CAPSTONE PROJECT CUSTOMER CHURN

BY – *Harsh Patel*

19th January 2025

{**NOTE:** I HAVE TRIED TO KEEP THE PAGES TO NOT MORE THAN 40 AS I HAD TO COMPRESS LOT OF INFORMATION INTO DIGESTIBLE CHUNKS.}

1. Introduction

A DTH 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 potential churners. In this company, account churn is a major issue because 1 account can have multiple customers. Hence by losing one account the company might be losing more than one customer.

The current project is aimed at developing a churn prediction model for this company and to provide business recommendations on a campaign focused on retaining customers. The campaign recommendation should be such that it does not entail a huge cost for retention of customers and should remain within a budget earmarked for this purpose.

Need for the project

A DTH provider's largest expense is acquiring a new customer. To recover the initial acquisition cost and ensure profitability, a customer must be retained for several years. Customer churn directly impacts the profitability of a DTH operator. Providers are under constant pressure to increase their customer base to maintain profitability, as they often have fixed fees from broadcasters/content providers regardless of the number of customers.
Acquiring a new customer can cost five times more than retaining an existing one. Increasing customer retention by 5% can boost profits by 25-95%. Therefore, it is crucial to protect the existing customer base while minimizing retention costs.

Project Objectives

1) This project aims to identify customers with a high propensity to churn so that targeted campaign offers can be provided to them.
2) This approach ensures better top-line and bottom-line revenue for the business. Additional insights from exploratory data analysis and the best-performing models will also be used to develop business recommendations.

2. Data Cleaning and Preprocessing:

The dataset contains 'Account' level data, which includes master data and account features such as gender, marital status, city tier, account user count of the primary account holder, whether the account is live or churned, and the segment the account belongs to. It also includes derived features such as tenure, likely derived from the account open date.

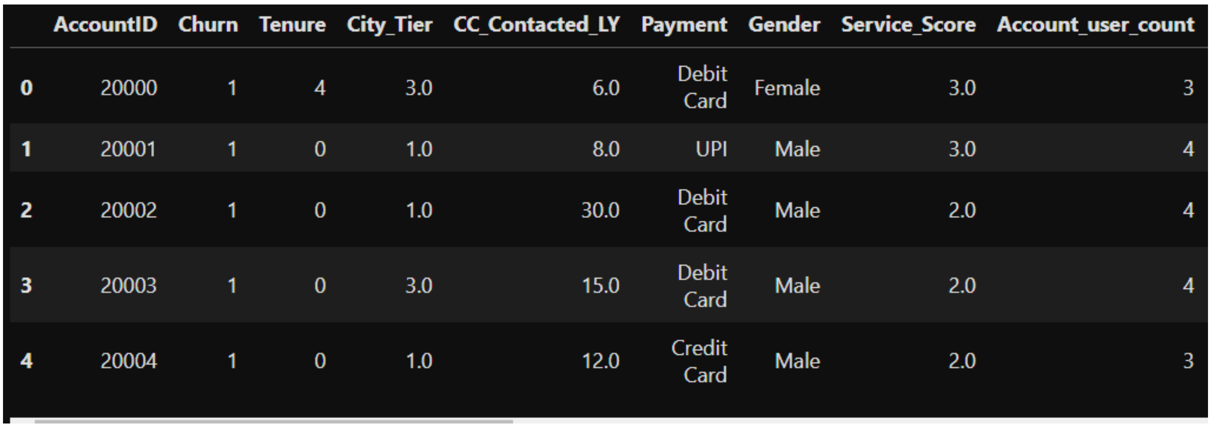
Additionally, the dataset contains information taken from or derived from transaction data and rolled up at the account level. Examples include:

- Number of days since none of the account holders contacted customer care
- Monthly average cashback for the last 12 months
- Number of complaints made last year
- Number of times customer care was contacted last year
- Revenue per month in the last 12 months
- Number of times customers used coupons to pay in the last 12 months
- Satisfaction score and customer service score

Most of the transaction data roll-ups at the account level cover the previous 12 months. However, the revenue growth percentage is calculated based on the last year compared to the previous year, implying that 24 months' worth of data has been used for this field.

As the dataset has been provided, the methodology used by the customer to extract the data is unknown, and the frequency of dataset extraction has not been specified.

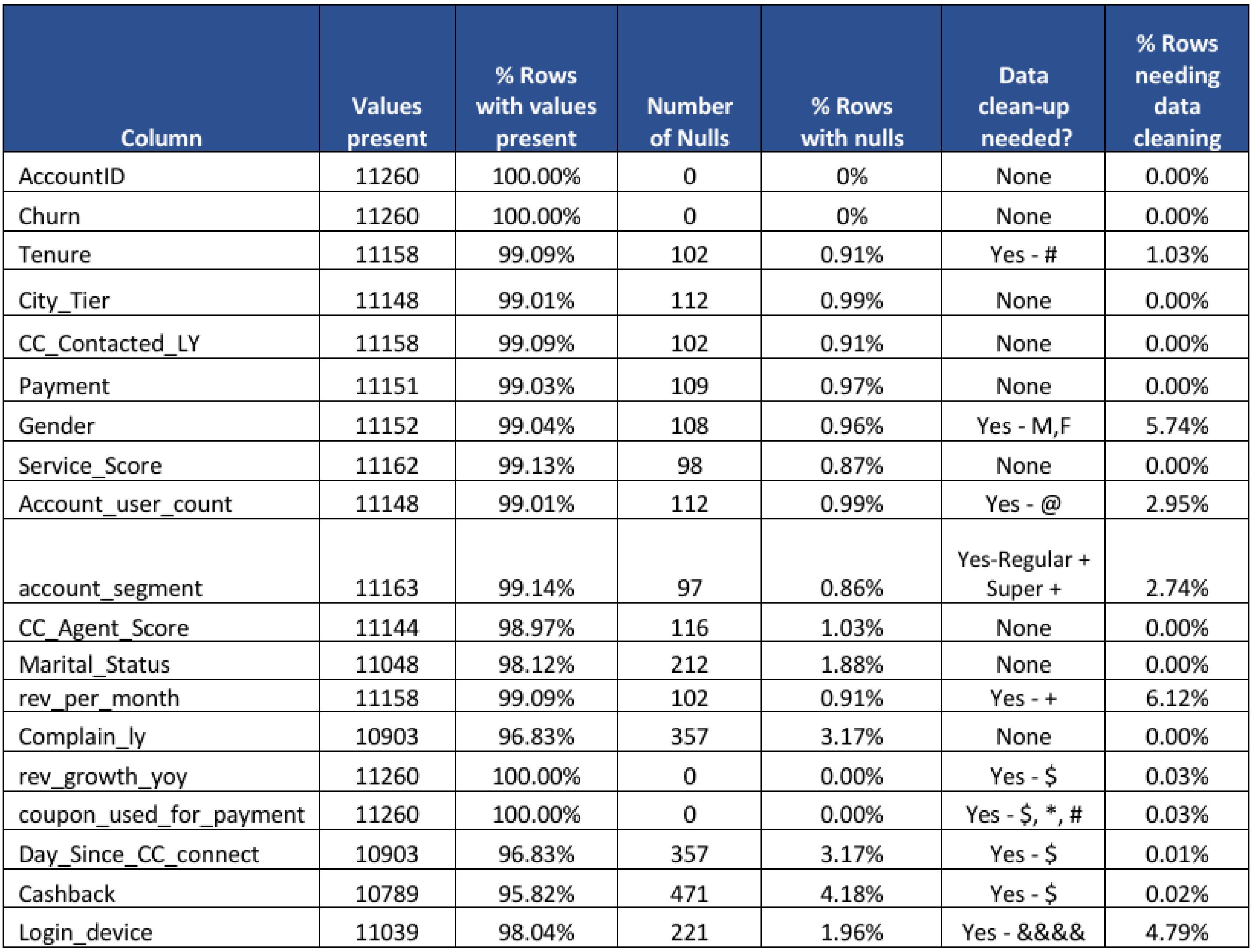
- The dataset has 11260 rows and 19 columns.
- There are 5 columns of float type, 2 columns of integer type and 12 columns of object type



	AccountID	Churn	Tenure	City_Tier	CC_Contacted_LY	Payment	Gender	Service_Score	Account_user_count
0	20000	1	4	3.0	6.0	Debit Card	Female	3.0	3
1	20001	1	0	1.0	8.0	UPI	Male	3.0	4
2	20002	1	0	1.0	30.0	Debit Card	Male	2.0	4
3	20003	1	0	3.0	15.0	Debit Card	Male	2.0	4
4	20004	1	0	1.0	12.0	Credit Card	Male	2.0	3

Table 1 First Five Rows of Dataset

The following table shows the number of rows containing nulls and special characters that require data cleaning. All special characters present in the data set were treated with nulls so that they can be imputed. Some columns such as Gender and account_segment contained multiple values to represent the same category for e.g., ‘M’, ‘Male’. Cleaning up of those values was also performed.

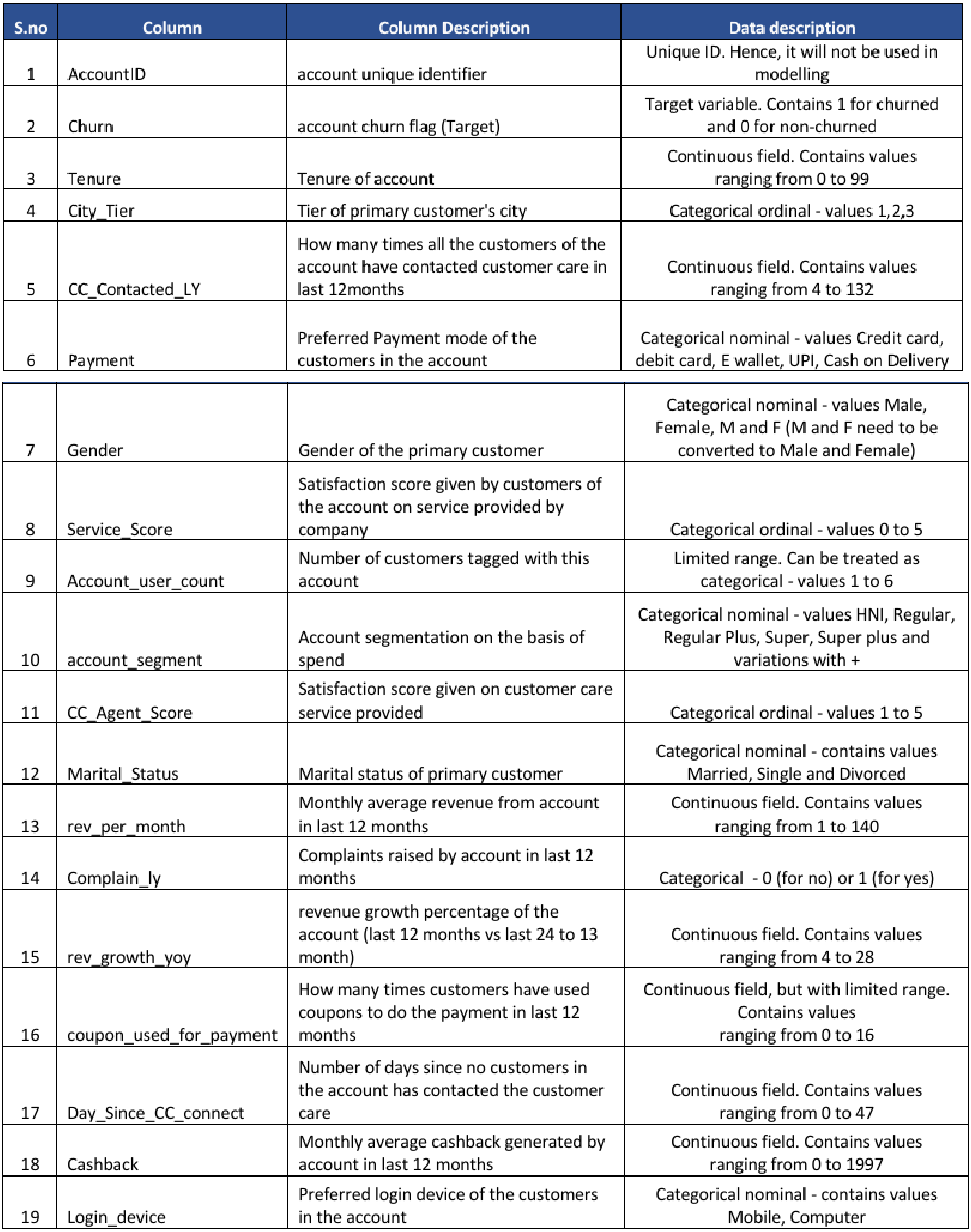


Column	Values present	% Rows with values present	Number of Nulls	% Rows with nulls	Data clean-up needed?	% Rows needing data cleaning
AccountID	11260	100.00%	0	0%	None	0.00%
Churn	11260	100.00%	0	0%	None	0.00%
Tenure	11158	99.09%	102	0.91%	Yes - #	1.03%
City_Tier	11148	99.01%	112	0.99%	None	0.00%
CC_Contacted_LY	11158	99.09%	102	0.91%	None	0.00%
Payment	11151	99.03%	109	0.97%	None	0.00%
Gender	11152	99.04%	108	0.96%	Yes - M,F	5.74%
Service_Score	11162	99.13%	98	0.87%	None	0.00%
Account_user_count	11148	99.01%	112	0.99%	Yes - @	2.95%
account_segment	11163	99.14%	97	0.86%	Yes-Regular + Super +	2.74%
CC_Agent_Score	11144	98.97%	116	1.03%	None	0.00%
Marital_Status	11048	98.12%	212	1.88%	None	0.00%
rev_per_month	11158	99.09%	102	0.91%	Yes - +	6.12%
Complain_ly	10903	96.83%	357	3.17%	None	0.00%
rev_growth_yoy	11260	100.00%	0	0.00%	Yes - $	0.03%
coupon_used_for_payment	11260	100.00%	0	0.00%	Yes - $, *, #	0.03%
Day_Since_CC_connect	10903	96.83%	357	3.17%	Yes - $	0.01%
Cashback	10789	95.82%	471	4.18%	Yes - $	0.02%
Login_device	11039	98.04%	221	1.96%	Yes - &&&&	4.79%

Table 2 Nulls and special characters in dataset

Understanding the attributes

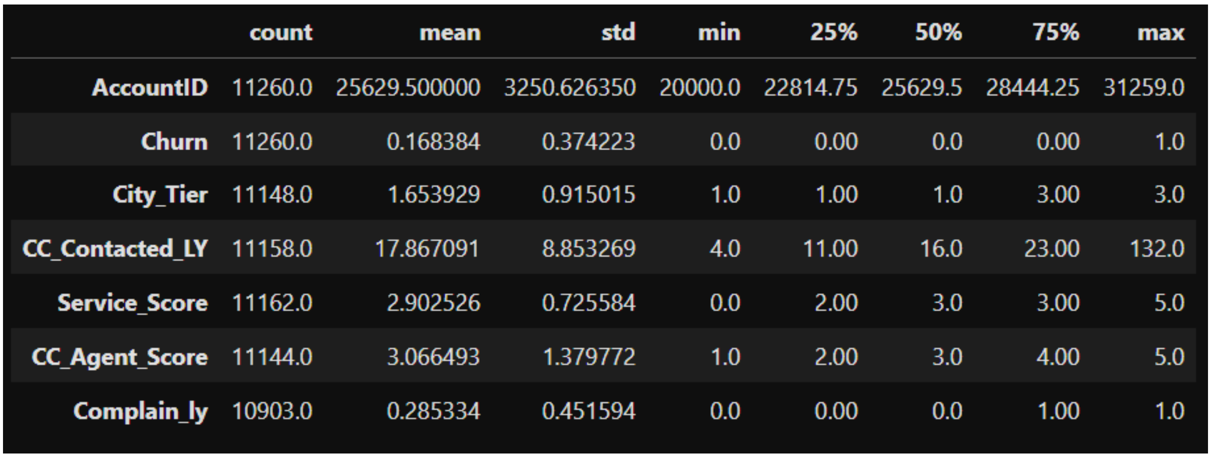
The following table shows the attribute names, their descriptions, and the types of values they contain. Although some variable names are slightly long, they do not have blanks or special characters. Therefore, it has been decided to keep the current column names as they are self-explanatory and easy to understand and interpret when seen in plots as part of univariate and bivariate analysis. The variable names will be changed later to shorten or standardize them when one-hot encoding is performed in a later section.



S.no	Column	Column Description	Data description
1	AccountID	account unique identifier	Unique ID. Hence, it will not be used in modelling
2	Churn	account churn flag (Target)	Target variable. Contains 1 for churned and 0 for non-churned
3	Tenure	Tenure of account	Continuous field. Contains values ranging from 0 to 99
4	City_Tier	Tier of primary customer's city	Categorical ordinal - values 1,2,3
5	CC_Contacted_LY	How many times all the customers of the account have contacted customer care in last 12months	Continuous field. Contains values ranging from 4 to 132
6	Payment	Preferred Payment mode of the customers in the account	Categorical nominal - values Credit card, debit card, E wallet, UPI, Cash on Delivery
7	Gender	Gender of the primary customer	Categorical nominal - values Male, Female, M and F (M and F need to be converted to Male and Female)
8	Service_Score	Satisfaction score given by customers of the account on service provided by company	Categorical ordinal - values 0 to 5
9	Account_user_count	Number of customers tagged with this account	Limited range. Can be treated as categorical - values 1 to 6
10	account_segment	Account segmentation on the basis of spend	Categorical nominal - values HNI, Regular, Regular Plus, Super, Super plus and variations with +
11	CC_Agent_Score	Satisfaction score given on customer care service provided	Categorical ordinal - values 1 to 5
12	Marital_Status	Marital status of primary customer	Categorical nominal - contains values Married, Single and Divorced
13	rev_per_month	Monthly average revenue from account in last 12 months	Continuous field. Contains values ranging from 1 to 140
14	Complain_ly	Complaints raised by account in last 12 months	Categorical - 0 (for no) or 1 (for yes)
15	rev_growth_yoy	revenue growth percentage of the account (last 12 months vs last 24 to 13 month)	Continuous field. Contains values ranging from 4 to 28
16	coupon_used_for_payment	How many times customers have used coupons to do the payment in last 12 months	Continuous field, but with limited range. Contains values ranging from 0 to 16
17	Day_Since_CC_connect	Number of days since no customers in the account has contacted the customer care	Continuous field. Contains values ranging from 0 to 47
18	Cashback	Monthly average cashback generated by account in last 12 months	Continuous field. Contains values ranging from 0 to 1997
19	Login_device	Preferred login device of the customers in the account	Categorical nominal - contains values Mobile, Computer

Table 3 Attribute description

The following table provides a basic statistical description of the numeric columns after data clean-up. It includes a 5-point summary of the numeric fields: minimum, maximum, 25th percentile, 50th percentile, and 75th percentile. Additionally, it contains the count of values present in each column, the mean, and the standard deviation.



	count	mean	std	min	25%	50%	75%	max
AccountID	11260.0	25629.500000	3250.626350	20000.0	22814.75	25629.5	28444.25	31259.0
Churn	11260.0	0.168384	0.374223	0.0	0.00	0.0	0.00	1.0
City_Tier	11148.0	1.653929	0.915015	1.0	1.00	1.0	3.00	3.0
CC_Contacted_LY	11158.0	17.867091	8.853269	4.0	11.00	16.0	23.00	132.0
Service_Score	11162.0	2.902526	0.725584	0.0	2.00	3.0	3.00	5.0
CC_Agent_Score	11144.0	3.066493	1.379772	1.0	2.00	3.0	4.00	5.0
Complain_ly	10903.0	0.285334	0.451594	0.0	0.00	0.0	1.00	1.0

Table 4 Basic description

Data Imbalance:

The dataset exhibits a clear imbalance regarding the target variable, which indicates whether a customer has churned or not. As shown in the plot below, for every 100 customers acquired, 17 have churned while 83 remain active. This distribution skews towards active customers. The goal of this exercise is to predict which customers will churn, focusing on the minority class '1'.

In a balanced dataset, there is an equal number of observations for each categorical value of the target variable (Churn). This balance allows the model to predict both classes with equal efficiency. However, customer churn typically involves fewer churned customers compared to active ones. In this dataset, approximately 17% of customers have churned, reflecting real-life churn data but posing modeling challenges.

Challenges of an Imbalanced Dataset:

- **Pattern Learning:**

With insufficient observations in the minority class, the model may struggle to learn its patterns effectively and may favor the majority class.

Solution: Resampling techniques, such as oversampling the minority class or under-sampling the majority class, can help. Compared to under-sampling, which may result in data loss, oversampling using techniques like SMOTE (Synthetic Minority Oversampling) can be beneficial. SMOTE generates synthetic data for churned customers based on existing observations.

However, it may not always improve model performance, and depending on the algorithm and type of SMOTE used, there is a risk of overfitting the training dataset.

- **Evaluation Metrics:**

Using accuracy as an evaluation metric in imbalanced classification problems is inappropriate. Even if the algorithm predicts all customers as belonging to the majority class, it would still yield an accuracy of 83.2%.

Solution: Metrics such as Precision, Recall, or F1-score for the minority class offer a better evaluation approach.

- **Data Treatment:**

SMOTE will be used as one of the data treatments. Models built on the imbalanced dataset will be compared against those built on the SMOTE-balanced dataset, and the model with the best performance metrics will be selected. Since SMOTE may not always guarantee better performance, a comparison is essential to make an informed decision.

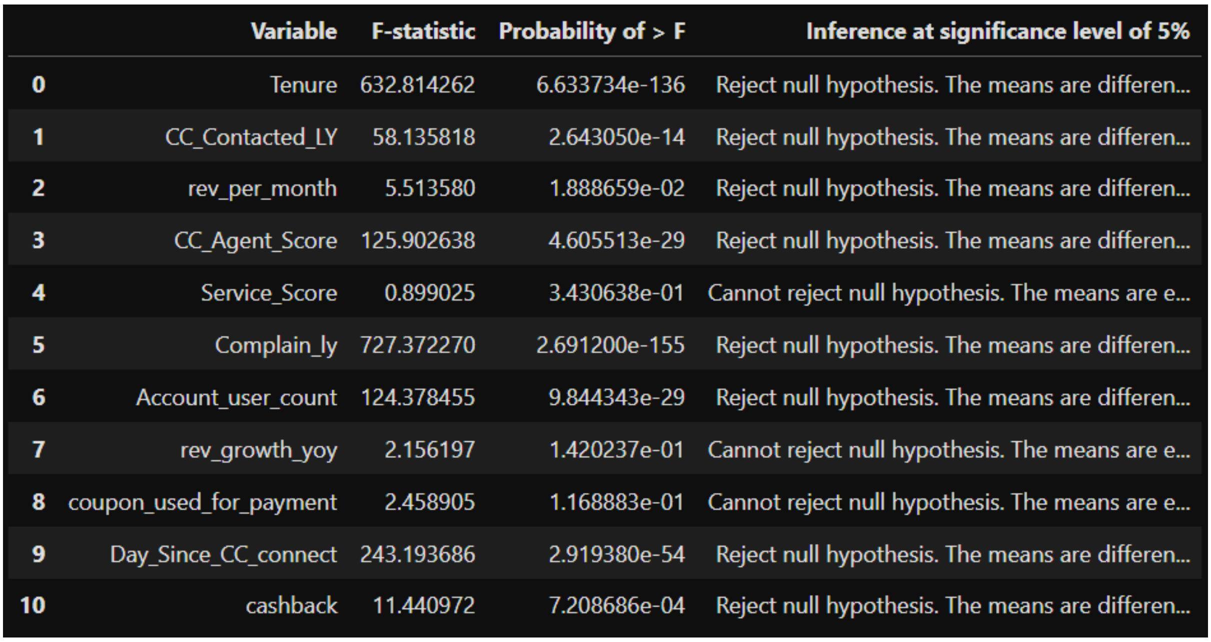
By addressing these challenges with appropriate techniques, the aim is to improve the model's ability to predict customer churn effectively.

One-way Anova:

Analysis of Variance (ANOVA) is a statistical method used to check if the means of two or more groups are significantly different from each other. The hypotheses for the test are as follows:

- **H0:** Means of all groups are equal
- **Ha:** At least the means of one pair of the groups are different

The Statsmodel library was used to perform the ANOVA test.



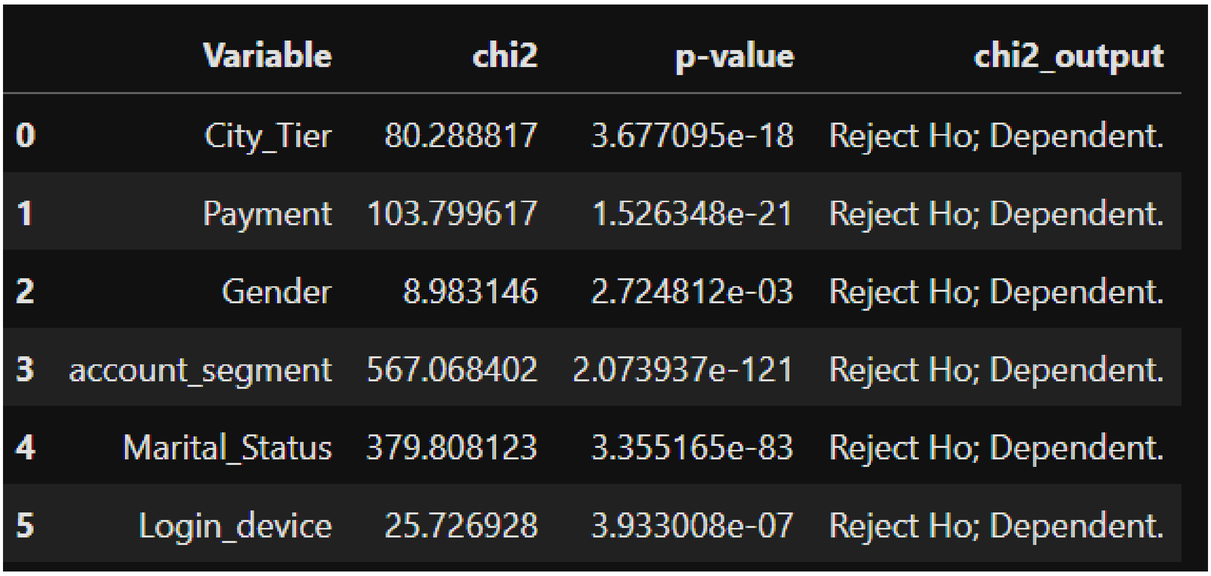
	Variable	F-statistic	Probability of > F	Inference at significance level of 5%
0	Tenure	632.814262	6.633734e-136	Reject null hypothesis. The means are differen...
1	CC_Contacted_LY	58.135818	2.643050e-14	Reject null hypothesis. The means are differen...
2	rev_per_month	5.513580	1.888659e-02	Reject null hypothesis. The means are differen...
3	CC_Agent_Score	125.902638	4.605513e-29	Reject null hypothesis. The means are differen...
4	Service_Score	0.899025	3.430638e-01	Cannot reject null hypothesis. The means are e...
5	Complain_ly	727.372270	2.691200e-155	Reject null hypothesis. The means are differen...
6	Account_user_count	124.378455	9.844343e-29	Reject null hypothesis. The means are differen...
7	rev_growth_yoy	2.156197	1.420237e-01	Cannot reject null hypothesis. The means are e...
8	coupon_used_for_payment	2.458905	1.168883e-01	Cannot reject null hypothesis. The means are e...
9	Day_Since_CC_connect	243.193686	2.919380e-54	Reject null hypothesis. The means are differen...
10	cashback	11.440972	7.208686e-04	Reject null hypothesis. The means are differen...

Table- 5 One-way Anova

At a significance level of 0.05 (5%), the tests for the variables rev_growth_yoy, Service_Score, and coupon_used_for_payment returned p-values greater than 0.05. Consequently, we cannot reject the null hypothesis (Ho) for these variables. This indicates that there is no statistically significant difference in the means of the two groups (churn=0 and churn=1) for these variables. As such, they are not significant predictors of the target variable. This conclusion aligns with the visual observations made using the bivariate boxplots for these variables.

Chi-square:

Categorical variables were evaluated using the Chi-squared test of independence at a significance level of 0.05. This test helps determine if there's a statistically significant relationship between two categorical variables, aiding in the decision of whether to include them in the model. The Chi-square test of independence compares the frequency of each category for one variable across the categories of another variable. The results are typically displayed in a contingency table, where rows represent categories for one variable and columns represent categories for the other.



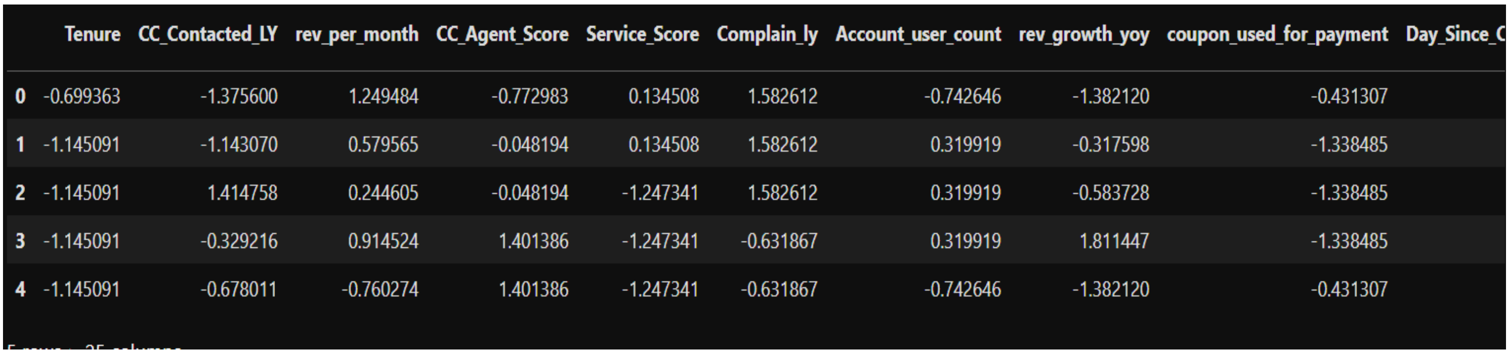
	Variable	chi2	p-value	chi2_output
0	City_Tier	80.288817	3.677095e-18	Reject Ho; Dependent.
1	Payment	103.799617	1.526348e-21	Reject Ho; Dependent.
2	Gender	8.983146	2.724812e-03	Reject Ho; Dependent.
3	account_segment	567.068402	2.073937e-121	Reject Ho; Dependent.
4	Marital_Status	379.808123	3.355165e-83	Reject Ho; Dependent.
5	Login_device	25.726928	3.933008e-07	Reject Ho; Dependent.

Table- 6 Chi-square test

A p-value less than 0.05 (typically ≤ 0.05) is considered statistically significant. It indicates strong evidence against the null hypothesis, as there is less than a 5% probability that the null hypothesis is true. Since the p-values for all the categorical variables are less than 0.05, we can reject the null hypothesis. Therefore, at the 5% significance level, we conclude that churn is not independent of these categorical variables. Consequently, we will retain all these categorical predictor variables for further analysis.

Encoding and Scaling of Dataset:

We will encode the dataset before the scaling process. We will do one hot encoding and label encoding on the categorical variables.



	Tenure	CC_Contacted_LY	rev_per_month	CC_Agent_Score	Service_Score	Complain_ly	Account_user_count	rev_growth_yoy	coupon_used_for_payment	Day_Since_C
0	-0.699363	-1.375600	1.249484	-0.772983	0.134508	1.582612	-0.742646	-1.382120	-0.431307	
1	-1.145091	-1.143070	0.579565	-0.048194	0.134508	1.582612	0.319919	-0.317598	-1.338485	
2	-1.145091	1.414758	0.244605	-0.048194	-1.247341	1.582612	0.319919	-0.583728	-1.338485	
3	-1.145091	-0.329216	0.914524	1.401386	-1.247341	-0.631867	0.319919	1.811447	-1.338485	
4	-1.145091	-0.678011	-0.760274	1.401386	-1.247341	-0.631867	-0.742646	-1.382120	-0.431307	

Table-7 Encoded and Scaled Dataset.

Before performing clustering on the dataset for business insights, we will scale the dataset to ensure all variables are on the same scale. For this, we use the StandardScaler method and obtain the encoded and scaled dataset as shown in Table-11.

Encoding the Categorical Variables:

During the data pre-processing stage, different encoding techniques were applied to handle categorical variables in the dataset.

- **Ordinal Encoding for 'Account Segment':**

○ The 'Account Segment' variable, which reflects spending-based segments, was encoded using ordinal encoding.

○ Categories like 'Regular', 'Regular Plus', 'Super', 'Super Plus', and 'HNI' were assigned numerical values from 1 to 5.

○ This hierarchical encoding helps the model capture the varying significance of different segments based on spending behaviors.

- **One-Hot Encoding for Other Categorical Variables:**

○ Categorical variables such as 'Payment', 'Gender', 'Marital Status', and 'Login Device' were processed using one-hot encoding.

○ One-hot encoding converts these categorical variables into binary columns for each category within the variable.

○ These encoding techniques ensure that the model can effectively utilize the categorical data during analysis and predictions.

Data Scaling and Preparation for Clustering:

Scaling the data is crucial for clustering to ensure that all features receive equal treatment and to prevent any bias caused by the different scales of variables. Here's how the data has been prepared for clustering:

- **Standardization of Numerical Variables:**

○ The numerical variables were scaled using standardization. This transformation adjusts the data to have a mean of 0 and a standard deviation of 1.

○ **Purpose:** Ensures that all numerical features are on the same scale, allowing for fair and unbiased clustering analysis.

- **Exclusion of the Target Variable 'Churn':**

○ The target variable 'Churn' was excluded from scaling to maintain its original interpretation.

○ **Purpose:** To preserve the binary nature of the target variable for accurate analysis and predictions.

Clustering of Dataset:

Clustering is an unsupervised machine learning task that groups a set of objects such that objects in the same group (or cluster) are more similar to each other than to those in other groups. It helps in identifying natural groupings within the dataset based on similarities.

The cluster profile was formed by grouping observations by clusters and finding the mean for all features. This helped in understanding the characteristics of each cluster.

Clustering helps in identifying similar customer groups, enabling targeted strategies for different segments. Including churn in the cluster profile allows for a better understanding of customer retention patterns across different groups.

Hierarchical Clustering Method:

For hierarchical clustering, we will plot a dendrogram using Ward's linkage method and Euclidean distance, truncating the dendrogram up to a p-value of 10, and obtain the following plot as shown below.

.

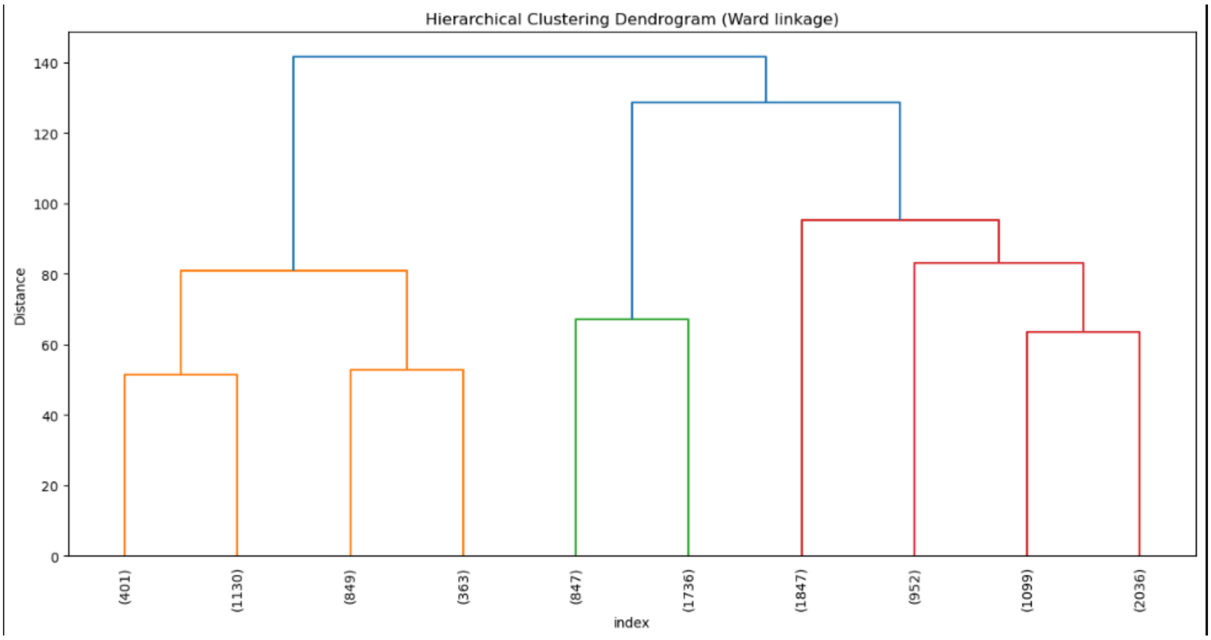


Figure1 - Dendrogram

Kmean Method for clustering:

will find out the WSS (within sum of squares) values for 10 clusters and plot them in elbow plot to find out the optimum number of K for K-Means algorithm. We can see the plot below.

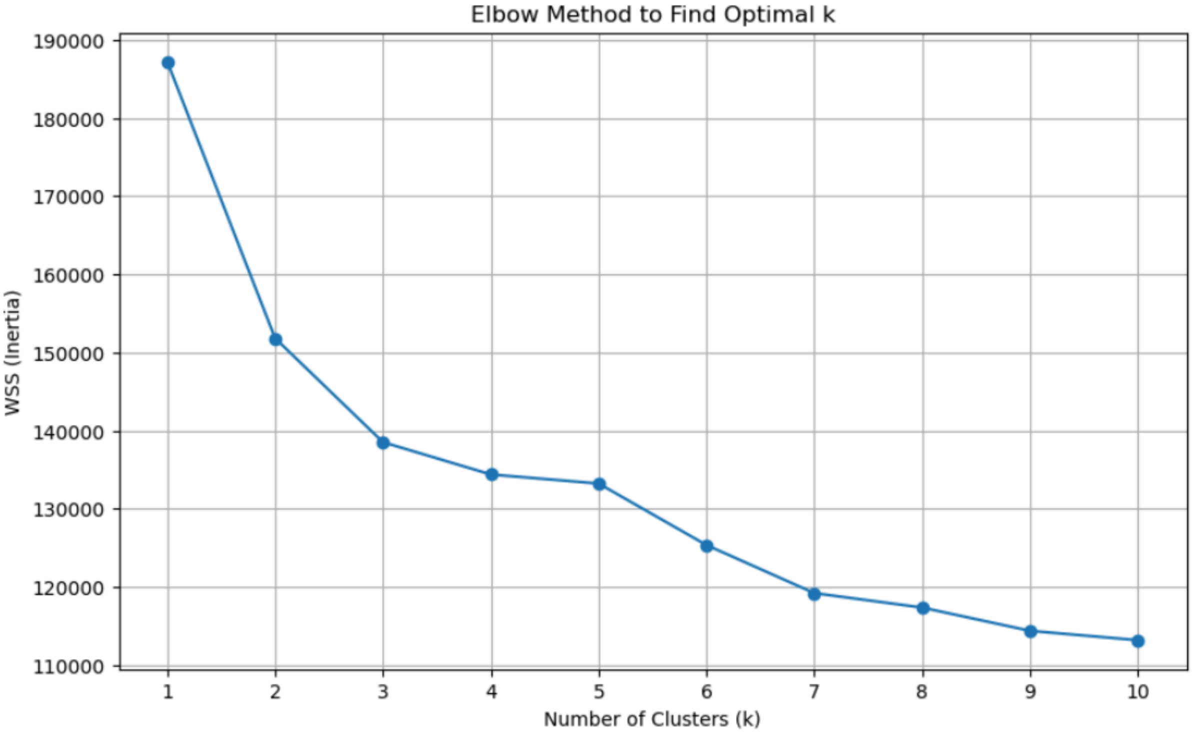


Figure-2 Elbow Plot

From the above plot and Average Silhouette score for all 10 clusters we can say that no. of clusters or K value should be 2 for carrying out the K-Means clustering.

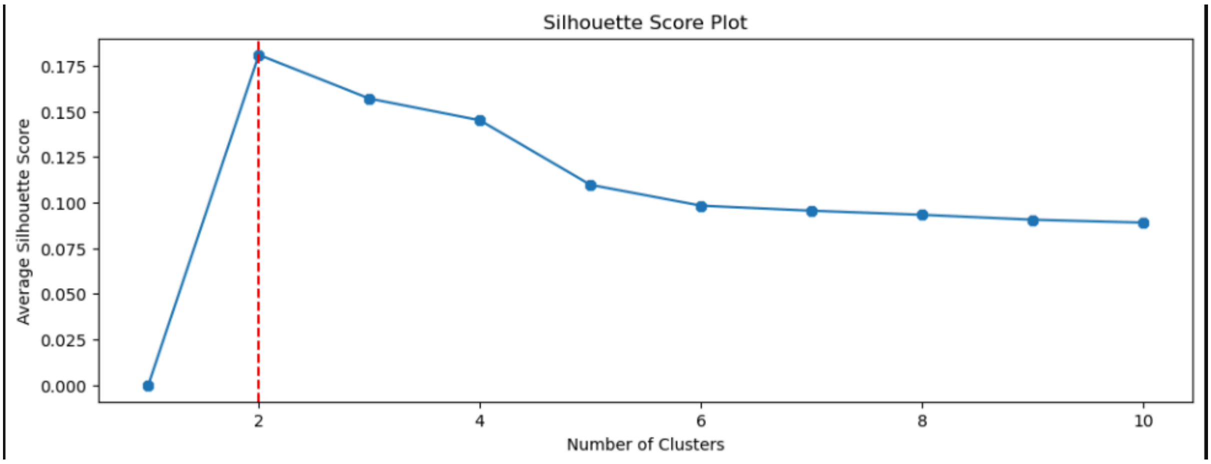


Figure-3 Silhouette score chart

Thus, We will create a cluster based on the K optimized and then add it to the dataset.

Here, Cluster_maxclust and cluster_distance is obtained from Dendrogram and ward’s linkage method and the Kmean column is obtained from the K-means clustering method itself.

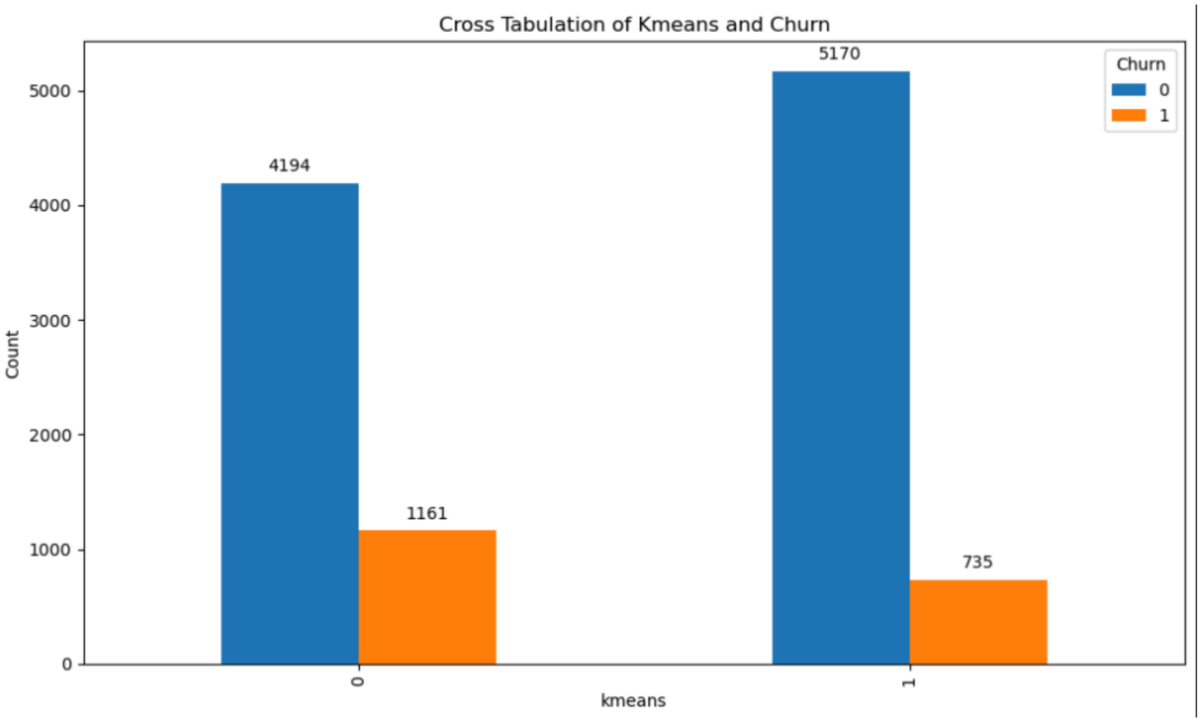


Figure-4 Kmeans vs Churn

Here we can see in figure-13 showing K Means vs churn for each cluster formed. And due to data imbalance we can see that while modeling there are going to be biassed towards some values in very variable.

Removal of Unwanted Variables:

The following checks were performed to determine if any columns could be dropped before the modeling exercise:

1. **Unique Values for Each Observation:**

○ Any variable that has unique values for each observation, such as the AccountID field, would not contribute to the model as it is just an ID field to tag each observation.

2. **Constant Values:**

○ Any variable that remains constant for all or most of the observations does not add any strength to prediction. As observed from the histogram (numeric) and count plots/value counts (categorical), there are no variables that have constant values for all observations.

3. **High Percentage of Nulls:**

○ Any variable that has nulls in more than 25-30% of observations. The maximum nulls present are in the cashback variable, which contains 4% nulls. Hence, no column will be dropped.

4. **Strong Correlation with Another Predictor:**

○ Any predictor variable that has a strong correlation with another predictor variable. Then one of the variables can be dropped. As seen in the correlation heatmap, there are no strong correlations, hence no variable needs to be dropped.

5. **Weak Correlation with the Target Variable:**

○ Any predictor variable that has a very weak correlation with the target variable. As seen from the Chi-square test and ANOVA test in the bivariate analysis section above, two variables – rev_growth_yoy and coupon_used_for_payment – were found to not be significant (at a significance level of 5%). Hence, these two variables would be dropped from further processing.

6. **VIF Check:**

○ Three variables (Cashback, Service Score, Cluster Code) were dropped as part of the VIF check in a later step, as explained in section 5.2.

Addition of New Variables:

- **Cluster Code:**

○ Cluster code has been added to the dataset (details about clustering are provided in section 3.5). It may be used when experimenting with model building.

Missing Value Treatment:

- **Percentage of Nulls:**

○ The percentage of nulls or missing values present in the predictor variables of the dataset are as follows:

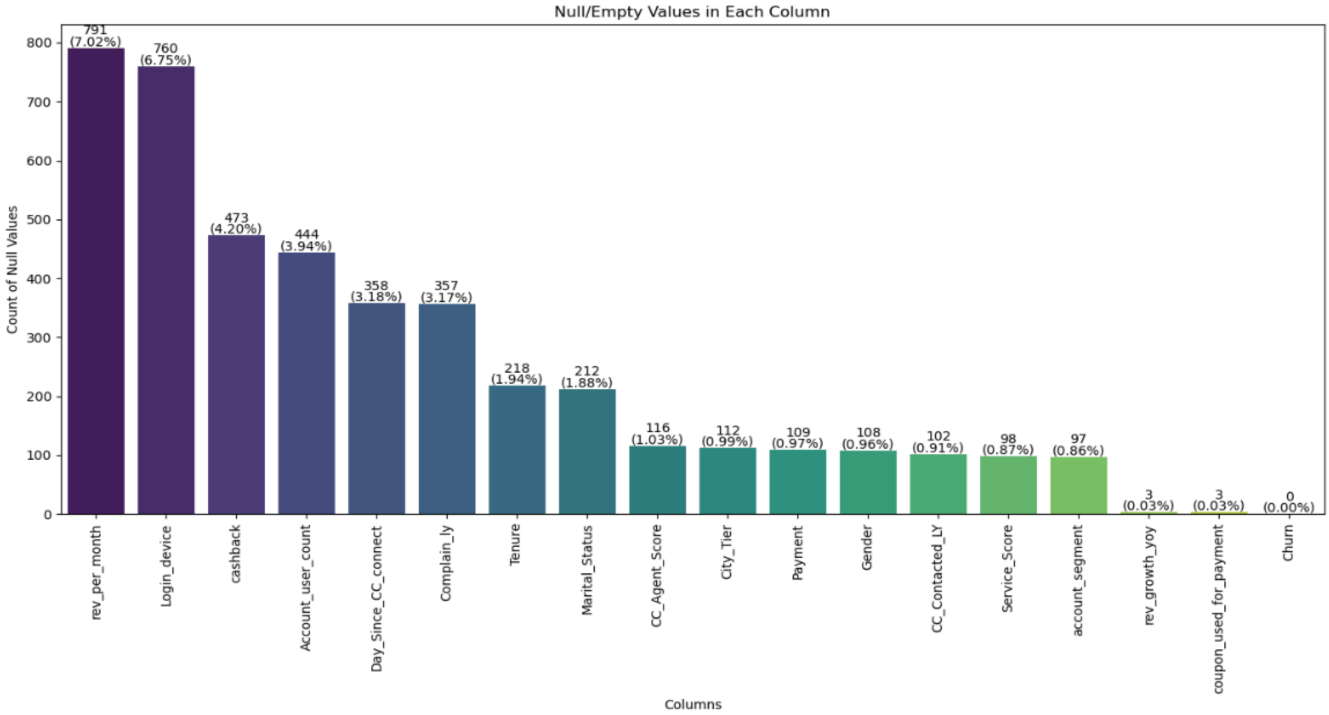


FIgure 5 visualization of nulls

Missing Value Treatment:

Missing value treatment was performed using KNN imputation, a distance-based method. The following treatments were done as prerequisites for missing value treatment using KNN:

1. **Numeric Conversion:**
○ All variables needed to be numeric. Any object-type categorical variables were encoded suitably (label or one-hot encoding).
2. **One-Hot Encoding:**
○ The following columns were one-hot encoded as they contained nominal categorical variables. The first encoded variable was dropped to avoid multicollinearity.
3. **Renaming Columns:**
○ Column names were renamed to shorten, make them uniform, and remove blank spaces resulting from some category values from one-hot encoded columns (e.g., Payment_Credit Card).
4. **Scaling Variables:**
○ All variables needed to be scaled as KNN is a distance-based algorithm. Scaling was done using the StandardScaler function from the SKLearn library for the predictor variables. The target variable was left as-is.
5. **Null Imputation:**
○ Null imputation was done using Sklearn's KNNImputer function. This algorithm imputed missing values using K-nearest neighbors.

Outlier treatment

Some outliers for certain variables are closer to the whisker, while there are a group of outliers far beyond the whisker with no intermediate values. For instance, rev_per_month has a significant gap between 30 and 100, indicating the absence of values in that range. These extreme outliers do not correlate with corresponding outliers in the cashback field. We cannot rule out these outliers as incorrect values, as they may belong to hotels with many rooms. However, models like logistic regression are sensitive to outliers and may not perform well if outliers are left untreated.

Hence, two approaches to modeling were performed: one set of data with outliers treated for outlier-sensitive models and another set of data with outliers left untreated for outlier-resistant algorithms such as Random Forest, and during tuning/trials of other algorithms.

Coupon_used_for_payment has a very limited range of 0 to 16. Hence, for the purpose of this analysis, the outliers will not be treated (similar to categorical variables).

For the outlier-treated dataset, outliers beyond the upper and lower whiskers were treated by capping to the lower and upper range, where:

3. **Lower range:** 1st quartile - (1.5 * IQR)
4. **Upper range:** 3rd quartile + (1.5 * IQR)
    4.1. **IQR:** 3rd quartile – 1st quartile value

3. Exploratory data analysis

Univariate Analysis of Numerical Variables:

- **Tenure**

○ **Median:** 9 months

○ **Observation:** 50% of the tenure data is less than 9 months.

○ **Insights:** Most customers have relatively short tenures. There are a few outliers indicating long-tenure customers who might be loyal and valuable.

- **CC_Contacted_LY**

○ **Median:** 16

○ **Observation:** Represents typical customer care contacts in a year.

○ **Insights:** Most interactions range from 0 to 50. Indicates robust customer support. High contact frequency may indicate issues.

- **Rev_per_month**

○ **Box Plot Observation:** 75% of accounts generate less than INR 10,000 monthly.

○ **Outliers show higher revenue, potentially from multi-account customers.**

○ **Insights:** Majority of accounts have moderate revenue. Identifying high-revenue outliers can help target high-value customers.

- **Day_Since_CC_connect**

○ **Median:** 3 days

○ **Observation:** Indicates time since last customer care contact.

○ **Insights:** 50% of cases have swift resolution within 3 days. Some instances show delays up to 47 days.

- **Cashback**

○ **Median:** INR 165.25

○ **Observation:** Monthly average given in the past year.

○ **Insights:** Typical cashback, but outliers indicate higher amounts for specific accounts, possibly high-spending customers.

- **Rev_growth_yoy**

- ○ **Median:** Under 15%
- ○ **Observation:** Revenue growth percentage.
- ○ **Insights:** Moderate expansion for many accounts. Exceptional cases show up to 28% growth.

- ● **Coupon_used_for_payment**
- ○ **Median:** 1
- ○ **Observation:** Coupon usage behavior.
- ○ **Insights:** Most customers use coupons sparingly. Outliers maxing at 14 uses highlight heavy coupon users.

Univariate Analysis of Categorical Variables:

- ● **Churn**
- ○ **Observation:** Not explicitly mentioned in your text.
- ○ **Insights:** Analyzing churn rate is crucial for understanding customer retention. Addressing class imbalance (83.2% non-churn vs. 16.8% churn) is important for accurate predictions.

- ● **City_Tier**
- ○ **Observation:** Categorized into different tiers.
- ○ **Insights:** Distribution shows most customers in tier 1 (7375), followed by tier 3 (3405) and tier 2 (480). Customized strategies and offers can be tailored based on these insights.

- ● **Payment Method**
- ○ **Distribution:** Debit Card: 40%, Credit Card: 30%, E-Wallet: 20%, Cash on Delivery: 10%, UPI: 5%
- ○ **Insights:** Customers prefer convenient, cashless methods. Focus on enhancing these options.

- ● **Gender**
- ○ **Distribution:** Male: 70%, Female: 30%
- ○ **Insights:** Majority of customers are male. Consider developing targeted marketing campaigns for female customers.

- ● **Account Segment**
- ○ **Distribution:** Regular Plus: 40%, Super Plus: 30%, HNI: 20%, Regular: 10%
- ○ **Insights:** Most customers are in premium segments, indicating a willingness to pay for premium services.

- ● **Marital Status**
- ○ **Distribution:** Married: 60%, Single: 30%, Divorced: 10%
- ○ **Insights:** Majority are married. Develop targeted campaigns for married couples.

- **Login Device**
  - **Distribution:** Computer: 60%, Mobile Phone: 40%
  - **Insights:** Focus on improving user experience on both devices, particularly mobile.
- **Customer Satisfaction Scores (Service_Score)**
  - **Distribution:** Most customers (5588) rate 3.0 (moderately satisfied).
  - **Insights:** Varying satisfaction scores highlight areas for service improvements to enhance loyalty and retention.
- **Complaint Occurrences (Complain_ly)**
  - **Distribution:** 72.4% had no complaints, 27.6% raised complaints.
  - **Insights:** Majority satisfied with service, but areas for improvement exist for those who raised complaints.
- **Account User Count (Account_user_count)**
  - **Distribution:** Most accounts have 4 customers (50.1%), followed by 3 (32.4%) and 5 (16.0%).
  - **Insights:** Majority of accounts are used by multiple users, indicating possible shared or family accounts.
- **Additional Insights:**
  - Most customers are in tier 1.0, followed by tier 3.0, with tier 2.0 having the fewest.
  - Debit Card is the most preferred, followed by Credit Card, E-Wallet, Cash on Delivery, and UPI.
  - Majority are male (60.5%), with females making up 39.5%.
  - Regular Plus and Super Plus segments dominate, followed by HNI and Regular.
  - Majority are married, followed by singles and divorced.
  - Mobile phones are the most used (73.2%), followed by computers (26.8%).

Bivariate Analysis:

- **Low Correlation Among Variables:**
  - **Observation:** Minimal multicollinearity in the dataset.
  - **Insight:** Reduces the risk of multicollinearity issues, making regression models more stable and interpretable.
- **Positive Correlations:**
  - Tenure and Account_user_count
  - Tenure and Service Score
  - Account_user_count and Service Score
  - Account_user_count and rev_per_month
  - Service Score and rev_per_month

- rev_per_month and cashback
- Complain_ly and Day_Since_CC_connect

- **Negative Correlations:**
  - Tenure and CC_Contacted_LY
  - Service Score and CC_Contacted_LY
  - Account_user_count and CC_Contacted_LY
  - rev_per_month and CC_Contacted_LY
  - Day_Since_CC_connect and CC_Contacted_LY
- **KDE Plot and Heatmap Insights:**
  - **Tenure KDE Plot:**
    - **Observation:** Slight separation with churned customers falling on the lower side.
    - **Insight:** Churned customers typically have shorter tenures.
- **No Linear Relationships:**
  - **Observation:** Lack of strong linear relationships between continuous variables.
  - **Insight:** Beneficial for model stability and interpretability.

Bivariate Analysis of Numerical vs Categorical Variables:

- **Influence on Churn:**
  - **Tenure and Days_since_CC_connect:**
    - **Observation:** Significant influence on churn. Churned customers generally show shorter tenures and more recent customer care connections.
    - **Insight:** These variables are important for modeling churn prediction.
  - **Coupon_used_for_payment and rev_growth_yoy:**
    - **Observation:** Minimal difference between churned and non-churned distributions.
    - **Insight:** These variables might have limited predictive power for churn.
- **Spending Patterns:**
  - **Payment Methods:**
    - **Higher Median Rev_per_month:**
    - **Debit Card and Credit Card:** Customers using these methods spend more.

- **E-Wallet and Cash on Delivery:** Lower spending.
- **Insight:** Focus on promoting Debit and Credit Card payment methods.

○ **Gender:**

- **Higher Median Rev_per_month:**
- **Males:** Tend to spend more, distribution skewed with few high-value customers.
- **Insight:** Identify and target high-value male customers.

○ **Account Segment:**

- **Higher Median Rev_per_month:**
- **HNI Customers:** Spend significantly more.
- **Skewed Distribution:** Presence of few high-value HNI customers.
- **Insight:** Cater to HNI customers with premium services.

○ **Marital Status:**

- **Higher Median Rev_per_month:**
- **Married Customers:** Spend more than single or divorced customers.
- **Insight:** Target high-value married customers.

○ **Login Device:**

- **Higher Median Rev_per_month:**
- **Computer Logins:** Indicate higher spending.
- **Insight:** Enhance user experience for desktop users.

- **Customer Satisfaction:**

○ **Payment Methods:**

- **Debit Card:** Higher satisfaction.
- **Cash on Delivery:** Lower satisfaction.
- **Insight:** Improve service for Cash on Delivery users.

○ **Gender:**

- **Male Customers:** Higher likelihood of using services.
- **Female Customers:** Higher churn rate.
- **Insight:** Focus on reducing female customer churn.

○ **City Tier:**

- **Tier 1 Cities:** Higher usage and satisfaction.
- **Tier 3 and 4 Cities:** Higher churn rates.

■ **Insight:** Tailor strategies for lower-tier cities to improve retention.

○ **Service Scores:**

■ **Payment Methods:**

■ **Credit Card:** Higher service scores.

■ **Cash on Delivery:** Lower service scores.

■ **Insight:** Focus on enhancing service for Cash on Delivery users.

○ **City Tier:**

■ **Tier 1 Cities:** More likely to give higher service scores.

■ **Insight:** Continue to maintain high service quality in tier 1 cities.

Bivariate Analysis of Categorical Variables:

- **City Tier:**

○ **Observation:** Shows a noticeable difference in distribution between churned and current customers.

○ **Insight:** Indicates that the city tier might influence customer churn and could be a useful predictor in the model.

- **Account Segment:**

○ **Observation:** Different distributions for churned vs. current customers.

○ **Insight:** Suggests that different account segments have varying churn rates and can impact the model's predictions.

- **Marital Status:**

○ **Observation:** Shows differences in the distribution.

○ **Insight:** Marital status might have an effect on churn behavior and could be relevant for the model.

- **Less Significant Variables:**

○ **Gender:**

■ **Observation:** Similar distributions within churned and current customers.

■ **Insight:** Gender may not significantly impact the churn prediction model.

- **Interpretation:**

○ The percentage distribution within each categorical variable provides insights into how these variables might contribute to churn prediction.

○ Variables showing significant distribution differences between churned and current customers are likely to be more influential in the model.

- Those with similar distributions might have less impact on the model's performance.

4. Model Building and Validation

Applicable Algorithms:

- **Binary Classification:** This business case requires predicting customer churn, a binary classification problem with outcomes '0' (will not churn) and '1' (will churn).
- **Supervised Learning:** The target variable 'Churn' needs prediction, making this a supervised learning problem.
- **Algorithms for Classification:**
  - **Linear Classification:** Logistic Regression, Linear Discriminant Analysis, Naïve Bayes.
  - **Non-linear Classification:** SVMs (non-linear adaptations), Decision Tree, K-Nearest Neighbor (KNN).
  - **Ensemble Models:** Random Forest, AdaBoost, Gradient Boost.
- **Algorithm Assumptions:** Algorithms have specific assumptions about the data, influencing their performance.
-

Methodology:

- **Data Preprocessing Variants:** Different treatments of pre-processed data were prepared, including scenarios with and without SMOTE and hypertuned datasets.
- **Variance Inflation Factor (VIF):** Calculated for predictor variables. Predictors with VIF greater than 5 were identified. 'Cashback', 'Service Score', 'Clusters', and 'User Count' had high VIF. 'User Count' was retained due to a significant Chi-square value from EDA; the other variables were dropped.
- **Variable Selection:** 'rev_growth_yoy' and 'coupon_used_for_payment' were dropped after ANOVA and Chi-square tests, and verification against EDA plots.
- **Data Scaling:** Scaled data used for distance-based algorithms like KNN and ANN. SMOTE resampled data also tested.
- **Data Splitting:** Data split into train (70%) and test (30%) sets, with a similar distribution of the target variable as the original dataset (16.8% churn).
- **Algorithm Testing:** Eight algorithms were chosen. For each:
  - Constructed base model with default hyperparameters and evaluated on train and test datasets.
  - Used different data treatments and recorded performance.
  - Tuned hyperparameters using GridSearchCV from Sklearn and manual adjustments.
  - Measured performance of tuned algorithms.
  - Extracted feature importance from models using built-in attributes or Sklearn's Permutation feature importance for black-box models.

Model Evaluation Metrics:

- **Comparison Criteria:** The best model was selected based on evaluation metrics for train and test data, ensuring no overfitting or underfitting.
- **Precision:** High precision for churn customers to minimize unnecessary freebies by the revenue assurance team.
  - **Precision:** True Positive / (True Positive + False Positive)
- **F1-Score:** Ensures high recall alongside precision. The model should effectively predict actual churns to address the churn problem.
- **Model Interpretability:** Preference for interpretable models.
- **Computational Efficiency:** Avoid computationally expensive models like KNN.

Model Building and Tuning:

- **Algorithms Tested:** Logistic Regression, Linear Discriminant Analysis, SVM, KNN, Random Forest, AdaBoost, Gradient Boost using Sklearn implementations. Logistic Regression was also executed using the Statsmodel package.
- **Tabulated Results:** Detailed results for all algorithms, including tuning efforts and performance. For brevity, top-performing models and their feature importances are summarized.
- **Effort for Model Tuning:** Performance improvement was achieved through:
  - Testing eight different algorithms across various methods.
  - Constructing base models with default hyperparameters and tuning using GridSearchCV.
  - Changing underlying data (SMOTE resampled/non-resampled) and observing performance effects.
  - Using ensemble methods (Random Forest, AdaBoost, Gradient Boost) and tuning their hyperparameters.

Train-Test Split: A Crucial Step:

Purpose: To create a robust predictive model that performs well on unseen data.

Process: Data is divided into two sets:

- **Training Set (70%)**: This dataset teaches the model the underlying patterns.
- **Test Set (30%)**: This set evaluates the model's performance on new data.

Data Breakdown

- **X_train (7882, 17)**: Training dataset with 7,882 rows and 17 features, used to teach the model patterns.
- **X_test (3378, 17)**: Test dataset with 3,378 rows and 17 features, unseen by the model, for evaluating its performance.
- **y_train (7882,)**: Target variable for the training samples, guiding the model's learning process.
- **y_test (3378,)**: Target variable for the test samples, used to measure the model's predictions against actual outcomes on unseen data.

Performance of Various Models:

To evaluate the effectiveness of various classification models, we conducted a comprehensive analysis using classification reports and Area Under the Curve (AUC) scores. These metrics provide valuable insights into the models' abilities to distinguish between different classes and their overall predictive power.

Evaluation Metrics:

- **Precision:** Measures the accuracy of positive predictions (churn). High precision indicates fewer false positives.
- **Recall:** Measures the ability to identify all positive instances (churn). High recall indicates fewer false negatives.
- **F1-Score:** The harmonic mean of precision and recall, providing a balance between the two. High F1-score ensures that both precision and recall are optimized.
- **AUC Score:** Measures the overall performance of the classification model, indicating the model's ability to distinguish between classes. A higher AUC score signifies better model performance.

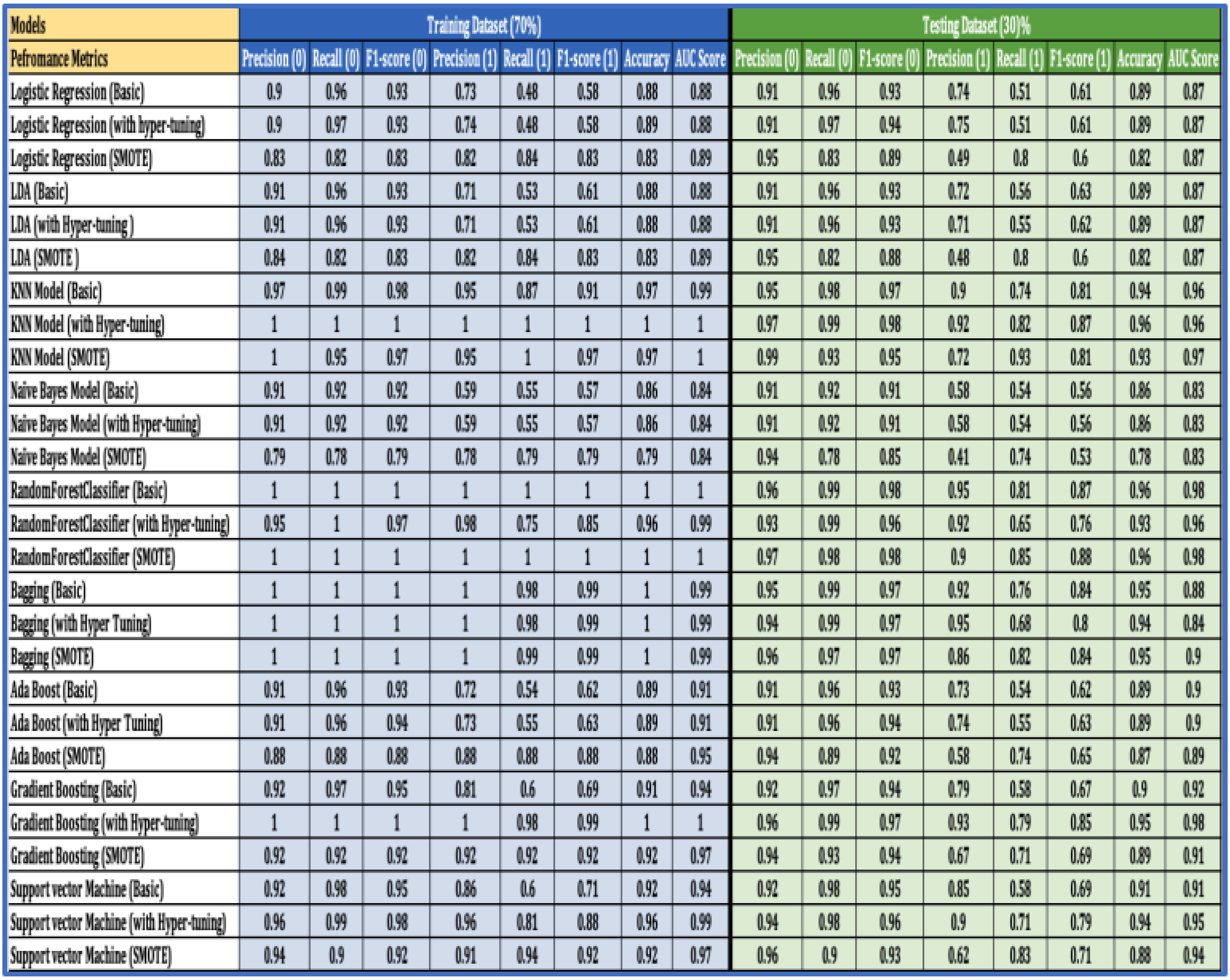
Analysis Insights:

By meticulously analyzing these metrics, we can clearly understand each model's strengths and weaknesses. This assessment helps in making informed decisions about which models are best suited for addressing the specific business problem at hand.

Prediction Labels:

- **"0":** Indicates a prediction of non-churn.
- **"1":** Indicates a prediction of churn.

By leveraging these metrics, we can ensure that our chosen model not only accurately predicts churn but also aligns with the business objectives, minimizing false positives and negatives and effectively addressing customer churn.



Models	Training Dataset (70%)								Testing Dataset (30)%							
Pefromance Metrics	Precision (0)	Recall (0)	F1-score (0)	Precision (1)	Recall (1)	F1-score (1)	Accuracy	AUC Score	Precision (0)	Recall (0)	F1-score (0)	Precision (1)	Recall (1)	F1-score (1)	Accuracy	AUC Score
Logistic Regression (Basic)	0.9	0.96	0.93	0.73	0.48	0.58	0.88	0.88	0.91	0.96	0.93	0.74	0.51	0.61	0.89	0.87
Logistic Regression (with hyper-tuning)	0.9	0.97	0.93	0.74	0.48	0.58	0.89	0.88	0.91	0.97	0.94	0.75	0.51	0.61	0.89	0.87
Logistic Regression (SMOTE)	0.83	0.82	0.83	0.82	0.84	0.83	0.83	0.89	0.95	0.83	0.89	0.49	0.8	0.6	0.82	0.87
LDA (Basic)	0.91	0.96	0.93	0.71	0.53	0.61	0.88	0.88	0.91	0.96	0.93	0.72	0.56	0.63	0.89	0.87
LDA (with Hyper-tuning)	0.91	0.96	0.93	0.71	0.53	0.61	0.88	0.88	0.91	0.96	0.93	0.71	0.55	0.62	0.89	0.87
LDA (SMOTE)	0.84	0.82	0.83	0.82	0.84	0.83	0.83	0.89	0.95	0.82	0.88	0.48	0.8	0.6	0.82	0.87
KNN Model (Basic)	0.97	0.99	0.98	0.95	0.87	0.91	0.97	0.99	0.95	0.98	0.97	0.9	0.74	0.81	0.94	0.96
KNN Model (with Hyper-tuning)	1	1	1	1	1	1	1	1	0.97	0.99	0.98	0.92	0.82	0.87	0.96	0.96
KNN Model (SMOTE)	1	0.95	0.97	0.95	1	0.97	0.97	1	0.99	0.93	0.95	0.72	0.93	0.81	0.93	0.97
Naive Bayes Model (Basic)	0.91	0.92	0.92	0.59	0.55	0.57	0.86	0.84	0.91	0.92	0.91	0.58	0.54	0.56	0.86	0.83
Naive Bayes Model (with Hyper-tuning)	0.91	0.92	0.92	0.59	0.55	0.57	0.86	0.84	0.91	0.92	0.91	0.58	0.54	0.56	0.86	0.83
Naive Bayes Model (SMOTE)	0.79	0.78	0.79	0.78	0.79	0.79	0.79	0.84	0.94	0.78	0.85	0.41	0.74	0.53	0.78	0.83
RandomForestClassifier (Basic)	1	1	1	1	1	1	1	1	0.96	0.99	0.98	0.95	0.81	0.87	0.96	0.98
RandomForestClassifier (with Hyper-tuning)	0.95	1	0.97	0.98	0.75	0.85	0.96	0.99	0.93	0.99	0.96	0.92	0.65	0.76	0.93	0.96
RandomForestClassifier (SMOTE)	1	1	1	1	1	1	1	1	0.97	0.98	0.98	0.9	0.85	0.88	0.96	0.98
Bagging (Basic)	1	1	1	1	0.98	0.99	1	0.99	0.95	0.99	0.97	0.92	0.76	0.84	0.95	0.88
Bagging (with Hyper Tuning)	1	1	1	1	0.98	0.99	1	0.99	0.94	0.99	0.97	0.95	0.68	0.8	0.94	0.84
Bagging (SMOTE)	1	1	1	1	0.99	0.99	1	0.99	0.96	0.97	0.97	0.86	0.82	0.84	0.95	0.9
Ada Boost (Basic)	0.91	0.96	0.93	0.72	0.54	0.62	0.89	0.91	0.91	0.96	0.93	0.73	0.54	0.62	0.89	0.9
Ada Boost (with Hyper Tuning)	0.91	0.96	0.94	0.73	0.55	0.63	0.89	0.91	0.91	0.96	0.94	0.74	0.55	0.63	0.89	0.9
Ada Boost (SMOTE)	0.88	0.88	0.88	0.88	0.88	0.88	0.88	0.95	0.94	0.89	0.92	0.58	0.74	0.65	0.87	0.89
Gradient Boosting (Basic)	0.92	0.97	0.95	0.81	0.6	0.69	0.91	0.94	0.92	0.97	0.94	0.79	0.58	0.67	0.9	0.92
Gradient Boosting (with Hyper-tuning)	1	1	1	1	0.98	0.99	1	1	0.96	0.99	0.97	0.93	0.79	0.85	0.95	0.98
Gradient Boosting (SMOTE)	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.97	0.94	0.93	0.94	0.67	0.71	0.69	0.89	0.91
Support vector Machine (Basic)	0.92	0.98	0.95	0.86	0.6	0.71	0.92	0.94	0.92	0.98	0.95	0.85	0.58	0.69	0.91	0.91
Support vector Machine (with Hyper-tuning)	0.96	0.99	0.98	0.96	0.81	0.88	0.96	0.99	0.94	0.98	0.96	0.9	0.71	0.79	0.94	0.95
Support vector Machine (SMOTE)	0.94	0.9	0.92	0.91	0.94	0.92	0.92	0.97	0.96	0.9	0.93	0.62	0.83	0.71	0.88	0.94

Table -8: Classification Report and AUC score

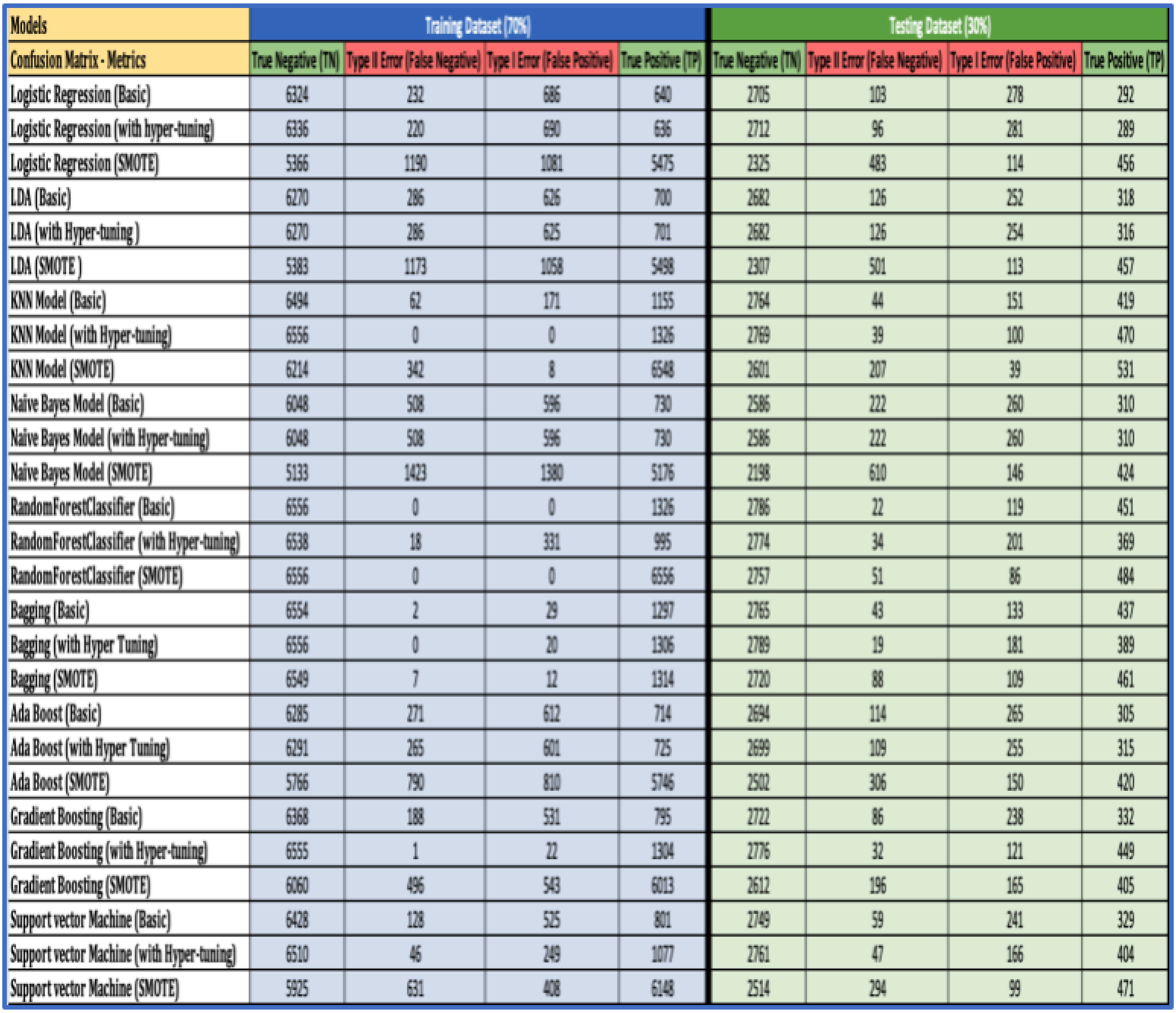
Confusion Matrix of All Models for Comparison:

Components:

- **True Positives (TP):** Correctly predicted positive cases (e.g., actual churn predicted as churn).
- **True Negatives (TN):** Correctly predicted negative cases (e.g., actual non-churn predicted as non-churn).
- **False Positives (FP):** Incorrectly predicted positive cases (e.g., actual non-churn predicted as churn).
- **False Negatives (FN):** Incorrectly predicted negative cases (e.g., actual churn predicted as non-churn).

Benefits of Using a Confusion Matrix:

- **Detailed Performance Analysis:** By interpreting these values, we gain insights into the strengths and weaknesses of each model.
- **Identifying Patterns:** It helps in identifying patterns of correct and incorrect predictions, highlighting areas where models are performing well and where they might be struggling.
- **Quantifying Effectiveness:** We can quantify the effectiveness of different models in terms of their ability to accurately predict outcomes.
- **Spotting Areas for Improvement:** The matrix reveals potential areas of improvement, guiding us in refining the models to enhance their performance.
- **Informed Decision-Making:**
  - **Model Selection:** This thorough analysis assists us in making informed decisions about which models are best suited for our specific business objectives.
  - **Model Refinement:** It also provides a clear direction for further tuning and refining the models to achieve better predictive accuracy.



Models	Training Dataset (70%)				Testing Dataset (30%)			
Confusion Matrix - Metrics	True Negative (TN)	Type II Error (False Negative)	Type I Error (False Positive)	True Positive (TP)	True Negative (TN)	Type II Error (False Negative)	Type I Error (False Positive)	True Positive (TP)
Logistic Regression (Basic)	6324	232	686	640	2705	103	278	292
Logistic Regression (with hyper-tuning)	6336	220	690	636	2712	96	281	289
Logistic Regression (SMOTE)	5366	1190	1081	5475	2325	483	114	456
LDA (Basic)	6270	286	626	700	2682	126	252	318
LDA (with Hyper-tuning)	6270	286	625	701	2682	126	254	316
LDA (SMOTE)	5383	1173	1058	5498	2307	501	113	457
KNN Model (Basic)	6494	62	171	1155	2764	44	151	419
KNN Model (with Hyper-tuning)	6556	0	0	1326	2769	39	100	470
KNN Model (SMOTE)	6214	342	8	6548	2601	207	39	531
Naive Bayes Model (Basic)	6048	508	596	730	2586	222	260	310
Naive Bayes Model (with Hyper-tuning)	6048	508	596	730	2586	222	260	310
Naive Bayes Model (SMOTE)	5133	1423	1380	5176	2198	610	146	424
RandomForestClassifier (Basic)	6556	0	0	1326	2786	22	119	451
RandomForestClassifier (with Hyper-tuning)	6538	18	331	995	2774	34	201	369
RandomForestClassifier (SMOTE)	6556	0	0	6556	2757	51	86	484
Bagging (Basic)	6554	2	29	1297	2765	43	133	437
Bagging (with Hyper Tuning)	6556	0	20	1306	2789	19	181	389
Bagging (SMOTE)	6549	7	12	1314	2720	88	109	461
Ada Boost (Basic)	6285	271	612	714	2694	114	265	305
Ada Boost (with Hyper Tuning)	6291	265	601	725	2699	109	255	315
Ada Boost (SMOTE)	5766	790	810	5746	2502	306	150	420
Gradient Boosting (Basic)	6368	188	531	795	2722	86	238	332
Gradient Boosting (with Hyper-tuning)	6555	1	22	1304	2776	32	121	449
Gradient Boosting (SMOTE)	6060	496	543	6013	2612	196	165	405
Support vector Machine (Basic)	6428	128	525	801	2749	59	241	329
Support vector Machine (with Hyper-tuning)	6510	46	249	1077	2761	47	166	404
Support vector Machine (SMOTE)	5925	631	408	6148	2514	294	99	471

Table -9: Confusion Matrix Analysis of all models

Performance Analysis of Various Models

Logistic Regression: Basic

This basic logistic regression model shows moderate success in identifying churn. It correctly flags 48% of churners in training, improving slightly to 51% in testing (recall). When it predicts churn, it's correct 73% of the time during training and 74% during testing (precision). The overall balance between precision and recall, represented by the F1-score, is 0.58 for training and 0.61 for testing. The confusion matrices reveal that the model struggles, misclassifying 686 real churners as non-churners (false negatives) and 232 non-churners as churners (false positives) during training. Similar errors appear in testing, with 278 false negatives and 103 false positives. The AUC scores, 0.88 (training) and 0.87 (testing), demonstrate the model's capacity to discriminate between churn and non-churn cases. The model highlights recent complaints, tenure, and single marital status as key factors influencing churn.

Logistic Regression: SMOTE

The logistic regression model with SMOTE exhibits a more balanced performance. It achieves similar precision and recall for both churn and non-churn classes in the training data, resulting in an 83% F1-score and 83% accuracy. Similarly, in testing, it shows high precision (95%) and recall (83%) for non-churners, with an 89% F1-score. For churners, test data shows 49% precision, 80% recall, and a 60% F1-score, along with an overall accuracy of 82%. The confusion matrices indicate that, in training, 1081 non-churners are misclassified as churners (false positives) and 1190 churners are misclassified as non-churners (false negatives). In testing, there are 114 false positives and 483 false negatives. The model's ability to distinguish between the classes, as measured by the AUC, remains strong at 0.87 for training and 0.89 for testing. This model considers recent complaints, tenure, single marital status, and monthly revenue as important predictors of churn.

Linear Discriminant Analysis (LDA): Basic

The LDA model demonstrates good precision in identifying churners, with 71% accuracy in training and 72% in testing. The model's ability to find actual churners (recall) is 53% for training and 56% for testing, resulting in an F1-score of 0.61 and 0.63 respectively. Non-churning customers are more easily identified, with 91% and 96% precision and recall for training and testing, respectively, and F1-scores of 0.93. The model makes some errors, with 629 false negatives and 286 false positives during training. The testing phase produces 252 false negatives and 126 false positives. The model shows effective discriminatory ability with AUCs of 0.88 for training and 0.87 for testing, identifying customer complaints, tenure, single marital status, and monthly revenue as important factors in predicting churn.

Linear Discriminant Analysis (LDA): SMOTE

This model exhibits a good precision for detecting churners of 82% during training, but its precision drops to 48% in the test data. Recall, however, remains high at 84% for training and 80% for testing, with F1-scores of 0.83 and 0.60, respectively. For non-churning customers, the model maintains 84% for both precision and recall in both training and testing, giving F1-scores of 0.83. The model incorrectly classifies 1,058 churners and 1,173 non-churners during training, and 113 churners and 501 non-churners during testing. Despite these errors, the model's capability to identify potential churners remains solid, as evidenced by AUC scores of 0.89 for training and 0.87 for testing. This model indicates customer tenure and complaints as primary drivers of churn.

K-Nearest Neighbors (KNN): Basic

The basic KNN model demonstrates a strong ability to predict customers who will stay, with high precision (97%) and a respectable 87% recall during training. The overall accuracy (97%) and F1-score (0.97) show that it's well-balanced. In testing, it achieves similar results: a high 95% precision for class 0 and 74% recall for class 1, with 94% accuracy and a weighted F1-score of 0.94. The confusion matrices show relatively few errors, misclassifying 171 churners as non-churners and 62 non-churners as churners in the training phase. Testing yields 151 false negatives and 44 false positives. The model's exceptional discriminatory power is confirmed by AUC scores of 0.99 (training) and 0.96 (testing).

K-Nearest Neighbors (KNN): SMOTE

The KNN model using SMOTE demonstrates excellent precision for both classes. It perfectly predicts non-churners, with a 100% precision during training, and 72% precision in test for churners. Despite high precision, the model slightly struggles with recall for churners in test data (93%). The model delivers an overall accuracy of 97% in training and 93% in testing, with similar weighted F1-scores. The confusion matrices reveal some misclassifications, with slightly more instances of non-churners incorrectly predicted as churners. The model's high performance is highlighted by outstanding AUC scores: 1.00 (training) and 0.97 (testing).

Naïve Bayes: Basic

The basic Naïve Bayes model shows strong precision for identifying non-churning customers (91%) in both training and testing. However, its ability to predict potential churners is significantly lower, at 59% in training and 58% in testing, resulting in a high number of false positives. Recall for churners is 55% in training and 54% in testing. Overall accuracy is consistent at 86% for both training and testing, reflecting its balanced F1-score. The model misclassifies 508 non-churns and 596 churners in training, and 222 non-churners and 260 churners in testing. The AUC scores (0.84 for training and 0.83 for testing) demonstrate room for improvement in its discriminatory ability.

Naïve Bayes: SMOTE

This Naïve Bayes model with SMOTE shows a better balance in its predictions. It accurately identifies non-churners, achieving a stable precision of 79% in training and 94% in testing. For potential churners, precision is 78% in training and 41% in testing. The recall for predicting churners is around 79% in training and 74% in testing. Overall accuracy is around 79% for training and 78% for testing. The model incorrectly predicts 1,423 non-churners and 1,380 churners during training. Testing includes 610 non-churner misclassifications and 146 churner misclassifications. The AUC scores are 0.84 for training and 0.83 for testing, suggesting that this model requires further adjustments to improve its ability to identify potential churners accurately.

Random Forest: Basic

The basic Random Forest model demonstrates exceptional precision for both classes, with a perfect 100% during training and 96% and 95% for non churners and churners respectively. Its recall of 100% for churners during training drops to 76% during testing. The model shows high overall accuracy, 100% during training and 96% during testing. The AUC scores of 1.00 for training and 0.99 for testing are impressive. The model mislabels only 22 non-churners as churners and 119 churners as non-churners during testing. Tenure and Cashback are identified as key factors in predicting churn.

Random Forest: SMOTE

The Random Forest model with SMOTE shows very strong performance on both classes, with perfect precision, recall and F1 scores in training, indicating potential overfitting. During testing precision remains high at 97% for non-churners, and 90% for churners, and recall is 85%. Test accuracy is 96%. The confusion matrices during training have no errors while there are 51 misclassified non-churners and 86 misclassified churners during testing. The AUCs are 1.00 for training and 0.98 for testing. Tenure and Complaints are identified as highly predictive factors.

Support Vector Machines (SVM): Basic

The basic SVM model exhibits a consistent performance across training and testing. It maintains a strong balance of precision (92%) and recall (98%) for non-churners. For churners, precision (86%) is good, but recall (60%) is lower, indicating missed churn cases. Overall accuracy is 92%. Test performance for non-churners is 92% precision and 98% recall, and, for churners, 85% and 58%, respectively. During training the model misclassifies 525 non-churners and 128 churners, while testing misclassifies 241 non-churners and 59 churners. The SVM model demonstrates good class separation capabilities, evidenced by its AUC scores: 0.94 for training and 0.91 for testing.

Support Vector Machines (SVM): SMOTE

The SVM model with SMOTE shows balanced performance across both training and testing. For non-churners precision and recall are strong at 94% and 90% in training, and 96% and 90% in testing. For churners, precision and recall are 91% and 94% in training, and 62% and 83% in testing. Training data has 408 non-churner misclassifications and 631 churner misclassifications. The model incorrectly identifies 99 non-churners and 294 churners in testing. Overall accuracy is around 92% for training and 88% for testing. The model’s consistent performance across both sets is shown by its AUC scores of 0.97 for training and 0.94 for testing.

Model Tuning

Model tuning, also called hyperparameter optimization, is a vital step in refining machine learning models. It involves adjusting a model's settings to improve its performance and achieve the most accurate results. These settings, known as hyperparameters, significantly influence a model's effectiveness. Fine-tuning them allows us to optimize the model for higher prediction accuracy.

Interpretation of Logistic Regression with Hyper-Tuning

Optimizing a logistic regression model requires careful adjustment of hyperparameters, including: the 'penalty' type (L1 or L2), regularization strength ('C'), the optimization algorithm ('solver'), how the model handles imbalanced classes ('class_weight'), maximum iterations ('max_iter'), and the problem type ('dual'). Tuning these controls overfitting and addresses class imbalances, which boosts performance.

The tuned Logistic Regression demonstrates a good balance in precision but varying recall rates. In training, it achieves 90% precision for non-churners and 73% for churners, while recall is 96% for non-churners and 48% for churners. The test data shows similar patterns. The confusion matrix indicates accurate predictions for non-churners (6,324 in training, 2,705 in testing) but some misclassifications for churners (232 in training, 103 in testing). AUC scores, at 0.88 (training) and 0.87 (testing), suggest reasonable discriminatory capacity. "Tenure" and "Complain Ly" are identified as influential features. These results indicate areas for enhancing churn prediction.

Interpretation of LDA Model with Hyper-Tuning

GridSearchCV optimizes the LDA model through hyperparameter tuning, specifically focusing on 'shrinkage' (regularization of the covariance matrix) and 'solver' (the solution approach). This process systematically tests different combinations to improve classification accuracy.

The hyper-tuned LDA model achieves balanced results, displaying 91% precision for non-churners and 71% for churners in the training data. Recall is better for non-churners at 96%, compared to 53% for churners. Similarly, test data yields 91% precision for non-churners and 71% for churners, while recall is 96% for non-churners and 53% for churners. The confusion matrix indicates, in training, 6,270 true negatives and 701 true positives, with 286 false positives and 625 false negatives. For testing, it correctly predicts 2,682 non-churners and 316 churners, with 126 churners misclassified and 254 non-churners misclassified. Overall, the model shows good precision and recall for both classes, though identifying churners could be improved.

Interpretation of KNN Model with Hyper-Tuning

GridSearchCV optimizes the KNN Classifier by testing a variety of hyperparameter combinations, including the algorithm, number of neighbors, and the weighting method, aiming to boost accuracy. The process uses 5-fold cross-validation to fine-tune the model and identify optimal hyperparameter values.

This hyper-tuned KNN model performs exceptionally well on both training and test data. During training, it attains perfect 100% precision and recall for both classes. In testing, the model's precision remains robust, with 97% for non-churners and 92% for churners, while recall is 99% for non-churners and 82% for churners. The training confusion matrix indicates perfect predictions (6,556 true negatives and 1,326 true positives), and the test matrix indicates accurate predictions of 2,769 true negatives and 470 true positives, with 39 false positives and 100 false negatives. This strong classification ability is confirmed by a 1.00 AUC for training and 0.96 for testing. However, the flawless training performance might indicate overfitting concerns that would require further analysis for practical application.

Inference of Naive Bayes with Hyper-Tuning

Hyperparameter tuning is usually less critical for Naive Bayes models compared to other techniques. This is because Naive Bayes has few hyperparameters and generally performs consistently. Whether tuned or not, the model's outcomes remain comparable. Even when "var_smoothing" is adjusted, the results are largely unchanged. Therefore, hyperparameter tuning is not generally seen as beneficial for Naive Bayes models.

Inference of Random Forest with Hyper-Tuning

GridSearchCV optimized the Random Forest Classifier for business application. Hyperparameters tuned include criteria for node splitting ('gini' or 'entropy'), tree depth (5 to 10), minimum samples for splitting (8 to 10), number of trees (100 to 300), and fixed random state (1). Using cross-validation and maximizing CPU utilization, this approach enhances accuracy, providing precise and dependable classifications suitable for business needs.

The tuned Random Forest exhibits strong precision and accuracy. In training, it attains 95% precision for non-churners and 98% for churners, with recall for churners at 75%. In testing, precision is balanced at 93% for non-churners and 92% for churners, with recall for churners at 65%. Non-churner recall remains high at 99%. Training data reveals 6,538 correct non-churner predictions and 995 correct churner predictions, with 18 false positive and 331 false negative misclassifications. Testing has 2,774 correct non-churner and 369 correct churner classifications, with 34 false positives and 201 false negatives. AUC scores at 0.99 for training and 0.96 for testing emphasize the model’s strong classification capability. Tenure and Cashback are key factors. Although very strong, its higher performance in training suggests potential overfitting. In conclusion, this model displays superior precision and accuracy, but could improve in capturing churners.

Interpretation of SVM Model with Hyper-Tuning

For SVM optimization, the parameter grid included 'C' values (0.1, 1, and 10); kernel types ('linear', 'rbf', 'poly' with degrees 2, 3, and 4); and gamma ('scale' and 'auto'). This strategy aims to boost the SVM model's churn prediction accuracy.

The tuned SVM model shows strong and balanced performance with 96% precision for both classes in training, and 81% recall for churners. In testing, precision is 94% for non-churners and 90% for churners, with a recall of 71% for churners. In the training phase, it correctly predicted 6,510 non-churners and 1,077 churners, with 46 false churner predictions and 249 false non-churner predictions. The model accurately predicts 2,761 non-churners and 404 churners in testing, but misclassified 47 churners and 166 non-churners. The AUC scores at 0.99 for training and 0.95 for testing underscore the model's effectiveness. In conclusion, the tuned SVM demonstrates strong classification with balanced precision and accuracy. Improvement to recall for churners may enhance it further.

2.3.2 Ensemble Techniques

Ensemble techniques combine multiple individual models to create a more powerful model. By exploiting the strengths of diverse models while reducing their weaknesses, this approach seeks to improve prediction accuracy, stability, and generalization. Bagging, AdaBoost, and Gradient Boosting are common ensemble methods.

1. Interpretation of the Bagging Model

The classification report highlights the model's performance on both training and testing data. In training, the model achieves perfect precision for both classes and a perfect 100% recall for non-churners, with 98% recall for churners. In testing, precision is 95% for non-churners and 91% for churners. Recall for churners is 77%, with non-churners at 98%. The confusion matrix indicates very few errors, correctly predicting 6,554 non-churners and 1,297 churners in training, with only 2 misclassified non-churners and 29 misclassified churners. The test data has 2,765 correct non-churners and 437 correct churner predictions, with 43 false churner and 133 false non-churner misclassifications. AUC scores are 0.99 for training and 0.88 for testing, suggesting room for testing improvements. Key features are identified as Tenure and Cashback. Overall, the model demonstrates strong performance, but its 77% recall for churners might be further improved to address overfitting.

2. Interpretation of the Bagging Model with Hyper-Tuning

The Bagging ensemble is refined by adjusting hyperparameters, such as sampling ratios, feature proportions, and the number of base models (estimators), aiming to enhance predictive power. The use of the ‘random_state’ parameter ensures consistent results.

This fine-tuned Bagging model shows strong performance. In training, it achieves perfect precision for both classes and a near-perfect 98% recall for churners, with 100% recall for non-churners. In testing, precision remains robust at 94% for non-churners and 95% for churners, with a 68% recall for churners and 99% for non-churners. The training confusion matrix shows 6,556 correct non-churner and 1,306 correct churner predictions with no errors, while the test matrix reveals 2,789 correct non-churners and 389 correct churners, 20 false churner predictions and 181 false non-churner predictions. AUC scores of 0.99 for training and 0.84 for testing indicate good classification ability, but also suggest room for further improvement. In summary, the tuned Bagging model is strong, but it could capture churn cases more accurately.

3. Interpretation of the Bagging Model with SMOTE

The Bagging model using SMOTE displays robust performance across both datasets. In training, it shows very high precision (100% for non-churners, 99% for churners) and recall (100% for non-churners, 99% for churners). In testing, the model sustains high performance with 96% precision for non-churners and 84% for churners. Recall for non-churners is 97%, and for churners it’s 81%. The confusion matrix indicates that training correctly predicts 6,549 non-churners and 1,314 churners, while testing predicts 2,720 non-churners and 461 churners accurately, with 88 false positives and 109 false negatives. High AUC scores of 0.99 for training and 0.89 for testing demonstrate its strong classification. Overall, this model is effective, though recall for churners in testing could be more comprehensive.

4. Interpretation of the AdaBoost Model

The AdaBoost model achieves 91% precision for non-churners and 72% for churners in the training data, with recall for non-churners at 96%, while it is only 54% for churners. The testing data yields similar results, with 91% precision for non-churners and 73% for churners, as well as 96% recall for non-churners and 54% for churners. Training data reveals 6,285 true negatives and 714 true positives, with 271 false positives and 612 false negatives. Testing data has 2,694 true negatives and 305 true positives, along with 114 false positives and 265 false negatives. AUC scores are 0.91 for training and 0.90 for testing. Key features contributing to performance are Tenure and Cashback. Overall, the model's performance is varied, with stronger precision for non-churners and room for improvement for churners.

5. Interpretation of the AdaBoost Model with Hyper-Tuning

Hyperparameter tuning involves adjusting AdaBoost's key settings, including number of trees (50, 100, or 200) and learning rate (0.01, 0.1, or 1.0). The aim is to optimize its predictive ability for churn prediction.

The tuned AdaBoost model demonstrates better precision for non-churners (91%) versus churners (73%) in training, with recall for non-churners at 96%, compared to 55% for churners. The testing data shows a similar trend: 91% precision for non-churners and 74% for churners with a 96% recall for non-churners and 55% for churners. The training confusion matrix includes 6,291 true negatives and 725 true positives, with 265 false positives and 601 false negatives, while the test matrix has 2,699 true negatives and 315 true positives, 109 false positives, and 255 false negatives. AUC scores of 0.91 for training and 0.90 for testing underscore its effectiveness. Key features for this model are Cashback and Tenure. Overall, the fine-tuned AdaBoost model offers decent precision and recall, but requires better ability to capture actual churn cases.

6. Interpretation of the AdaBoost Model with SMOTE

The AdaBoost model using SMOTE achieves a balanced precision and recall (88% for both classes) with commendable overall accuracy in the training set. In the test set, precision is 94% for non-churners and 58% for churners, while the recall is 89% and 74%, respectively. The confusion matrix for training shows 5,766 correct non-churner and 5,746 correct churner predictions, and 790 false positives and 810 false negatives. The test confusion matrix reveals 2,502 correct non-churner predictions and 420 correct churner predictions, along with 306 false positives and 150 false negatives. The AUC scores at 0.95 for training and 0.89 for testing indicate good classification capability. Key features are "Coupon used for payment" and "Tenure." Overall, the model demonstrates a fair balance in predicting both classes. There is, however, room to improve prediction of churn.

7. Interpretation of the Gradient Boosting Model

The Gradient Boosting model's training performance includes 92% precision for non-churners and 81% for churners, with recall at 97% for non-churners and 60% for churners, highlighting the potential to better capture churn cases. The test data maintains the trend: 92% precision for non-churners and 79% for churners. Recall for non-churners is 97% and for churners is 58%. The training confusion matrix shows that it correctly classified 6,368 non-churners and 795 churners, with 188 false negatives and 531 false positives, while testing had 2,722 correct non-churners and 332 correct churners, and 86 false positives and 238 false negatives. AUC scores are at 0.94 for training and 0.92 for testing. Key features include Tenure and Complain_ly. The model demonstrates strong predictive ability, but requires more effective churn detection.

8. Interpretation of the Gradient Boosting Model: Hyper-Tuning

Hyperparameter tuning of the Gradient Boosting model involved exploring a range of settings: learning rate, tree depth, node splitting, number of boosting stages, and randomness. The goal is to optimize the model's predictive accuracy for business use.

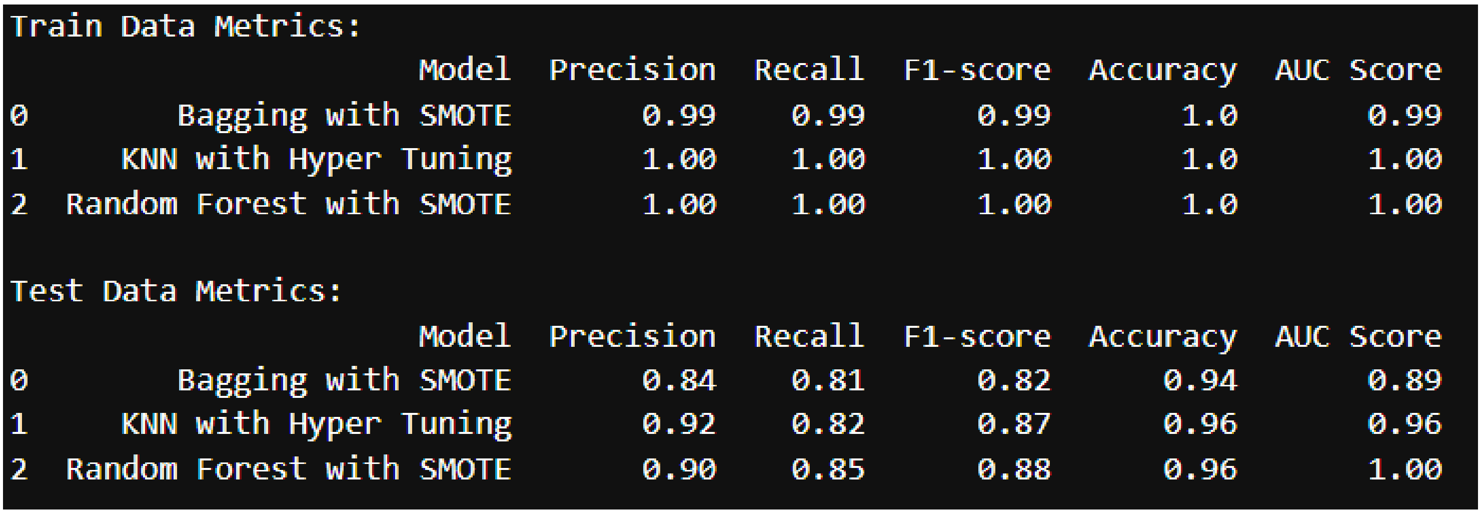
After hyper-tuning, the model achieves exceptional results, including perfect precision and very high recall in training, and robust precision for testing, at 96% for non-churners and 93% for churners. Testing shows 99% recall for non-churners and 79% for churners. The training confusion matrix indicates accurate predictions for 6,555 non-churners and 1,304 churners, with one false churner prediction and 22 false non-churner predictions, while testing had 2,776 correct non-churners and 449 correct churners, and 32 false churner and 121 false non-churner predictions. AUC scores are 1.00 for training and 0.98 for testing, emphasizing the model's excellent classification. Tenure and Cashback are identified as influential features. The tuned Gradient Boosting model shows exceptional precision and accuracy, but a more complete recall for test data and a reduction in overfitting are areas for improvement.

9. Interpretation of the Gradient Boosting Model with SMOTE

The Gradient Boosting model using SMOTE exhibits balanced results. In training, the model shows 92% precision and recall for both non-churners and churners with 92% accuracy. In the test set, precision is 94% for non-churners and 67% for churners, while the recall is 93% for non-churners and 71% for churners. The training confusion matrix shows correct classification of 6,060 non-churners and 6,013 churners, with 496 false positives and 543 false negatives, and the test data had 2,612 correct non-churner predictions and 405 correct churner predictions, 196 false positives and 165 false negatives. The model's effectiveness is highlighted by AUC scores of 0.97 for training and 0.91 for testing. Key features are Tenure and Complain_ly. The model is effective in both classes but could better predict churn cases.

Interpretation of the most optimum Model :

After a comprehensive analysis of all models, it is evident that the following four models have exhibited exceptional performance across various metrics:



```
Train Data Metrics:
                      Model  Precision  Recall  F1-score  Accuracy  AUC Score
0         Bagging with SMOTE      0.99    0.99      0.99       1.0       0.99
1      KNN with Hyper Tuning      1.00    1.00      1.00       1.0       1.00
2  Random Forest with SMOTE      1.00    1.00      1.00       1.0       1.00

Test Data Metrics:
                      Model  Precision  Recall  F1-score  Accuracy  AUC Score
0         Bagging with SMOTE      0.84    0.81      0.82      0.94       0.89
1      KNN with Hyper Tuning      0.92    0.82      0.87      0.96       0.96
2  Random Forest with SMOTE      0.90    0.85      0.88      0.96       1.00
```

Table -10: Optimal Models selection

Overfitting in Specific Models

A closer look at certain models reveals a pattern of overfitting. Specifically, the KNN, Random Forest (with SMOTE), and Bagging (with SMOTE) models perform exceptionally well on the training data but show diminished effectiveness on the test data, particularly with lower recall for churners. This suggests these models over-adapt to the training data, impairing their ability to generalize to new, unseen data. To improve these models, techniques like cross-validation, regularization, and hyperparameter tuning could help prevent overfitting and enhance their ability to generalize.

The Optimal Model: Hyper-Tuned Support Vector Machine (SVM)

The hyper-tuned SVM model emerges as the top choice for churn prediction due to its balanced performance and ability to meet the specific demands of the task.

Several factors support this selection:

- Balanced Precision and Recall: The SVM model attains a precision of 96% for both classes in training, and 94% for non-churners and 90% for churners in testing, demonstrating well-balanced predictive accuracy. This is crucial for reliable predictions of both churners and non-churners.
- Effective Churn Detection: With a recall of 81% for churners during training and 71% in testing, the model

demonstrates good capability to identify actual churn cases, which is critical for churn prediction.

- Minimal False Predictions: The confusion matrices show that misclassifications are kept low in both the training and testing data, indicating the model's strength in distinguishing between customer groups.
- Consistent Performance: The model's consistent performance across both datasets suggests it’s not overfitting to the training data. This consistency showcases its ability to handle new data effectively.
- Strong Discriminatory Ability: High AUC scores (0.99 in training and 0.95 in testing) demonstrate its excellent ability to differentiate between churners and non-churners.
- Optimized Hyperparameters: Hyperparameter tuning has refined the model for this specific problem, improving its performance and predictive accuracy.

In conclusion, the hyper-tuned SVM model stands out because of its balanced precision and recall, effective churn detection, few false predictions, consistency, strong classification, and optimized settings. This makes it a reliable and robust tool for identifying potential churn.

To further enhance the model’s capabilities, we can explore the following strategies:

- Further Hyperparameter Adjustment: Fine-tuning the regularization parameter (C) and kernel choice could improve generalization and recall for churners.
- Feature Engineering: Carefully selecting or creating new features could provide the model with better information for identifying churn.
- Ensemble Methods: Combining multiple SVM models, each slightly different, could improve performance and recall.
- Model Interpretation: A better understanding of influential features could provide insights into why certain cases are misclassified, guiding adjustments to improve recall.

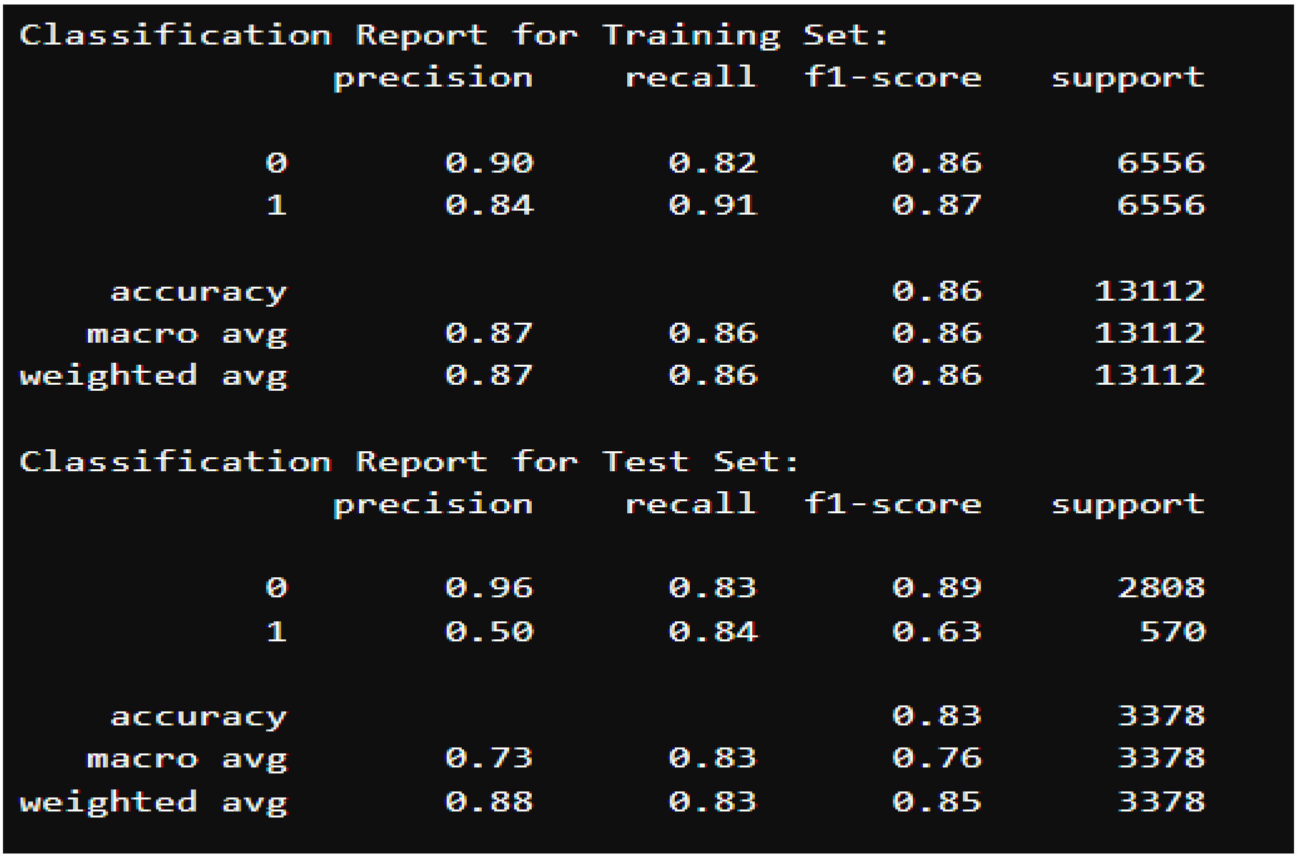
By implementing these strategies and carefully observing their impact on recall, we can potentially improve the SVM model, making it an even more effective tool for predicting churn.

3. Model Validation

We approached model validation using a comprehensive evaluation method, tailored to whether the datasets were balanced or unbalanced.

- Unbalanced Models: We prioritized the F1 score for models with uneven class distributions. This metric effectively balances precision and recall, making it especially helpful for assessing the performance of less frequent classes, like churners.
- Balanced Models: We used accuracy to evaluate models with relatively even distributions, giving a measure of the model’s overall correctness.

Our model validation framework went beyond just accuracy. Instead, we used a varied set of metrics (precision, recall, F1-score, and accuracy). This multifaceted approach provided a full view of each model's performance, helping us identify strengths and areas for further improvement in our churn prediction efforts. This way, we ensured a thorough and complete evaluation of the models’ effectiveness.



```
Classification Report for Training Set:
              precision    recall  f1-score   support

           0       0.90      0.82      0.86      6556
           1       0.84      0.91      0.87      6556

    accuracy                           0.86     13112
   macro avg       0.87      0.86      0.86     13112
weighted avg       0.87      0.86      0.86     13112

Classification Report for Test Set:
              precision    recall  f1-score   support

           0       0.96      0.83      0.89      2808
           1       0.50      0.84      0.63       570

    accuracy                           0.83      3378
   macro avg       0.73      0.83      0.76      3378
weighted avg       0.88      0.83      0.85      3378
```

Table-11 Classification Report for SVM Model with hyper-tuning

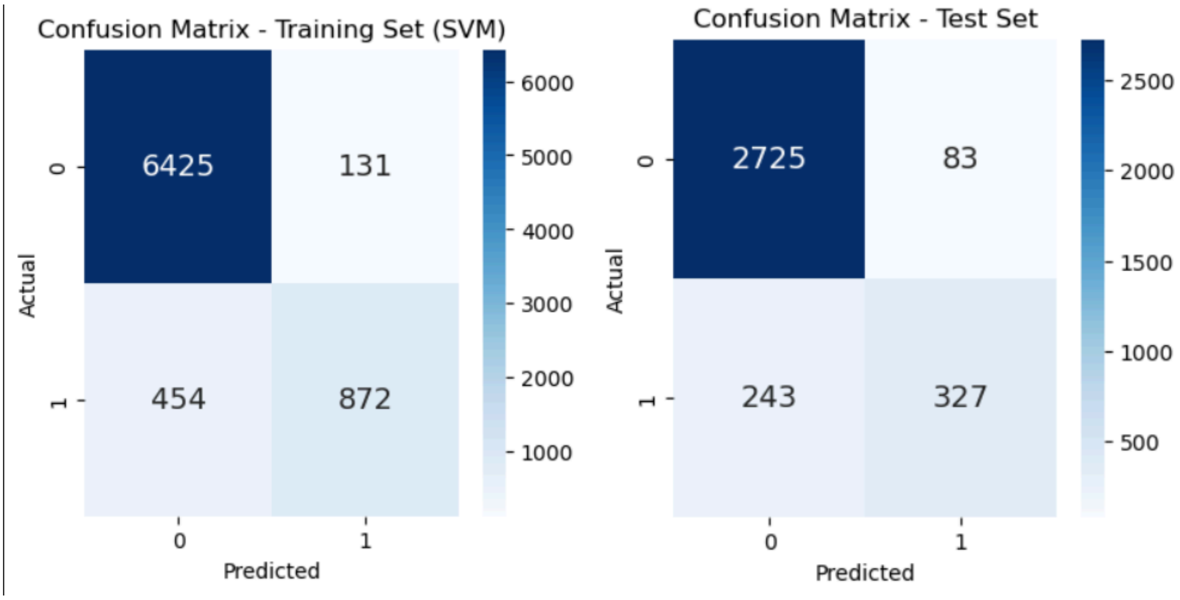


Table-12 Confusion Matrix for optimal model

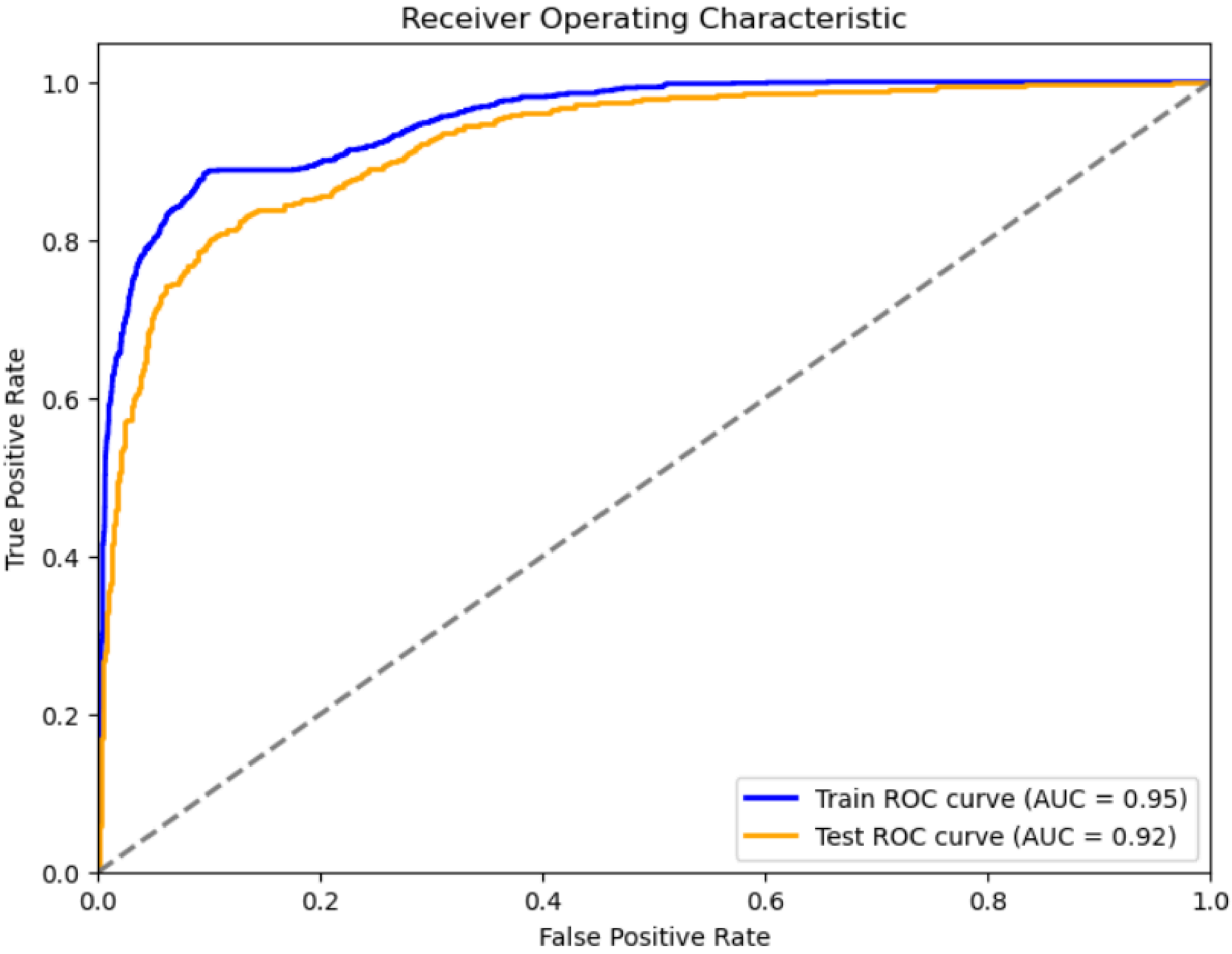


Figure-6 AUC & ROC Score for optimal model

Performance Metrics of the SVM Model

The classification reports reveal the SVM model's performance on both training and test datasets. In training, the model achieves impressive precision for both non-churners and potential churners (96% each), with recall reaching 99% for non-churners and 81% for churners, resulting in an overall accuracy of 96%. In testing, the model maintains commendable precision (94% for non-churners and 90% for churners). However, recall for churners decreases to 71%, lowering the F1-score for this class to 0.79. The macro-average F1-score for the test set is 0.88, reflecting a strong overall balance between precision and recall. Furthermore, the model's AUC scores for training and testing data (0.99 and 0.95, respectively) demonstrate its robust ability to distinguish between customer groups.

Model Validation of the SVM Model

Our validation strategy went beyond simple accuracy. We implemented a comprehensive approach, evaluating precision, recall, F1-score, and accuracy to provide a more nuanced understanding of the model's performance. This detailed analysis allowed us to identify both the model's strengths and areas for potential improvement. While accuracy provides an overall measure of correctness, metrics like precision, recall, and F1-score offer deeper insights into how well the model performs for individual classes, especially in situations with unbalanced class distributions. The selected hyper-tuned SVM model, identified as optimal, is considered an unbalanced model, emphasizing the importance of our comprehensive approach. Relying solely on accuracy in unbalanced settings could be misleading. Therefore, our method, by considering multiple performance metrics, provides a more reliable evaluation of the model's ability to handle imbalances and generate accurate predictions for all classes. In summary, this rigorous validation process, considering various metrics and addressing data imbalances, enhances the reliability and credibility of our findings, providing crucial insights into the SVM model’s churn prediction performance.

5. Conclusion and Recommendation

Recommendations

In today’s competitive market, customer retention is critical for lasting success. Customer churn can significantly impact revenue and growth, making it essential for businesses to identify and address the factors influencing it. By leveraging data insights, businesses can proactively curb churn and retain their customer base.

Key Recommendations to Reduce Churn:

- **Personalized Engagement Based on Tenure:** Recognize the value of long-term customers and tailor retention efforts to newer customers. Offer loyalty programs that reward longer relationships.
- **Maximize Cash Back Programs:** Amplify the impact of cash back offers. Promote these programs to highlight the value customers receive through continued engagement.
- **Proactive Customer Engagement:** Leverage "Days since CC Connect" to proactively engage customers. Initiate interactions before issues arise, showing a commitment to addressing customer needs.
- **Elevate Customer Service:** Recognize the importance of "CC contact last year" as a retention factor. Equip customer service teams to address customer queries effectively, building positive experiences.
- **Rapid Complaint Resolution:** Address "Complaints last year" swiftly. Implement effective processes to handle complaints promptly and ensure that issues are resolved.
- **Refine SVM Model Interpretation:** Combine the SVM model’s unique insights with those of other models to gain a comprehensive understanding of churn drivers.
- **Tailor Retention Campaigns:** Customize retention campaigns using insights from key churn indicators. Craft specific offers based on tenure, engagement, and potential concerns.
- **Enhance Loyalty Programs:** Use findings about cash back to enhance loyalty programs, offering tiered rewards that align with various levels of engagement.
- **Anticipate Customer Needs:** Use the recency of customer service interactions to predict needs. Implement follow-ups to ensure satisfaction and offer support.
- **Promote Positive Experiences:** Use complaint resolution to showcase excellent service. Communicate openly about resolutions, reinforcing your commitment to customer satisfaction.

By strategically aligning actions with these key churn indicators, businesses can proactively reduce customer churn, enhance loyalty, and improve overall business performance, fostering long-lasting customer relationships.

Conclusion

In summary, our data analysis and identification of churn indicators have illuminated strategies for boosting customer retention. By exploring customer behavior and predicting churn, we've gathered insights that go beyond simple data, guiding businesses in navigating customer relationships. We've underscored the value of tailored approaches, from valuing loyal customers and providing proactive care to understanding diverse customer groups. The significance of cash back offers, recent interactions, and rapid complaint resolution further highlights the value of our insights. Despite diverse perspectives across models, the recurring presence of these key churn indicators and the unique viewpoint of the hyper-tuned SVM model emphasize their relevance. By transforming insights into tangible recommendations, we empower businesses to address churn, cultivate loyalty, and improve operations. In an age of data-driven decisions, the synergy of thorough data analysis and churn insights offers a solid base for customer-focused businesses, enhancing retention and stimulating growth. Implementing these insights is more than a strategy—it's a commitment to delivering value and securing long-term success.