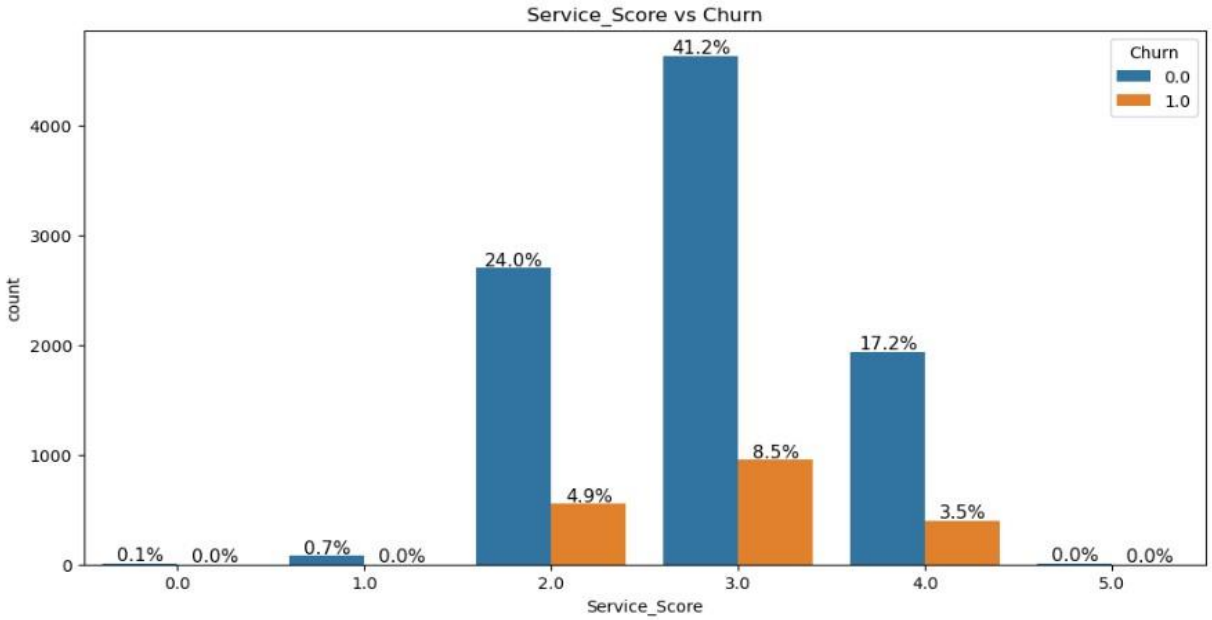
Bivariate Analysis:

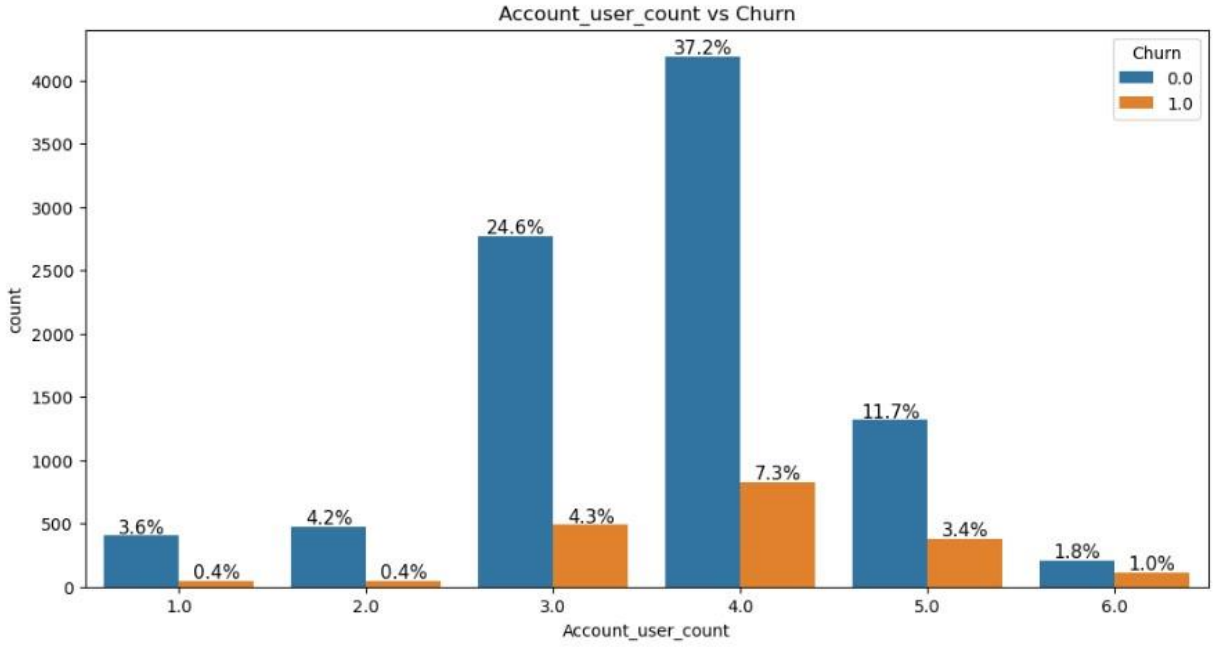
In bivariate analysis we try to explore the relation between two variables. For example, City_Teir of users and churned users.

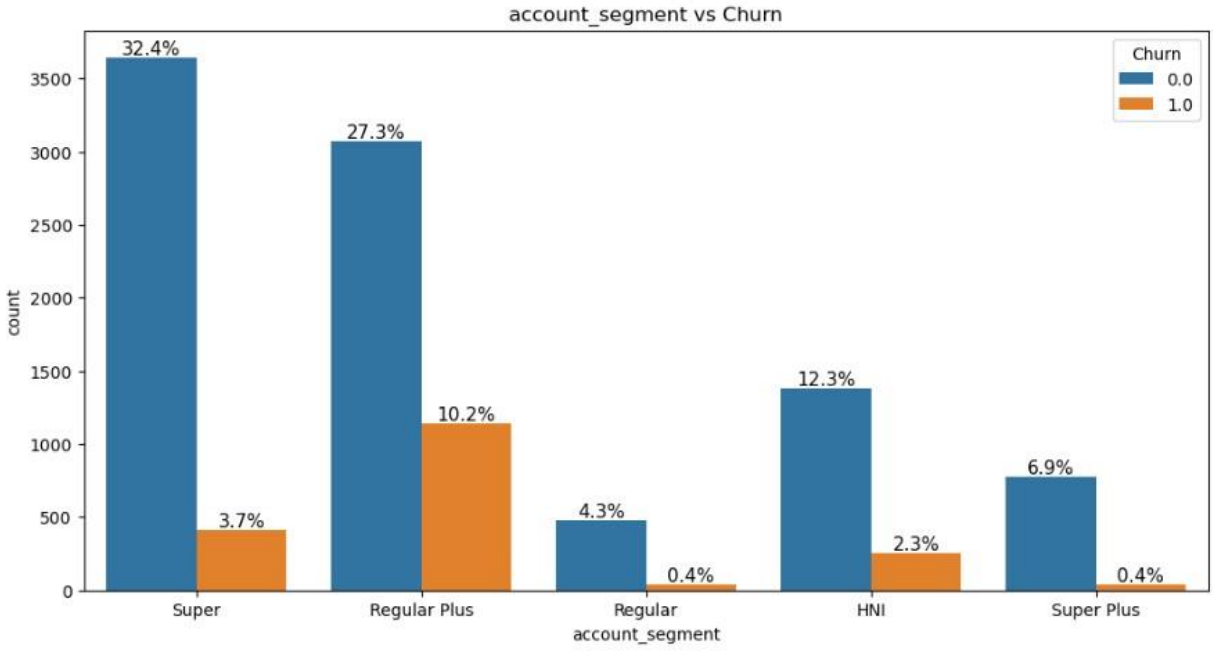


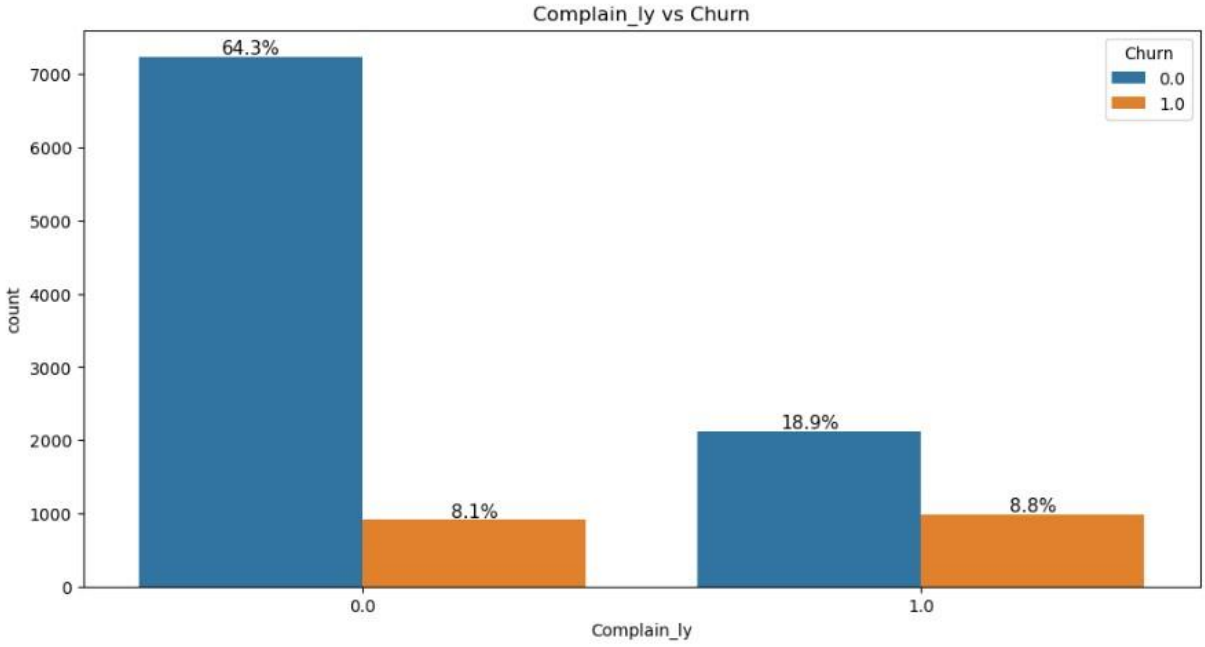

Payment vs Churn
Churn
0.0
1.0
count
Payment
Debit Card
UPI
Credit Card
Cash on Delivery
E wallet
35.3%
6.4%
6.0%
1.3%
26.7%
4.4%
6.7%
2.3%
8.4%
2.5%

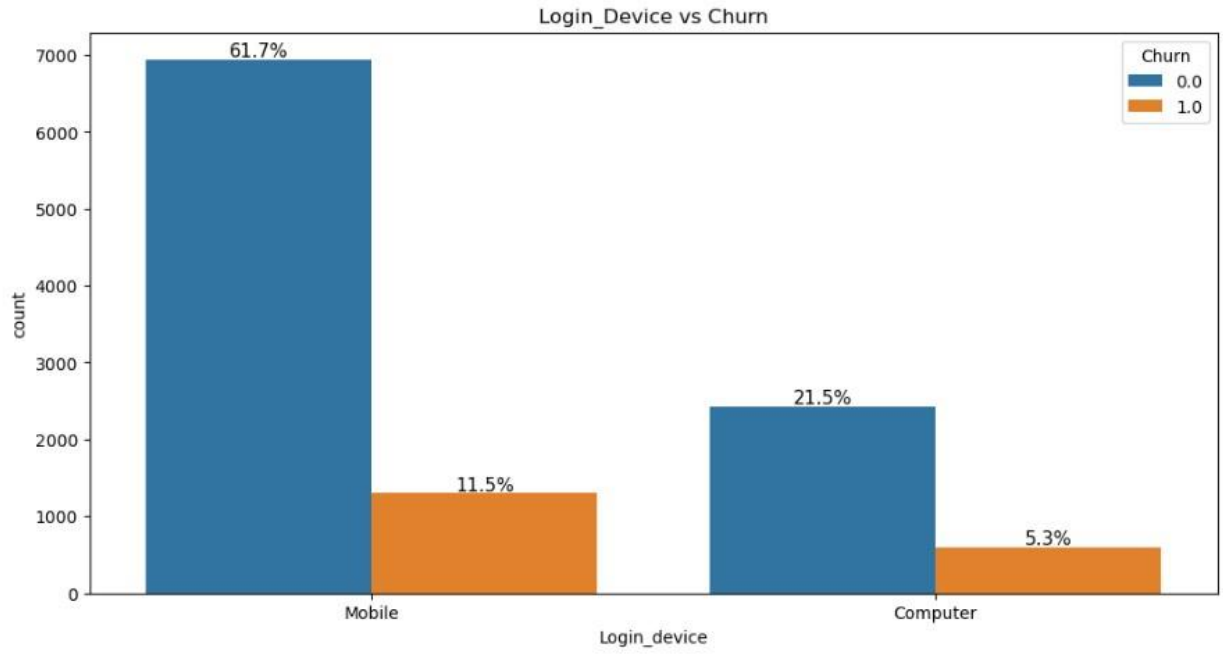

Gender vs Churn
Churn
0.0
1.0
count
Gender
Female
Male
33.4%
6.1%
49.8%
10.7%


Service_Score vs Churn
Churn
0.0
1.0
count
Service_Score
0.1%
0.0%
0.7%
0.0%
24.0%
4.9%
41.2%
8.5%
17.2%
3.5%
0.0%
0.0%


Account_user_count vs Churn
Churn
0.0
1.0
count
Account_user_count
3.6%
0.4%
4.2%
0.4%
24.6%
4.3%
37.2%
7.3%
11.7%
3.4%
1.8%
1.0%

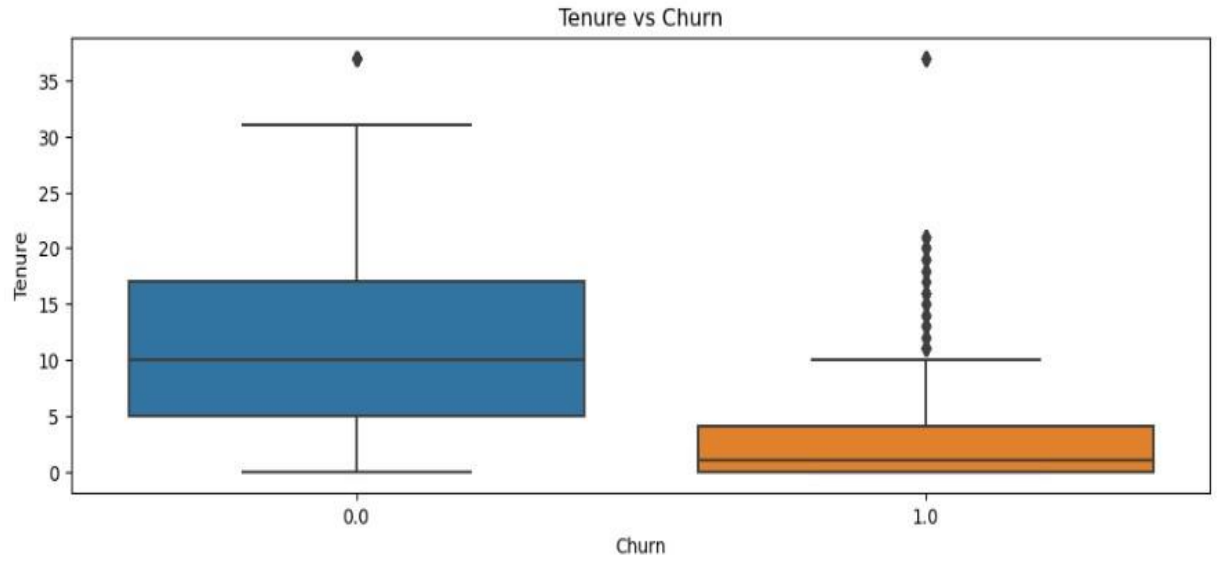

account_segment vs Churn
Churn
0.0
1.0
count
32.4%
3.7%
27.3%
10.2%
4.3%
0.4%
12.3%
2.3%
6.9%
0.4%
Super
Regular Plus
Regular
HNI
Super Plus
account_segment

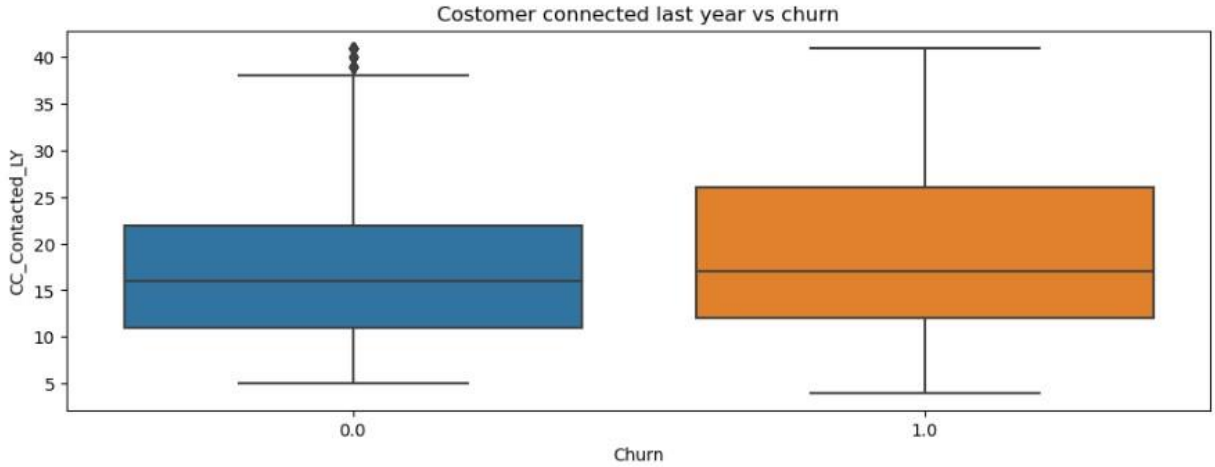

Complain_ly vs Churn
Churn
0.0
1.0
count
64.3%
8.1%
18.9%
8.8%
0.0
1.0
Complain_ly

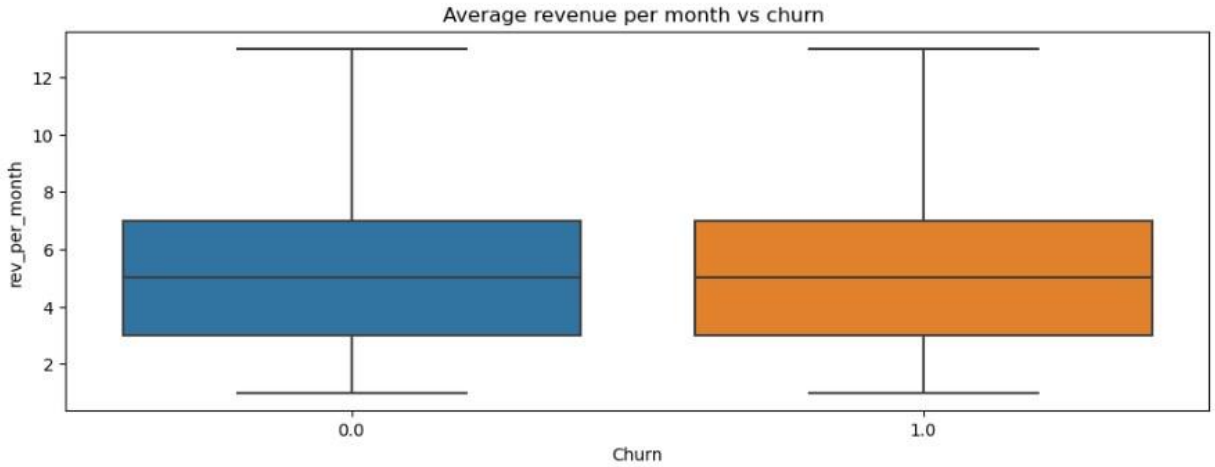


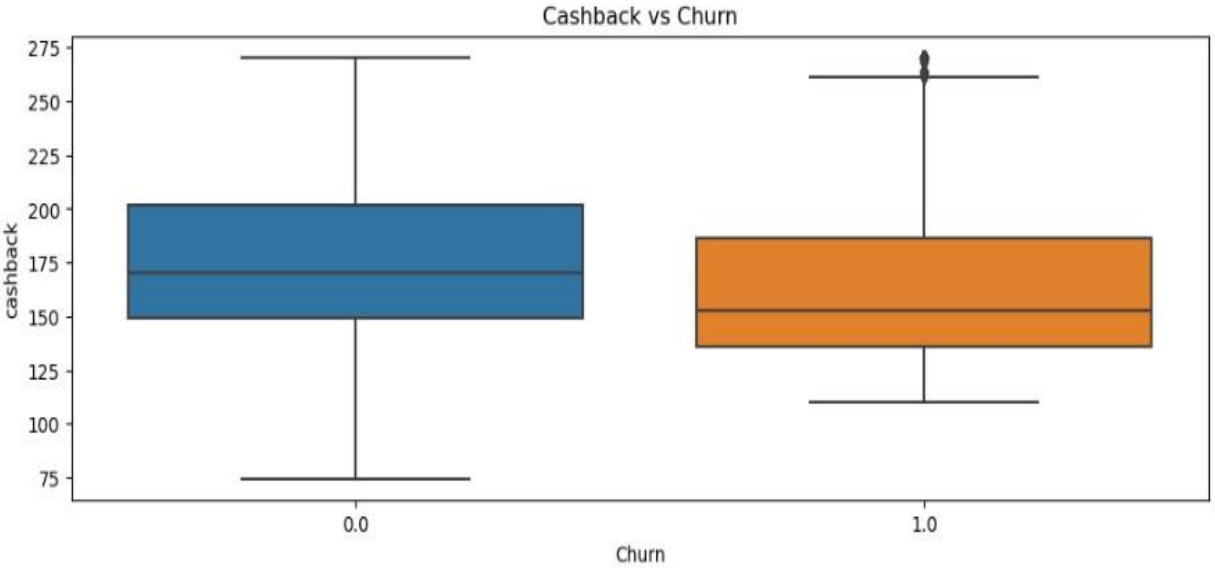
Bivariate analysis of churn variables with the continuous variables. This is important in 2 ways.

- How the continuous variable affects the classification of the user as churned or not.
- And it shows the behavioural patterns of users.




Costomer connected last year vs churn
CC_Contacted_LY
40
35
30
25
20
15
10
5
0.0
1.0
Churn


Average revenue per month vs churn
rev_per_month
12
10
8
6
4
2
0.0
1.0
Churn

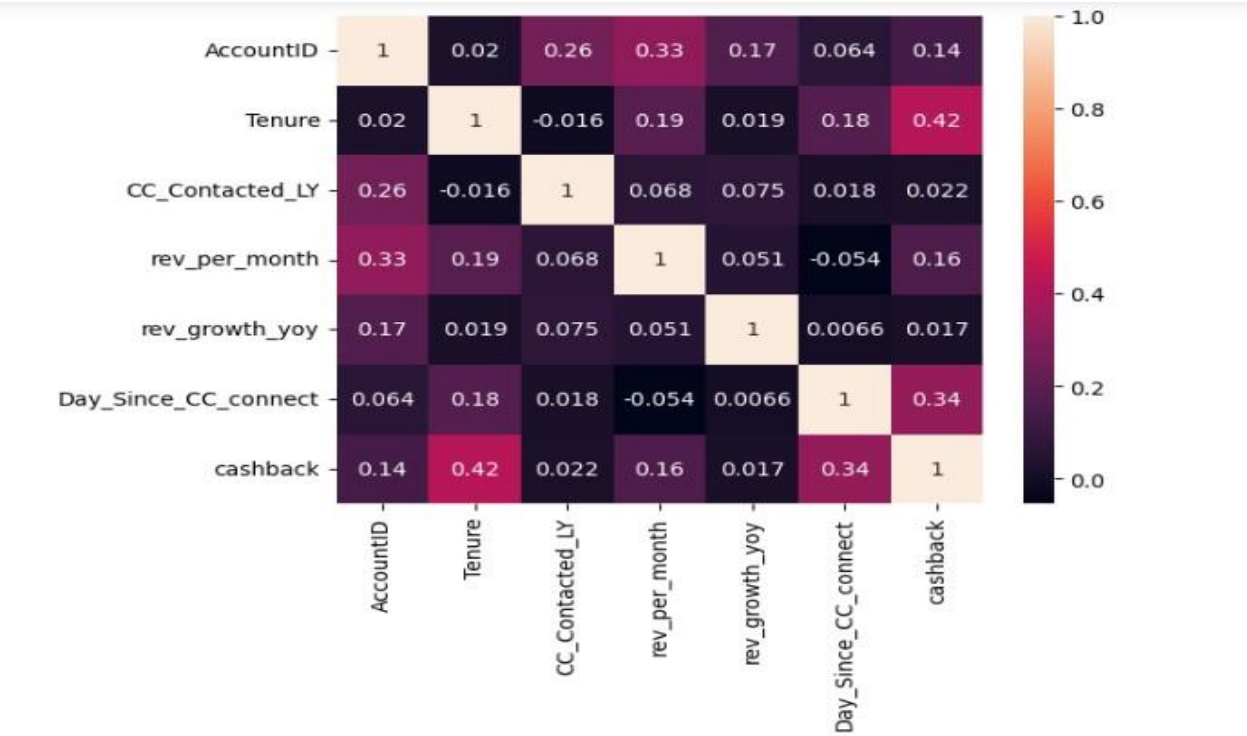

Cashback vs Churn
cashback
275
250
225
200
175
150
125
100
75
0.0
1.0
Churn

Insights from bivariate analysis:

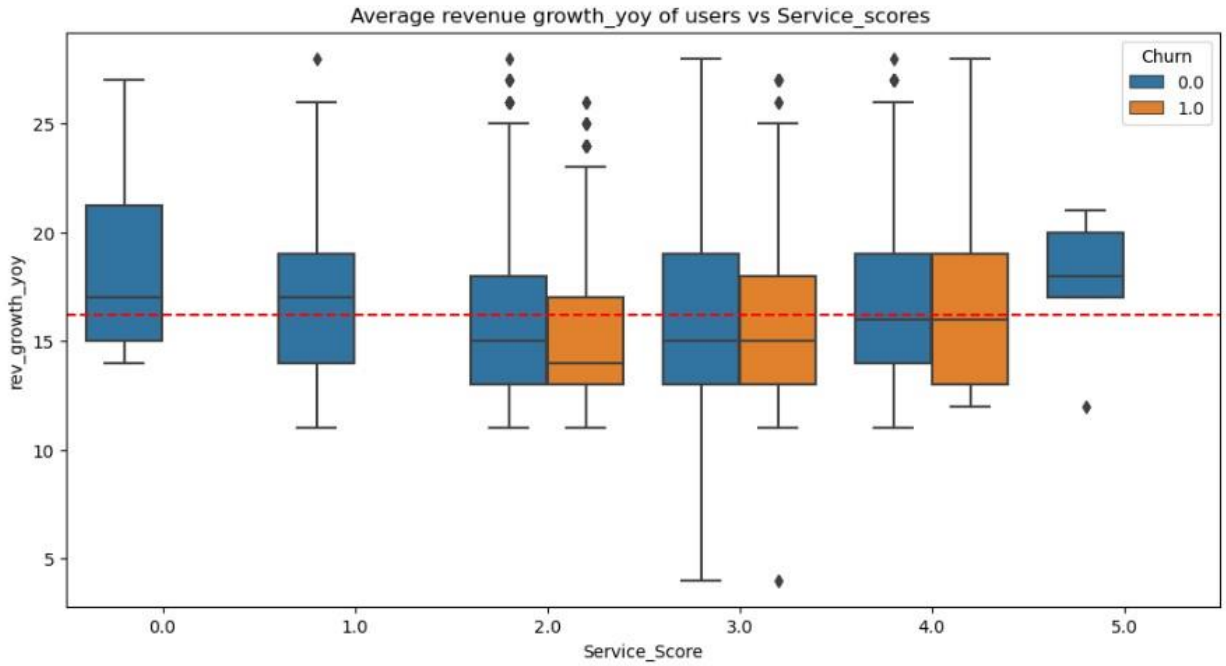
- Almost 10% percent of the total users churned are from tier 1 cities.
- Out of 17% churned users, 10% used credit/debit cards.
- 10.7% male and 6% are female who churned.
- 15% of the total user's churned accounts have 3, 4 and 5 number of users. This has a serious business implication.
- Although regular plus is one of the most popular segments, most of the churned users are from this segment only.
- Customers who are churned have lower median tenure than the users who are not churned.
- Churned users connected to customer care more than the users not churned.
- Average revenue per month is almost the same for both categories.
- Midian cashback earned by churned users is lower than the non-churned users.

Multivariate Analysis:

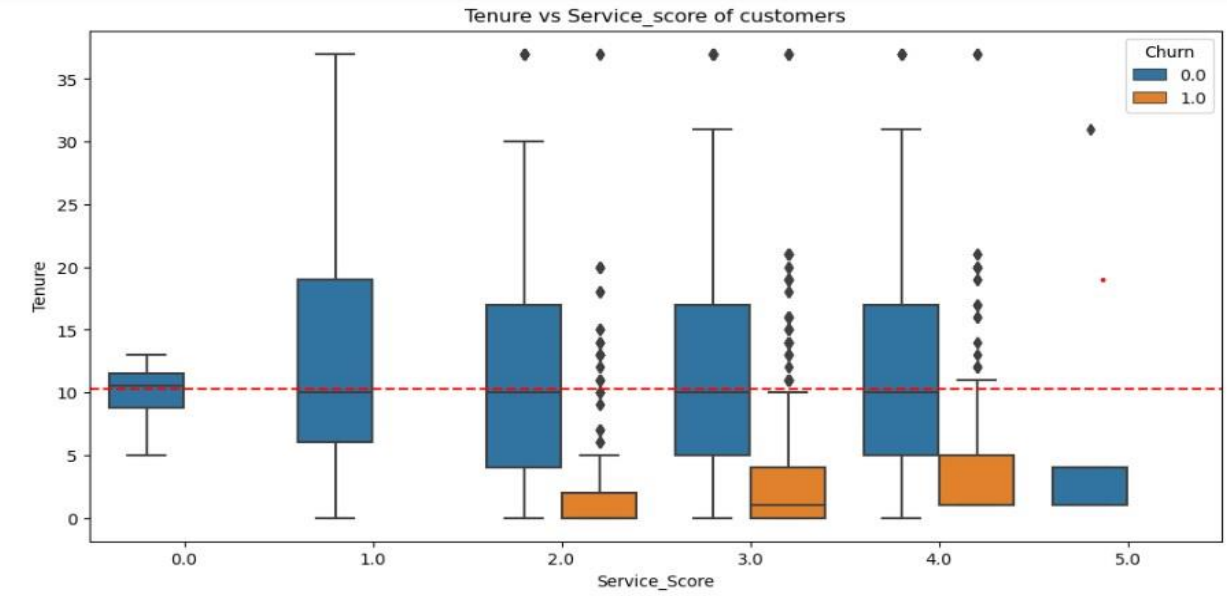
The relationship between multiple variables is explored in this section. Finding the correlation between different variables.



This heatmap chart shows the correlation between the variables which are continuous in nature. This correlation chart shows that the variables have low to very low correlation between the variables.



Here, the red dashed line shows the population average of the revenue growth year on year. This chart clearly shows that the customers rated Service_Score 2 has least median revenue growth and are most likely to churn.

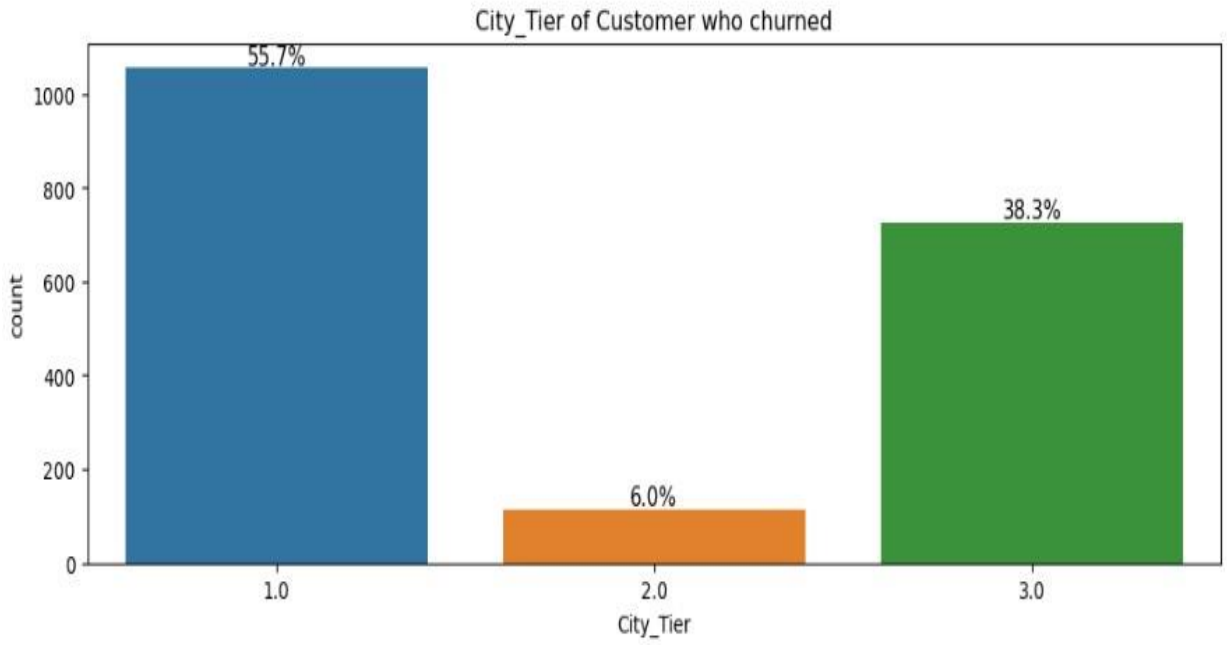


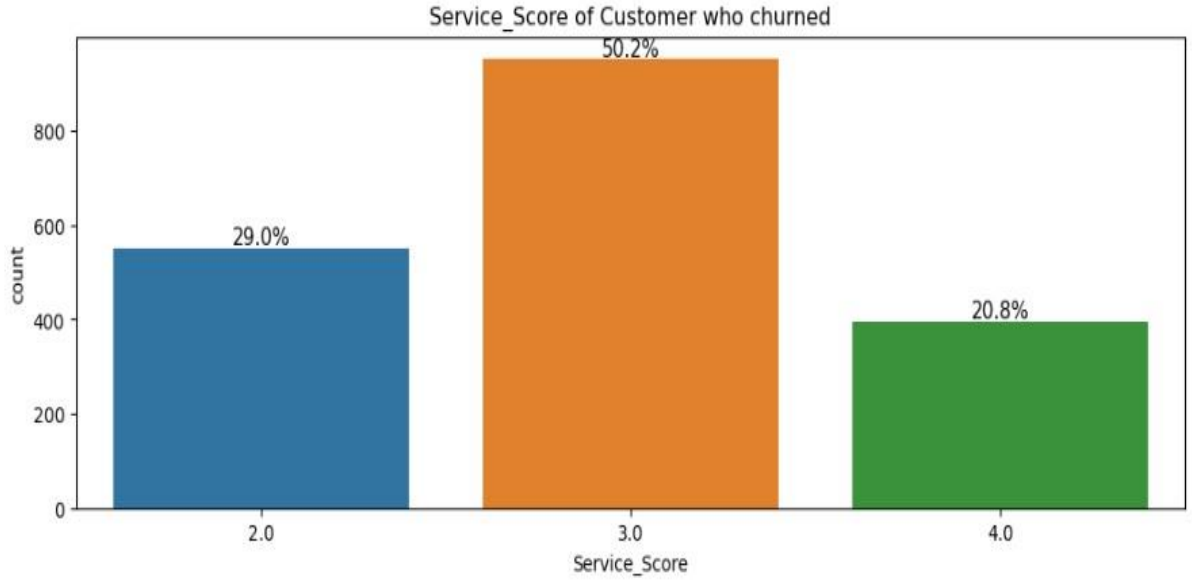
This chart shows the tenure of churned customers is lower than non-churned customers. And as compared to the population average the tenure is very low for the churned customer.

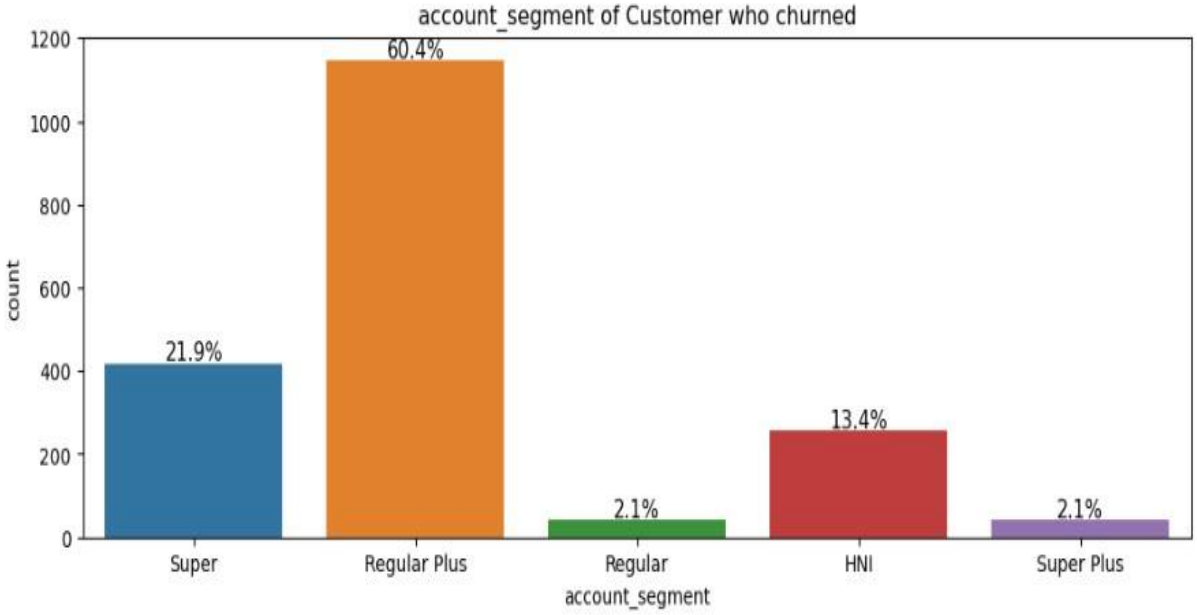
Segmentation:

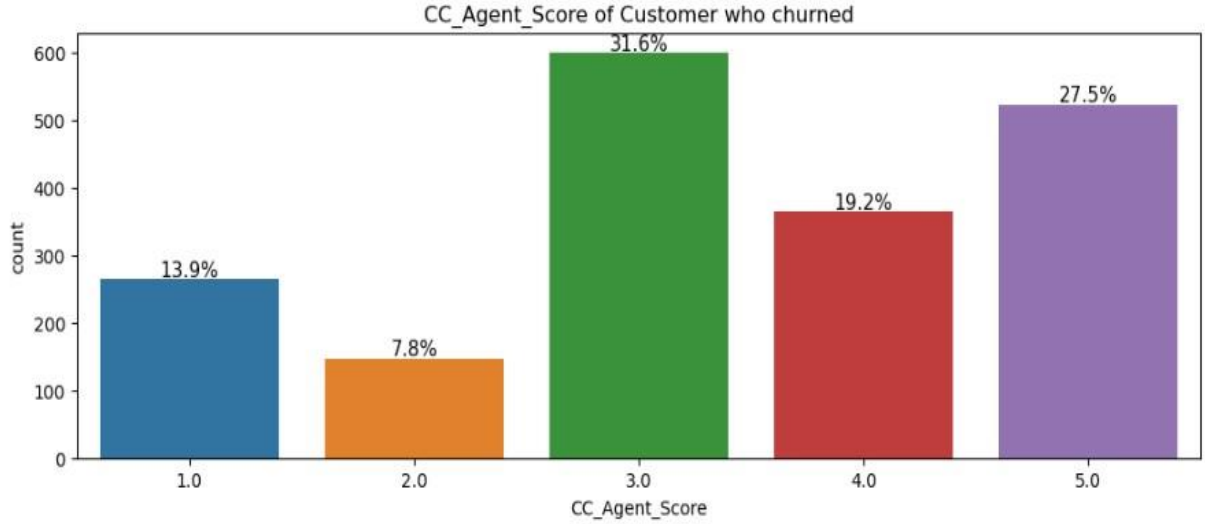
On segmenting the dataset into churned and non-churned customers we get 2 different datasets so that we can analyse the behaviour of the customers who churned.

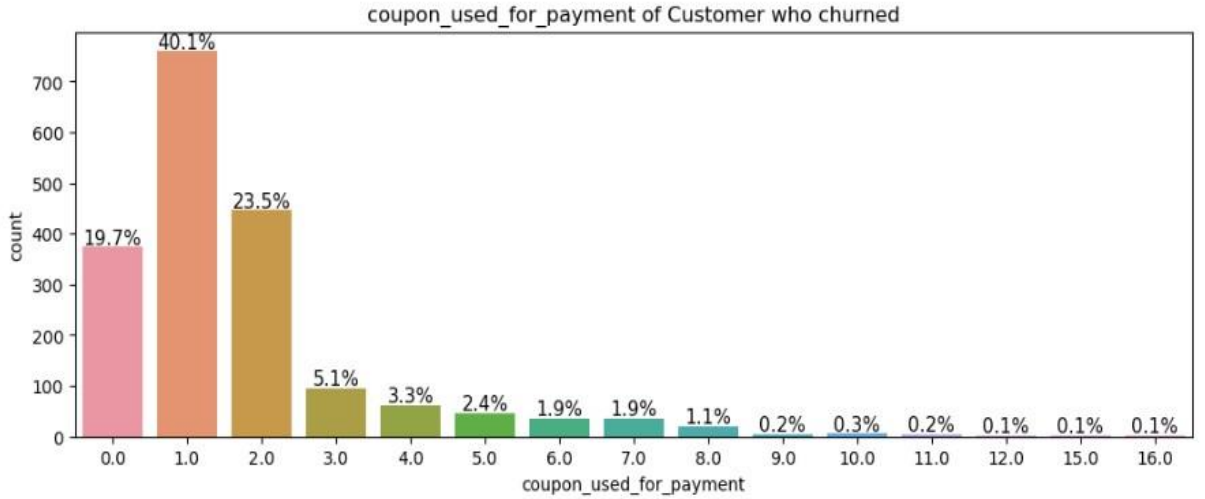
Here are some charts to show the analysis.




Service_Score of Customer who churned
50.2%
29.0%
20.8%
800
600
400
200
0
count
2.0
3.0
4.0
Service_Score

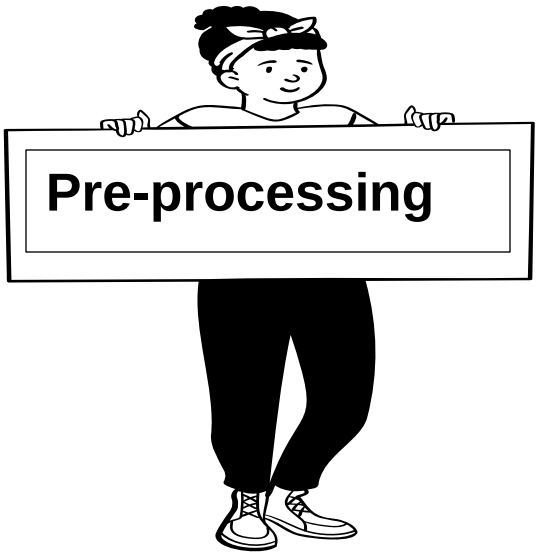

account_segment of Customer who churned
60.4%
21.9%
13.4%
2.1%
2.1%
1200
1000
800
600
400
200
0
count
Super
Regular Plus
Regular
HNI
Super Plus
account_segment



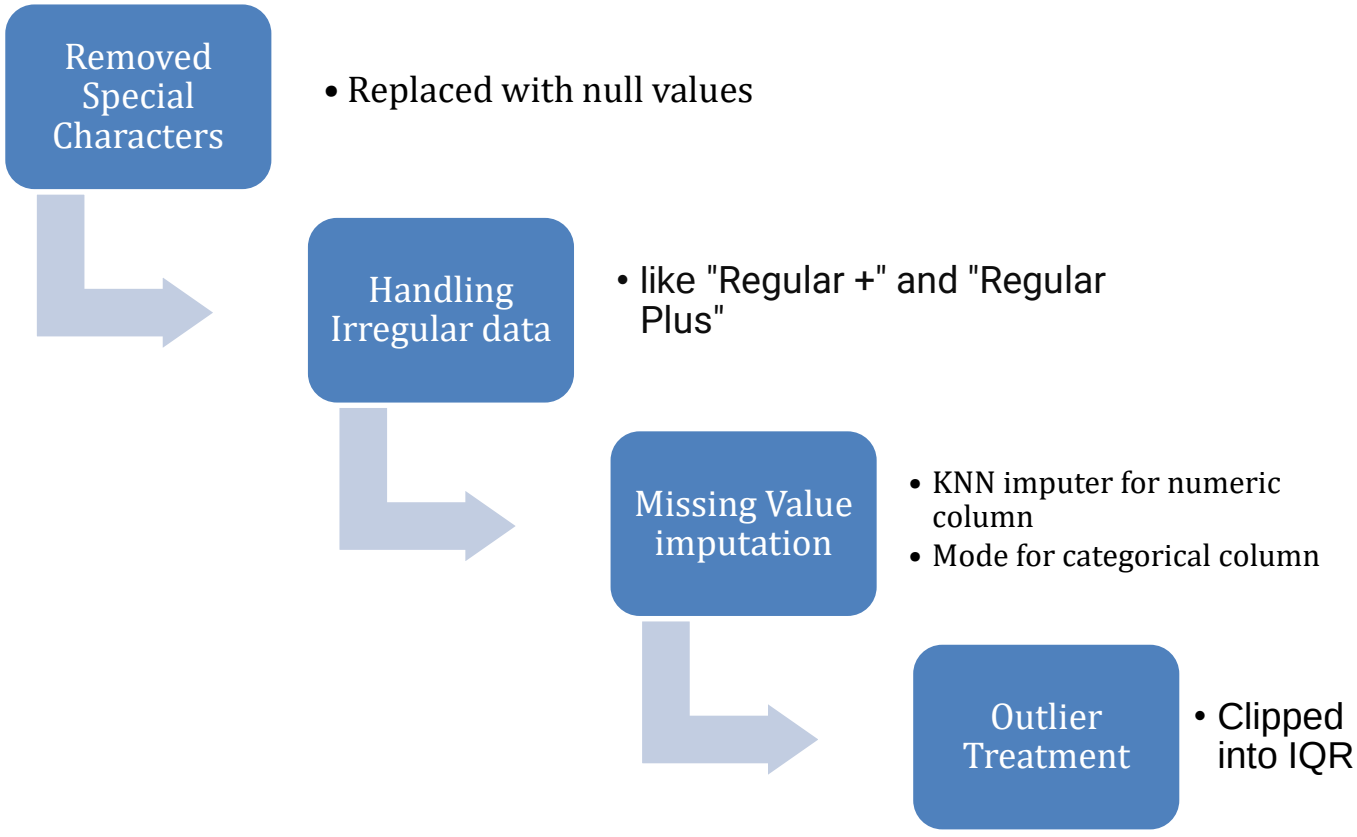


Insights from segmentation:

- 50% of the churned customers are from tier 1 cities.
- 70% of the churned customers rated 2 or 3 on service score.
- 60% of the churned customers are from the Regular Plus segment.
- Surprisingly almost 80% of the churned customers rated the customer care agents average or above average.
- Almost 60% of churned users used coupons to pay bills only once or never.



Data Cleaning



This dataset needs some cleaning before proceeding for EDA.

- First, we have special characters as a value of a cell in a lot of columns. We need to replace them.
- We are replacing them with the np.nan making it a null value.
- These missing values will be imputed later by using appropriate methods.

After executing the 1st step of the data cleaning steps. The numbers of null values increased as we replaced the special characters with the null values.
Irregular data like “F”, “Female” in gender column, “Regular +” and “Regular Plus” in Segment column are also treated.

Missing value treatment.

After replacing the special characters and irregular data, time to impute the missing values. If we check the missing values in every variable/column, almost every column has some missing values.

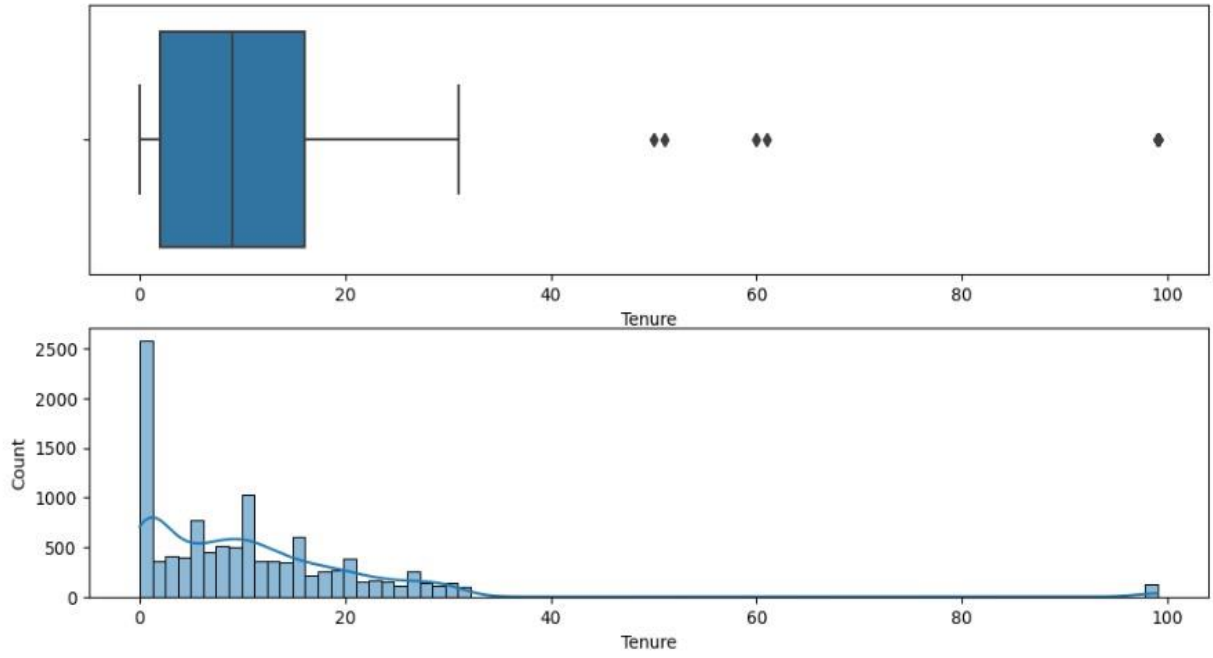
AccountID	0
Churn	0
Tenure	218
City_Tier	112
CC_Contacted_LY	102
Payment	109
Gender	108
Service_Score	98
Account_user_count	444
account_segment	97
CC_Agent_Score	116
Marital_Status	212
rev_per_month	791
Complain_ly	357
rev_growth_yoy	3
coupon_used_for_payment	3
Day_Since_CC_connect	358
cashback	473
Login_device	760

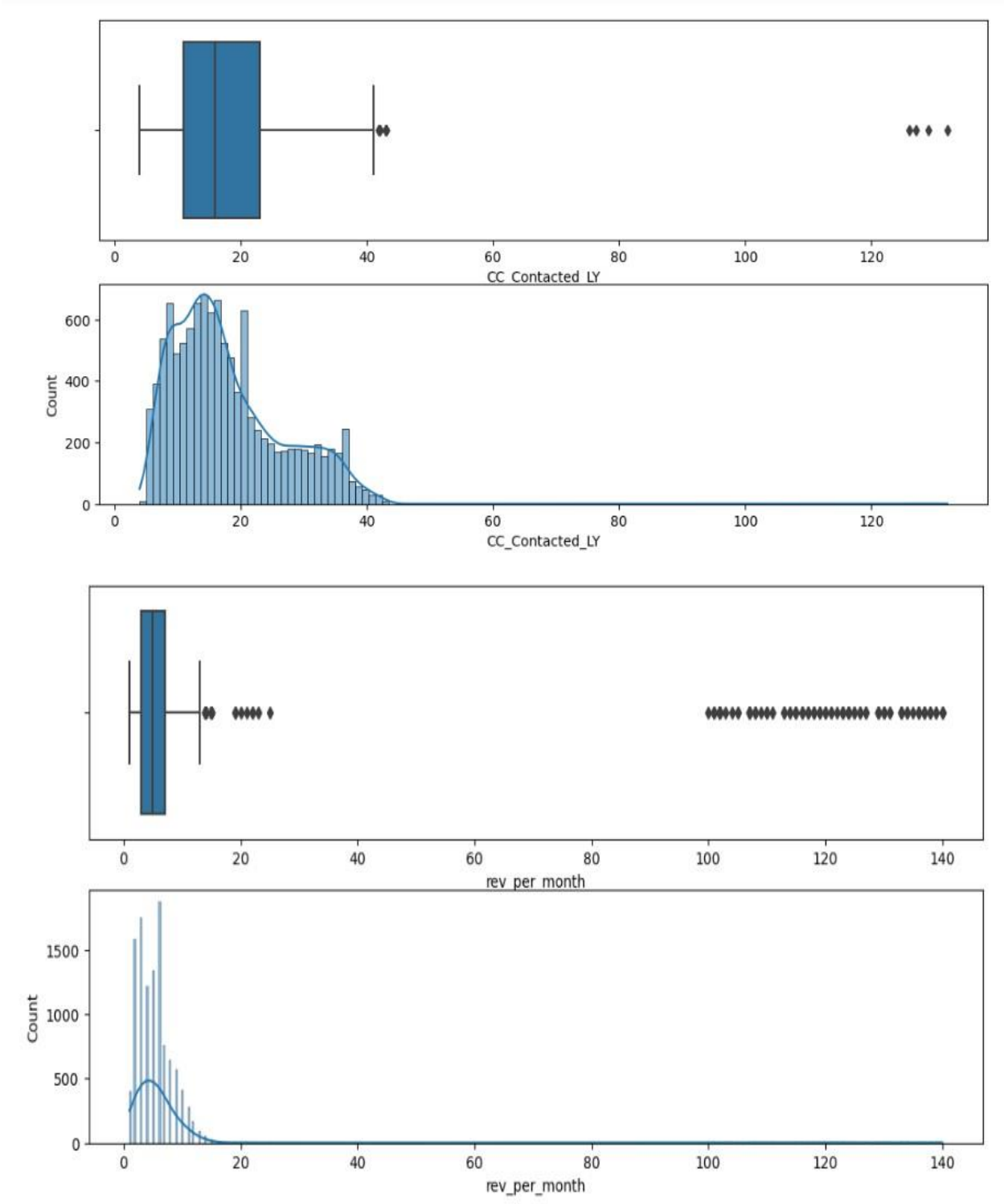
For imputation of null values in columns which have numerical data type, we are using KNN Imputer. And for those which have Object as data type we have used Mode of the variable.

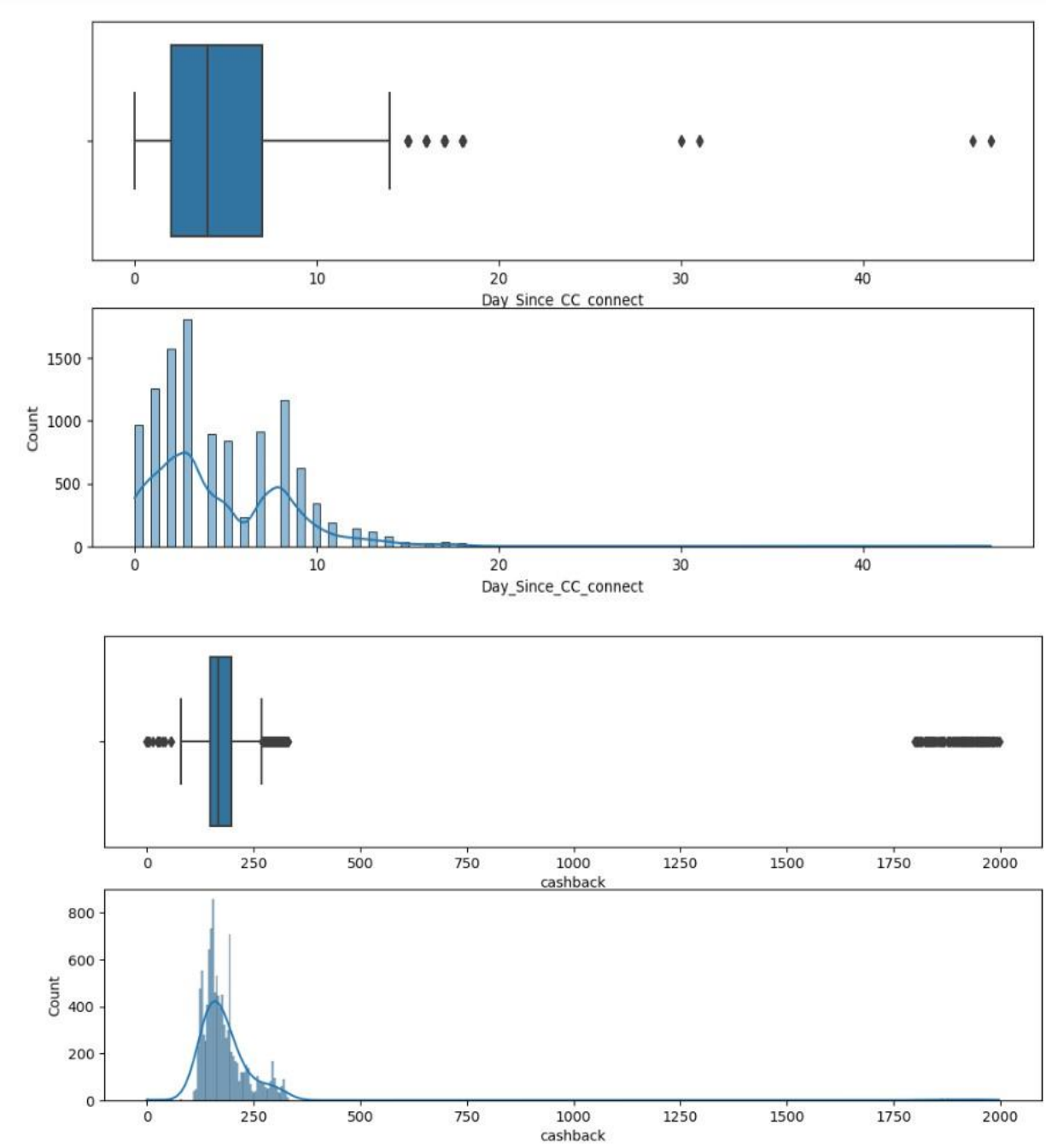
```
missing value in Tenure imputed with KNNImputer
missing value in City_Tier imputed with KNNImputer
missing value in CC_Contacted_LY imputed with KNNImputer
missing value in Service_Score imputed with KNNImputer
missing value in Account_user_count imputed with KNNImputer
missing value in CC_Agent_Score imputed with KNNImputer
missing value in rev_per_month imputed with KNNImputer
missing value in Complain_ly imputed with KNNImputer
missing value in rev_growth_yoy imputed with KNNImputer
missing value in coupon_used_for_payment imputed with KNNImputer
missing value in Day_Since_CC_connect imputed with KNNImputer
missing value in cashback imputed with KNNImputer

missing value in Payment with mode of Payment
Debit Card
missing value in Gender with mode of Gender
Male
missing value in account_segment with mode of account_segment
Regular Plus
missing value in Marital_Status with mode of Marital_Status
Married
missing value in Login_device with mode of Login_device
Mobile
```

Outlier treatment

We have columns having outliers.

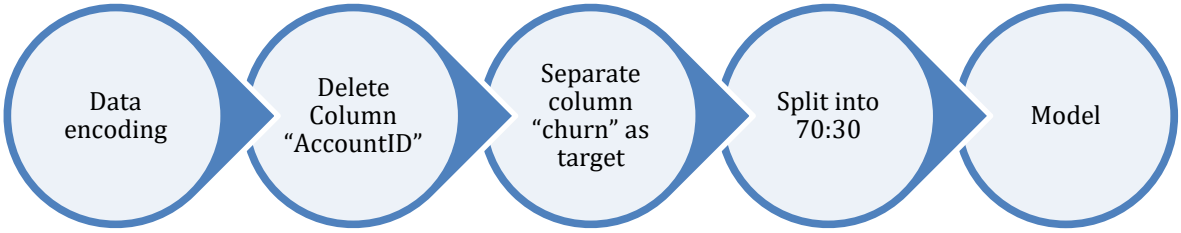



0
20
40
60
80
100
120
CC_Contacted_LY
600
400
200
0
Count
0
20
40
60
80
100
120
CC_Contacted_LY
0
20
40
60
80
100
120
140
rev_per_month
1500
1000
500
0
Count
0
20
40
60
80
100
120
140
rev_per_month



To treat outliers in those columns we have used the IQR and clipped the outlying values into the range. For this purpose, a function is defined to calculate the IQR upper bound and lower bound. The function returns the treated column.

Pre-Processing of Machine Learning



Data Encoding

Before starting to build various models to predict the churned users, we must process the data to make it usable for the model. Most of the learning models calculate the similarity and dissimilarity between data points based on some distance matrix like Euclidean distance, thus supporting only continuous data.

The categories of the data are converted into numerical codes with the help of categorical encoding method. One-hot encoding is another way to execute similar tasks.

Train-Test Split

The data containing the "AccountID" column is of no use because it is just used for indexing. And the "Churn" column is our target variable.

This is our x variables:

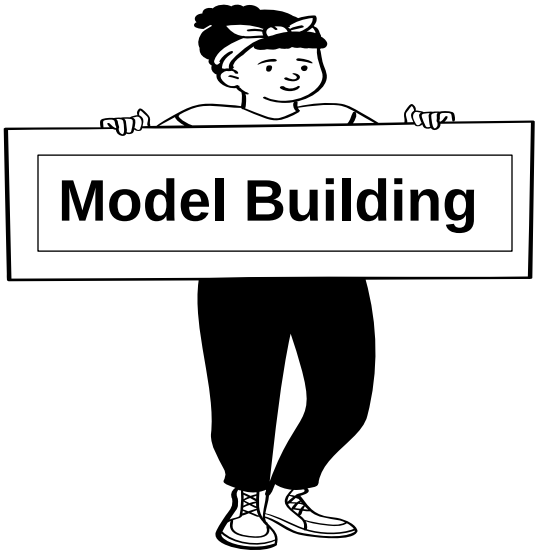
['Tenure', 'City_Tier', 'CC_Contacted_LY', 'Payment', 'Gender','Service_Score', 'Account_user_count', 'account_segment', 'CC_Agent_Score', 'Marital_Status', 'rev_per_month', 'Complain_ly', 'rev_growth_yoy', 'coupon_used_for_payment', 'Day_Since_CC_connect', 'cashback ', 'Login_device']

The Y variable is "Churn".

By using train_test_split () function, the data is splatted into train and test in a 70:30 ratio.

After splitting the data when we checked for the value counts of both classes, it seems that the data is unbalanced.
For both training and test data
Class 0: 83%
Class 1: 17%

Since our goal is to predict the class 1 (that is the churned customers), this unbalanced data may not get us the desired result. And to counter this we are going to use SMOTE to generate the synthetic data to balance it.

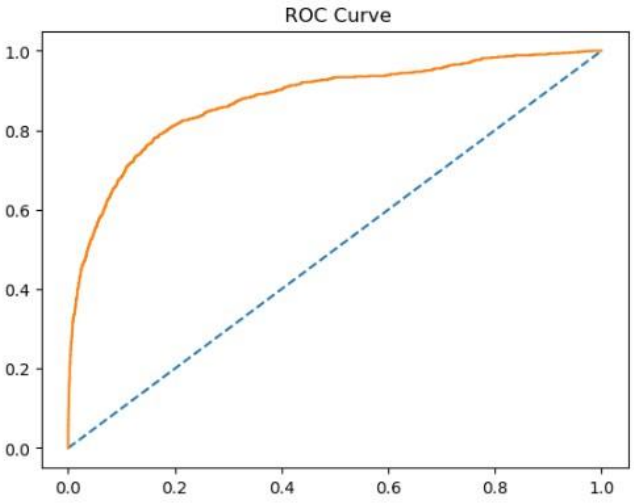


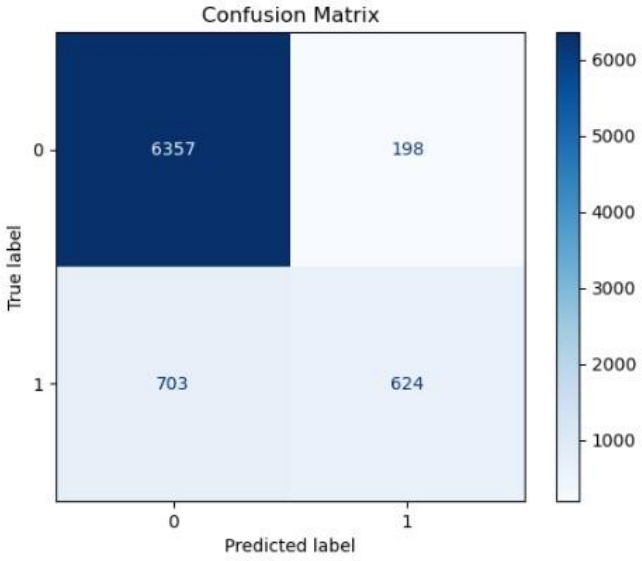
Model 1 : Logistic Regression

```
LogisticRegression
LogisticRegression(max_iter=10000, n_jobs=2, solver='newton-cg', verbose=True)
```

Model evaluation on train data.

Model score (overall accuracy): 0.885.
Area under the curve (AUC): 0.876





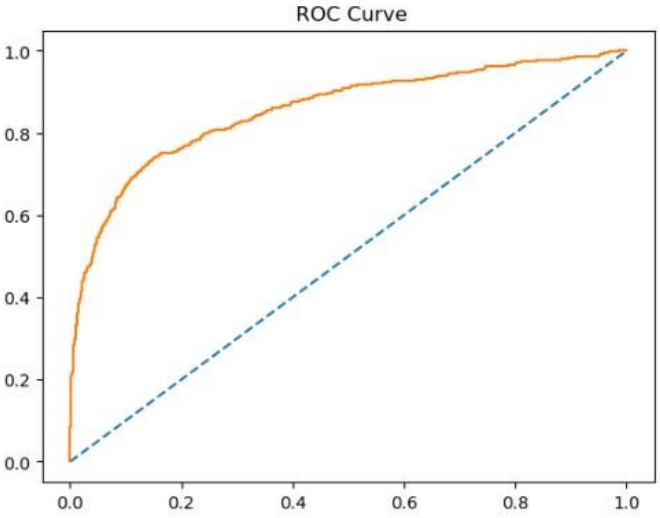
Classification Report

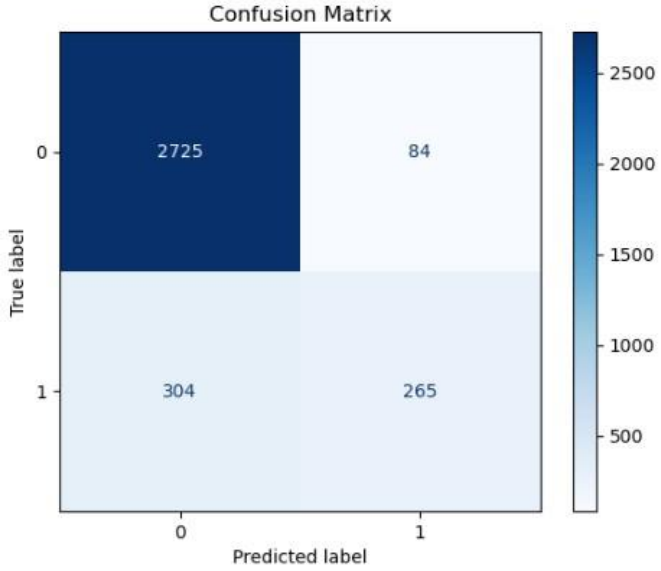
	precision	recall	f1-score	support
0	0.90	0.97	0.93	6555
1	0.76	0.47	0.58	1327
accuracy			0.89	7882
macro avg	0.83	0.72	0.76	7882
weighted avg	0.88	0.89	0.87	7882

False positive rate = 0.030205949656750573
True positive rate = 0.4702336096458176

Model evaluation on test data.

Model score (overall accuracy): 0.8851391355831854.
Area under the curve (AUC): 0.856





Classification Report

	precision	recall	f1-score	support
0	0.90	0.97	0.93	2809
1	0.76	0.47	0.58	569
accuracy			0.89	3378
macro avg	0.83	0.72	0.76	3378
weighted avg	0.88	0.89	0.87	3378

False positive rate = 0.029903880384478464
True positive rate = 0.46572934973637964

Interpretation of the model

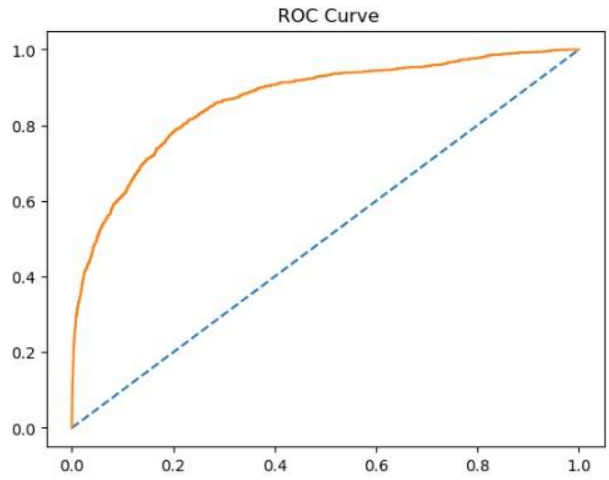
- The overall accuracy of the model is 89% for both train and test data.
- The AUC value for train and test data is 0.867 and 0.856 resp. This indicates that the model has good discriminatory power which means the model is effective at distinguishing between the positive and negative classes, as demonstrated by the ROC curve.
- For class 1, the precision is 0.76, indicating that 76% of instances predicted as class 1 are actually class 1.
- For class 1, the recall is 0.47, indicating that 47% of actual class 1 instances were correctly classified as class 1.
- Recall is also called true positive rate (TPR). The proportion of instances belonging to the positive class that were correctly classified as positive.
- FPR is the proportion of instances belonging to the negative class that were incorrectly classified as positive. It is around 3% for both train and test data.

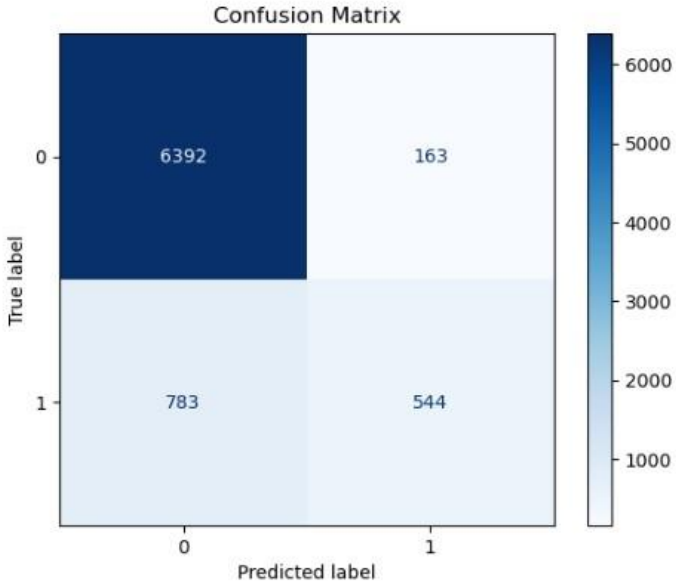
Model 2 : Linear Discriminant Analysis

```
▾ LinearDiscriminantAnalysis
LinearDiscriminantAnalysis()
```

Model evaluation on train data.

Model score (overall accuracy): 0.8799797005836082.
Area under the curve (AUC): 0.865





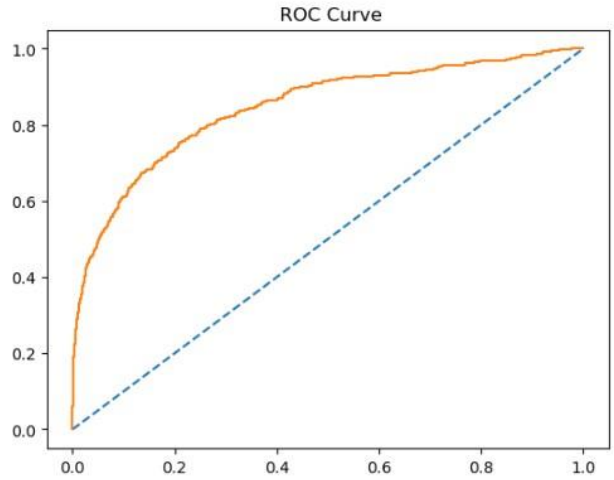
Classification Report

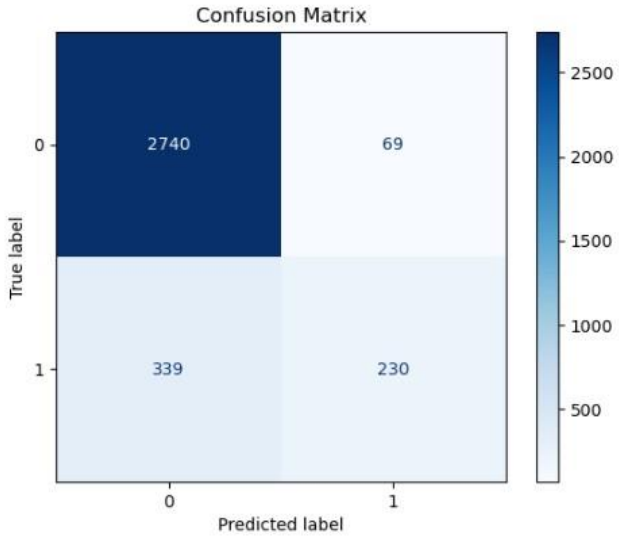
	precision	recall	f1-score	support
0	0.89	0.98	0.93	6555
1	0.77	0.41	0.53	1327
accuracy			0.88	7882
macro avg	0.83	0.69	0.73	7882
weighted avg	0.87	0.88	0.86	7882

False positive rate = 0.02486651411136537
True positive rate = 0.40994724943481536

Model evaluation on test data.

Model score (overall accuracy): 0.8792184724689165.
Area under the curve (AUC): 0.846





Classification Report

	precision	recall	f1-score	support
0	0.89	0.98	0.93	2809
1	0.77	0.40	0.53	569
accuracy			0.88	3378
macro avg	0.83	0.69	0.73	3378
weighted avg	0.87	0.88	0.86	3378

False positive rate = 0.024563901744393022
True positive rate = 0.40421792618629176

Interpretation of the model

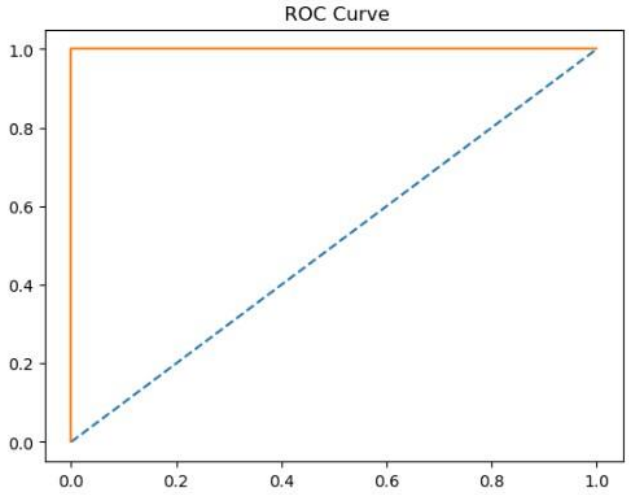
- For both classes (0 and 1), the precision, recall, and F1-score values are quite similar between the training and test datasets which suggests consistent performance across both datasets.
- For both training and test datasets, the model's performance is lower for class 1 (positive class), with lower precision, recall, and F1-score.
- The accuracy is relatively high for both datasets, indicating that the model performs well overall.
- While the model performs well overall, there is room for improvement in correctly identifying positive instances.

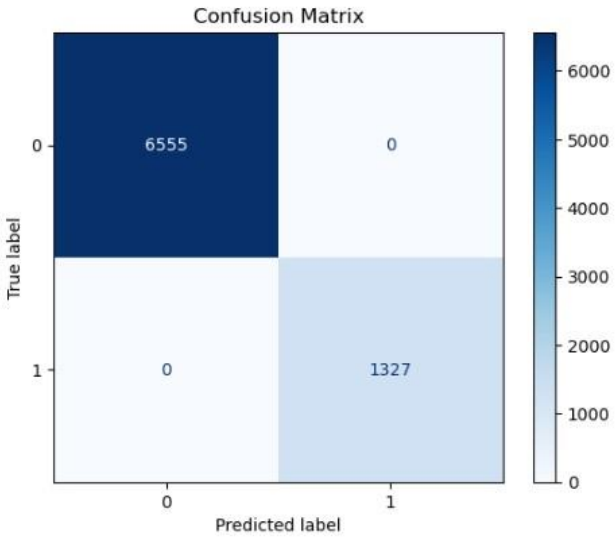
Model 3: K-Nearest Neighbours

```
KNeighborsClassifier(weights='distance')
```

Model evaluation on train data.

Model score (overall accuracy): 1.0
Area under the curve (AUC): 1.000





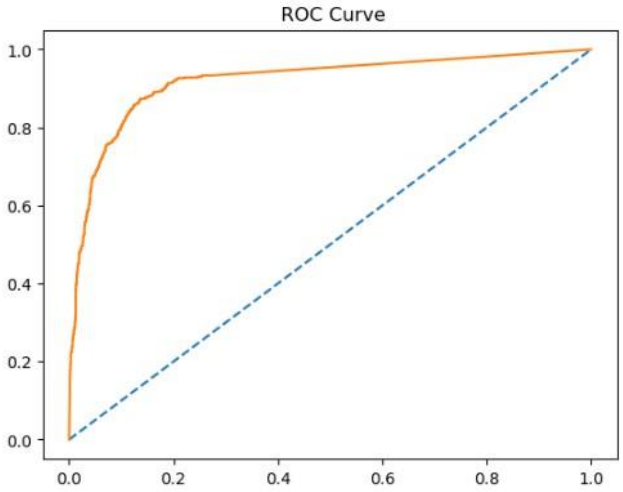
Classification Report

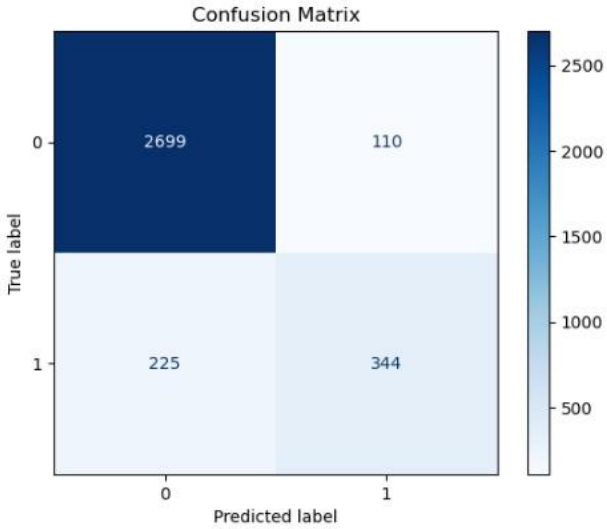
	precision	recall	f1-score	support
0	1.00	1.00	1.00	6555
1	1.00	1.00	1.00	1327
accuracy			1.00	7882
macro avg	1.00	1.00	1.00	7882
weighted avg	1.00	1.00	1.00	7882

False positive rate = 0.0
True positive rate = 1.0

Model evaluation on test data.

Model score (overall accuracy): 0.9008288928359977.
Area under the curve (AUC): 0.919





Classification Report

	precision	recall	f1-score	support
0	0.92	0.96	0.94	2809
1	0.76	0.60	0.67	569
accuracy			0.90	3378
macro avg	0.84	0.78	0.81	3378
weighted avg	0.90	0.90	0.90	3378

False positive rate = 0.039159843360626556
True positive rate = 0.6045694200351494

Interpretation of the model

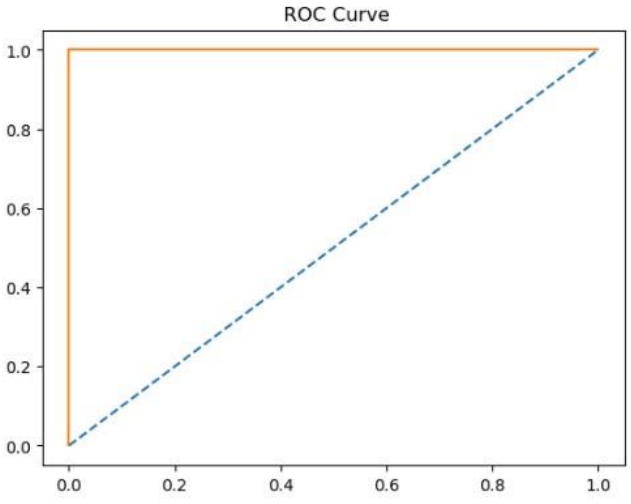
- AUC value for train = 1.0 and value for test = 0.919 indicates that the model has excellent discriminatory power
- It suggests that the model is highly effective at distinguishing between the positive and negative classes, as demonstrated by the ROC curves.
- On test data for class 1 (positive class), precision is 0.76, indicating that 76% of instances predicted as class 1 are actually class 1.
- And recall is 0.60, indicating that 60% of actual class 1 instances were correctly classified as class 1.
- The overall accuracy of the model is 0.90 or 90%, indicating that 90% of instances were correctly classified.
- FPR is 0.039, indicating that 3.9% of actual negative instances were incorrectly classified as positive.
- For class 1, while precision is moderate, recall is relatively lower.

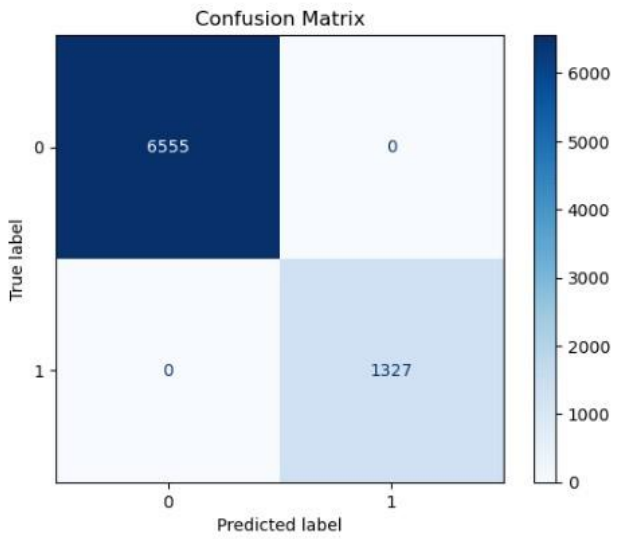
Model 4 : Decision Tree Classifier

```
DecisionTreeClassifier
DecisionTreeClassifier(random_state=1)
```

Model evaluation on train data.

Model score (overall accuracy): 1.0.
Area under the curve (AUC): 1.000





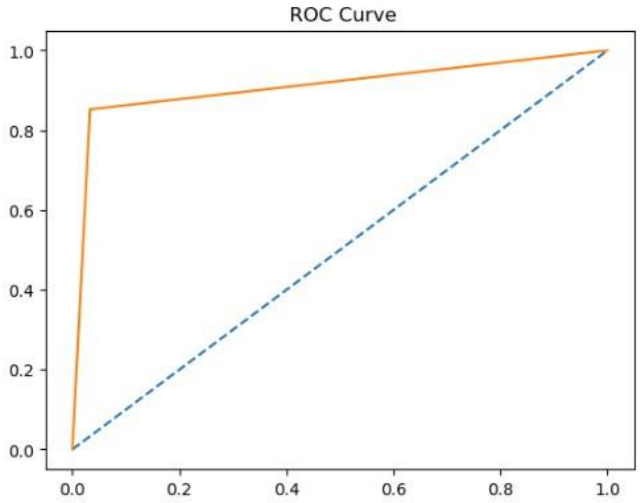
Classification Report

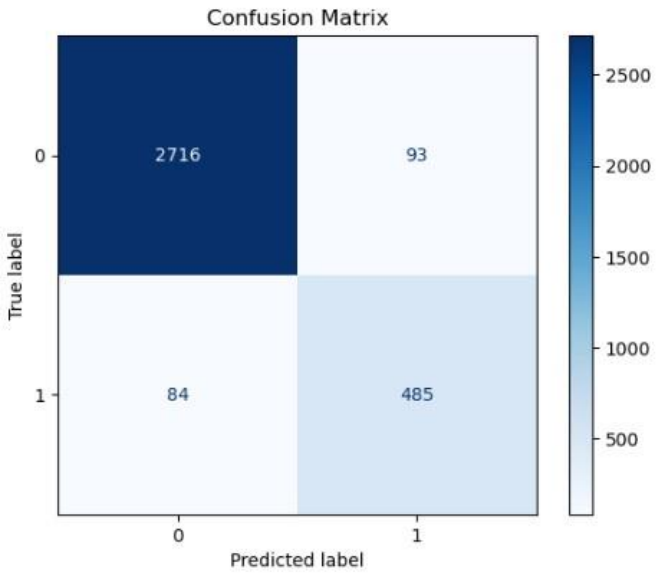
	precision	recall	f1-score	support
0	1.00	1.00	1.00	6555
1	1.00	1.00	1.00	1327
accuracy			1.00	7882
macro avg	1.00	1.00	1.00	7882
weighted avg	1.00	1.00	1.00	7882

False positive rate = 0.0
True positive rate = 1.0

Model evaluation on test data.

Model score (overall accuracy): 0.9476021314387212.
Area under the curve (AUC): 0.910





Classification Report

	precision	recall	f1-score	support
0	0.97	0.97	0.97	2809
1	0.84	0.85	0.85	569
accuracy			0.95	3378
macro avg	0.90	0.91	0.91	3378
weighted avg	0.95	0.95	0.95	3378

False positive rate = 0.03310786756852972
True positive rate = 0.8523725834797891

Interpretation of the model

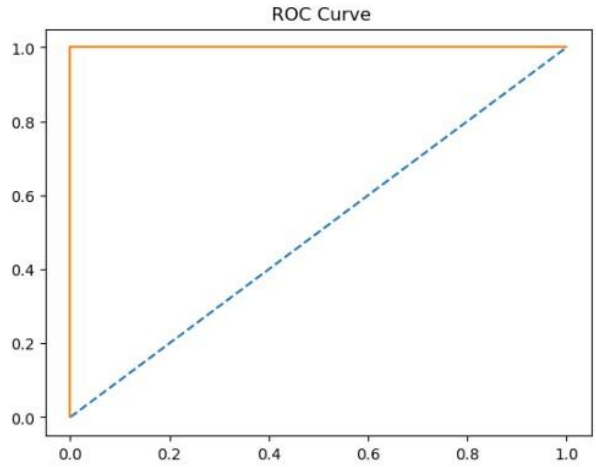
- The overall accuracy of the model is 0.95 or 95%, indicating that 95% of instances were correctly classified.
- An AUC of 0.910 suggests that the model has excellent discriminatory power.
- On test data, precision value on test data is 0.84, indicating that 84% of instances predicted as class 1 are actually class 1.
- And recall is 0.85, indicating that 85% of actual class 1 instances were correctly classified as class 1.
- FPR is 0.033, indicating that 3.3% of actual negative instances were incorrectly classified as positive.
- The model demonstrates strong performance with high precision, recall, and F1-score values for both classes.
- The AUC value further confirms the model's effectiveness in discriminating between positive and negative instances.

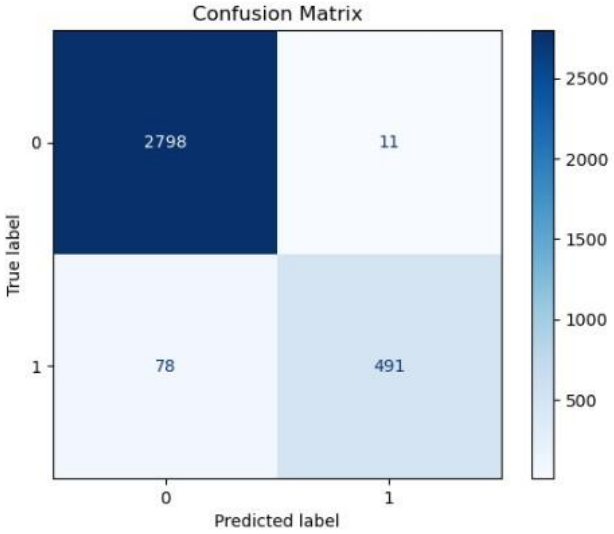
Model 5: Random Forest Classifier

```
RandomForestClassifier
RandomForestClassifier(random_state=1)
```

Model evaluation on train data.

Model score (overall accuracy): 1.0.
Area under the curve (AUC): 1.000





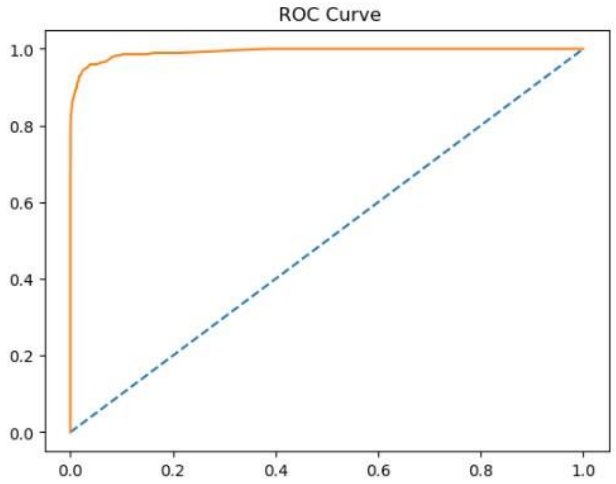
Classification Report

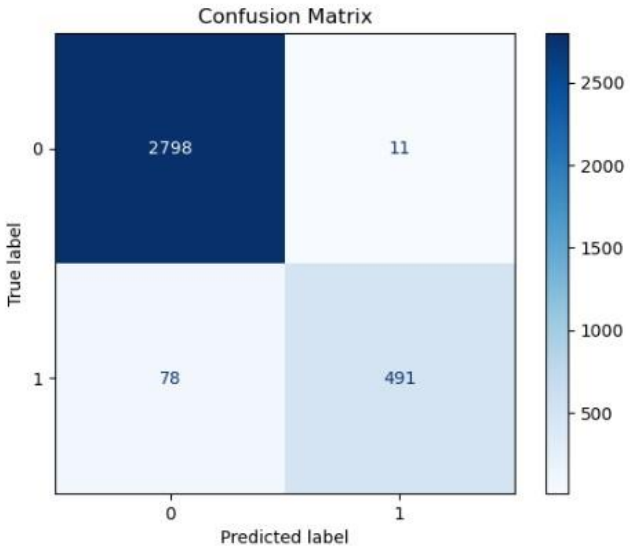
	precision	recall	f1-score	support
0	0.97	1.00	0.98	2809
1	0.98	0.86	0.92	569
accuracy			0.97	3378
macro avg	0.98	0.93	0.95	3378
weighted avg	0.97	0.97	0.97	3378

False positive rate = 0.0
True positive rate = 1.0

Model evaluation on test data.

Model score (overall accuracy): 0.9736530491415039.
Area under the curve (AUC): 0.993





Classification Report

	precision	recall	f1-score	support
0	0.97	1.00	0.98	2809
1	0.98	0.86	0.92	569
accuracy			0.97	3378
macro avg	0.98	0.93	0.95	3378
weighted avg	0.97	0.97	0.97	3378

False positive rate = 0.003915984336062656
True positive rate = 0.8629173989455184

Interpretation of the model

- The AUC value on test data is 0.993, indicating the model's exceptional performance in distinguishing between the positive and negative classes using the Receiver Operating Characteristic (ROC) curve.
- On test data, for class 1, precision is 0.98, indicating that 98% of instances predicted as class 1 are actually class 1.
- TPR or recall is 0.863, indicating that 86.3% of actual positive instances were correctly classified as positive.
- For class 1, the value of F1 score is 0.92, which suggests the exceptional performance balanced between the precision and recall.
- FPR is 0.004, indicating that 0.4% of actual negative instances were incorrectly classified as positive.

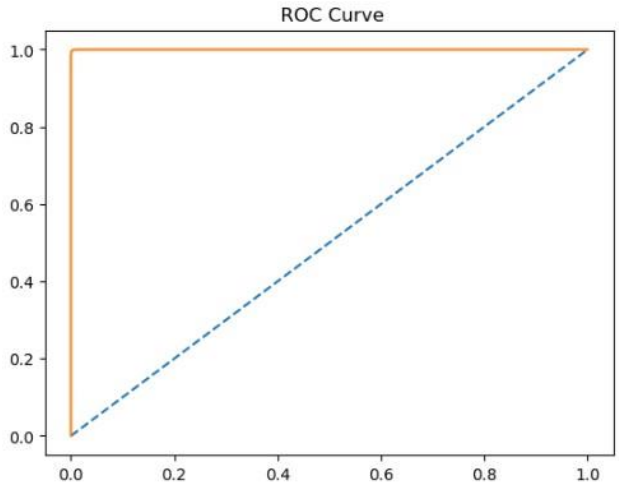
In summary, the model demonstrates exceptional performance with high precision, recall, and F1-score values for both classes. The AUC value further confirms the model's effectiveness in discriminating between positive and negative instances. Overall, the model appears to be highly accurate and reliable for the given classification task.

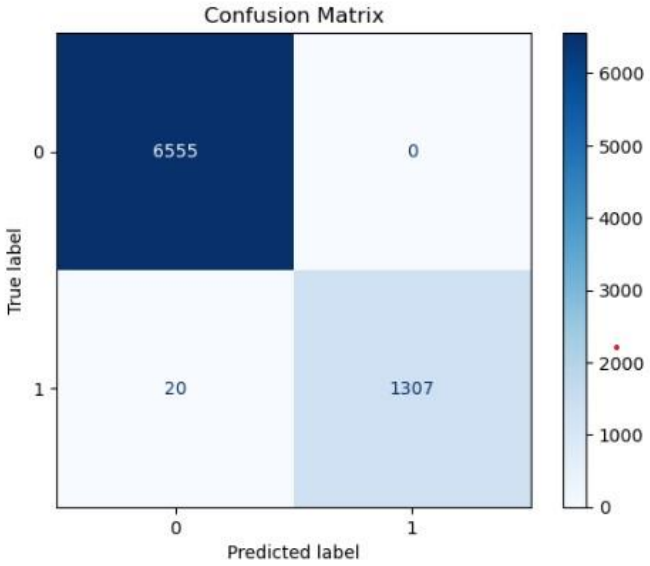
Model 6: Bagging Classifier

```
BaggingClassifier
BaggingClassifier(random_state=1)
```

Model Evaluation on train data.

Model score (overall accuracy): 0.9974625729510277.
Area under the curve (AUC): 1.000





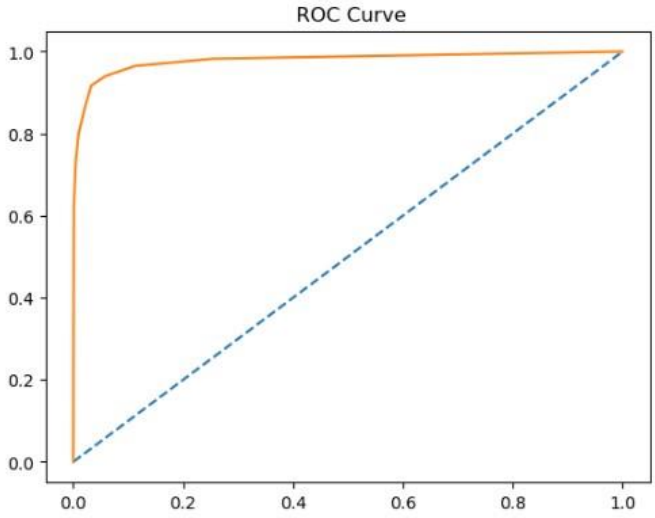
Classification Report

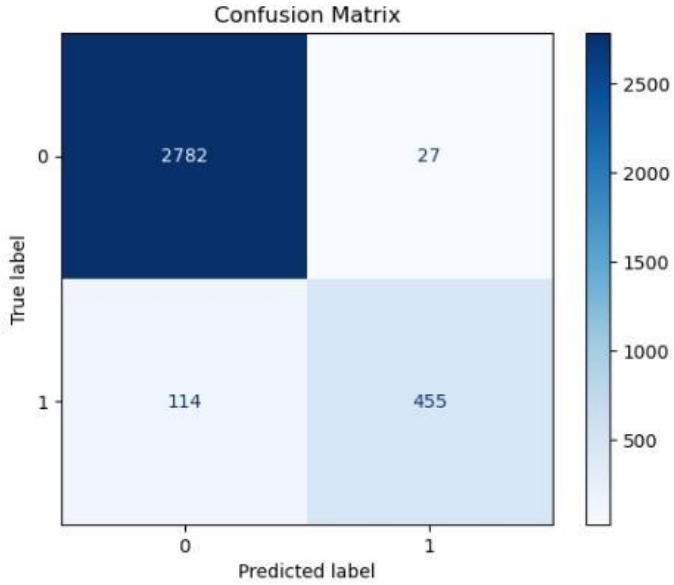
	precision	recall	f1-score	support
0	1.00	1.00	1.00	6555
1	1.00	0.98	0.99	1327
accuracy			1.00	7882
macro avg	1.00	0.99	1.00	7882
weighted avg	1.00	1.00	1.00	7882

False positive rate = 0.0
True positive rate = 0.9849284099472494

Model evaluation on test data.

Model score (overall accuracy): 0.9582593250444049.
Area under the curve (AUC): 0.979





Classification Report

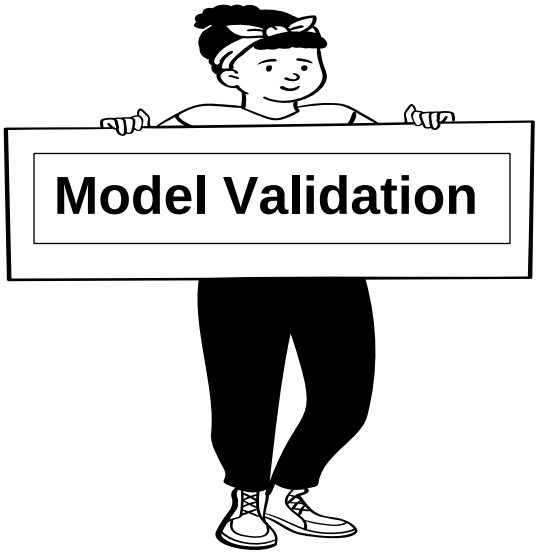
	precision	recall	f1-score	support
0	0.96	0.99	0.98	2809
1	0.94	0.80	0.87	569
accuracy			0.96	3378
macro avg	0.95	0.90	0.92	3378
weighted avg	0.96	0.96	0.96	3378

False positive rate = 0.009611961552153792
True positive rate = 0.7996485061511424

Interpretation of the model

- An AUC of 0.979 suggests that the model has outstanding discriminatory power.
- For class 1 (positive class), precision is 0.94, indicating that 94% of instances predicted as class 1 are actually class 1.
- For class 1, recall is 0.80, indicating that 80% of actual class 1 instances were correctly classified as class 1.
- For class 0, the F1-score is 0.98, and for class 1, it is 0.87.
- The F1-score is the harmonic mean of precision and recall, providing a balanced measure of the model's performance.
- The overall accuracy of the model is 0.96 or 96%, indicating that 96% of instances were correctly classified.
- FPR is 0.010, indicating that 1.0% of actual negative instances were incorrectly classified as positive.

The model demonstrates excellent performance with high precision, recall, and F1-score values for both classes. The AUC value further confirms the model's effectiveness in discriminating between positive and negative instances. Overall, the model appears to be highly accurate and reliable for the given classification task.



Model Comparison:

We have built 6 classification models. Trained and evaluated them on the same training data and test data. To see the comparison between these all models, here is the table.

	Training data (70%)					Test data (30%)				
Model	**Accuracy**	**AUC**	**Precision**	**Recall**	F1	**Accuracy**	**AUC**	**Precision**	**Recall**	**F1**
Logistic Regression	76.18	85.3	40	80	53	88.51	85.6	76	41	58
Logistic Regression (Balanced data)	78.63	86.1	77	82	79	77.8	84.9	76	81	79
LDA	87.99	86.5	77	41	53	87.92	84.6	77	40	53
KNN	100	100	100	100	100	90.08	91.9	76	60	67
Decision Tree	100	100	100	100	100	94.76	91	84	85	85
Random Forest	100	100	100	100	100	97.36	99.3	98	86	92
Bagging Classifier (default parameters)	99.74	100	100	98	99	95.82	97.9	94	80	87

Random forest classifier performs best on the test data. The model demonstrates exceptional performance with high precision, recall, and F1-score values for both classes. The AUC value further confirms the model's effectiveness in discriminating between positive and negative instances. Overall, the model appears to be highly accurate and reliable.

Model Validation and Tuning:

To check the reliability of the model we have some validation techniques.

K-Fold Cross-Validation (CV) is a popular technique used in machine learning for model evaluation and validation. It is particularly useful for assessing the performance of a model and estimating its generalization ability on unseen data.

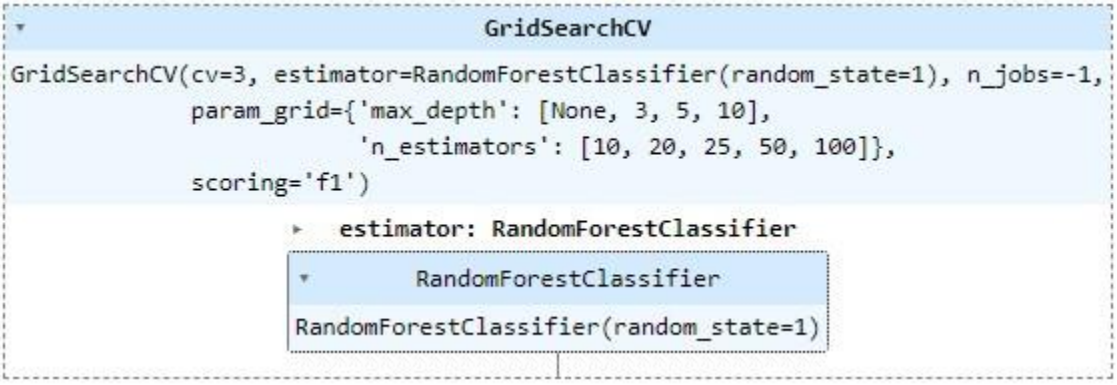
The main idea behind K-Fold CV is to divide the dataset into K equal-sized folds, or subsets, and iteratively train and test the model K times, each time using a different fold as the test set and the remaining folds as the training set.

Here, the data is divided into 20 subsets. And calculated the accuracy of each subset.

```
Avg Accuracy and Standard Deviation : 98.259% (0.563%)
```

We got the average accuracy more than the model itself. This cross validation provided the evidence that the model is performing exceptionally well in this classification problem.

GridSearchCV, is a technique used in machine learning to find the optimal hyperparameters for a given model. It systematically searches through a predefined grid of hyperparameters, evaluating each combination using cross-validation to determine the best-performing set of hyperparameters.



After applying above defined GridSearch for random forest classifier:

Model
RandomForestClassifier(random_state=1)
Best Parameters
{'max_depth': None, 'n_estimators': 100}

On testing this model on test data with these parameters the results are same:

Model score (overall accuracy): 1.00.
Area under the curve (AUC): 1.00
False positive rate = 0.00
True positive rate = 1.00

Classification Report

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2809
1	1.00	1.00	1.00	569
accuracy			1.00	3378
macro avg	1.00	1.00	1.00	3378
weighted avg	1.00	1.00	1.00	3378

False positive rate = 0.0
True positive rate = 1.0

Interpretation (Best model: Random Forest Classifier):

- The Random Forest Classifier model achieved an overall accuracy of 97.37%, indicating that it correctly classified approximately 97.37% of the instances in the dataset.
- The Area Under the Curve (AUC) score of 0.993 suggests that the model has excellent discriminatory power and performs significantly better than random guessing when distinguishing between the positive and negative classes.
- Additionally, the false positive rate (FPR) is low at 0.39%, indicating that only a small percentage of actual negative instances were incorrectly classified as positive (class 1).
- The true positive rate (TPR) is high at 86.29%, indicating that a large percentage of actual positive instances were correctly classified as positive (class 1).

In terms of precision, recall, and F1-score, the model performed exceptionally well for both classes. For class 0 (negative class), precision, recall, and F1-score are all above 0.97, indicating high accuracy in identifying negative instances. For class 1 (positive class), precision is 0.98, indicating that 98% of instances predicted as class 1 are actually class 1. However, the recall for class 1 is slightly lower at 0.86, indicating that 86% of actual positive instances were correctly classified as class 1.

Insights and Recommendations

City Distribution	Churn Patterns	Ratings	Segment	features
• High concentration in urban areas	• Urban areas experience significant churn	• 30% Rated CC below average • Service ratings are below avg 80% of the time	• Regular Plus segment has high Tendency to churn	• tenure • revenue per month • Cashback • Variables related to customer satisfaction

Enhance Service Quality and Customer Support:

- Monitor and evaluate
- Improve the performance and quality

Segment-Specific Retention Strategies:

- Different needs and preferences

Urban Market:

- Increasing customer engagement in urban areas
- Target customers with personalized offers and incentives

Customer Engagement and Loyalty Programs:

- Prioritize initiatives aimed at enhancing customer engagement and loyalty.
- Design loyalty programs and special offers

1. **Enhance Service Quality and Customer Support:**

   - Invest in training programs to improve the performance and quality of customer support agents, addressing the concerns raised by customers who rated support agents below average.
   - Implement measures to monitor and evaluate customer interactions with support agents to ensure consistent and high-quality service delivery.

2. **Segment-Specific Retention Strategies:**

   - Develop targeted retention strategies tailored to the needs and preferences of different customer segments, particularly focusing on the Regular Plus segment, which exhibits a higher propensity for churn.

3. **Urban Market Expansion and Customer Engagement:**

   - Explore opportunities for expanding operations and increasing customer engagement in tier 1 cities to mitigate churn rates in urban areas.
   - Implement marketing campaigns and promotional activities to attract and retain customers in tier 1 cities, leveraging insights into payment preferences and service satisfaction.

4. **Focus on Customer Engagement and Loyalty Programs:**

   - Given the high importance of features such as tenure and revenue per month in churn prediction, prioritize initiatives aimed at enhancing customer engagement and loyalty.
   - Implement loyalty programs, special offers, and rewards for long-tenured customers to incentivize their continued patronage and reduce the likelihood of churn.

5. **Optimize Marketing and Promotional Efforts:**

   - Utilize insights into churn patterns and feature importance in churn prediction to refine marketing and promotional strategies.
   - Target customers with personalized offers and incentives based on their tenure, spending behaviour, and engagement history to increase loyalty and reduce churn.

By implementing these recommendations, the company can strengthen its customer retention efforts, improve service quality, and enhance overall customer satisfaction, leading to sustainable business growth and profitability.