greatlearning
Power Ahead

CAPSTONE

DTH CHRUN PREDICTION

FINAL REPORT

Mohamed Rifaz Ali K S

PGP-DSBA Online

July' 22

greatlearning
Power Ahead

Contents

List of Tables & Figures

1. Introduction

Problem Statement

A Direct to Home (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 the potential churners. In this company, account churn is a major thing because **1 account can have multiple customers**. Hence by losing one account the company might be losing more than one customer.

Need of the study

Churn prediction means detecting which customers are likely to leave a service or to cancel a subscription to a service. It is a critical prediction for many businesses because acquiring new clients often costs more than retaining existing ones in telecommunications industries. Predicting when a client is likely to leave and offering them discounts, promotions, and combo offers to stay can offer considerable savings to a business.

2. EDA and Business Implication

- There are 11260 rows and 19 columns are present in the customer churn dataset
- Independent variables: 5 Categorical & 12 Continuous variables. Churn is the target variable for this problem. It will be used as a response variable when we do machine learning classification algorithms.
- The primary drivers of the dataset of the customers are account id, churn flag, Tenure, City Tier, Payment method, services offered, agent score, revenue growth, cashback, gender, marital status, any complaints raised by customers, account type, and login device.
- At the first sight itself, we are able to notice that there are some special characters present in the columns.
- There are no duplicate records present in the dataset.

Churn (target Variable)

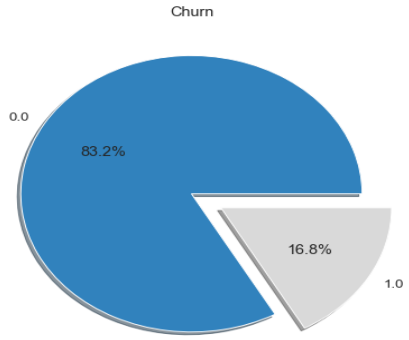

Figure 1.Proportion of Churn variable

Data seems to be highly imbalanced with only 17% of 1's and 83% of 0's in the target variable.

#	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

Missing values

Table 1.Customer Churn Data Info

Univariate analysis

There are just 1.25% of the missing values present in the entire dataset. So, we perform the Univariate and Bivariate analyses after treating the anomalies and imputing the missing values in the dataset. We will be discussing this in detail. Please refer Outlier Section and Missing Value treatments.

Let us understand the distribution of continuous variables data through histograms.

Continuous Variables

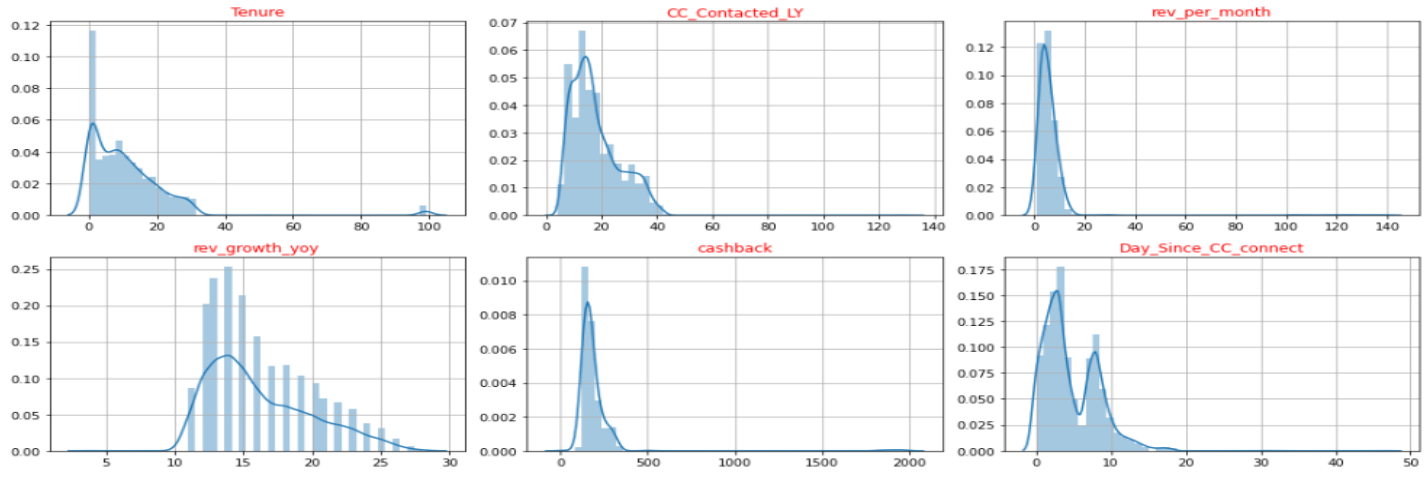

Figure 2. Data Distribution of Continuous variables

Skewness

	skew
Tenure	3.916348
CC_Contacted_LY	1.427159
rev_per_month	9.317431
rev_growth_yoy	0.752475
cashback	8.848078
Day_Since_CC_connect	1.280300

Fairly Symmetrical	-0.5 to 0.5
Moderate Skewed	-0.5 to -1.0 and 0.5 to 1.0
Highly Skewed	< -1.0 and > 1.0

Table 2.Skewness Table

- The variables such as 'Tenure','CC_Contacted_LY','rev_per_month', 'cashback','Day_Since_CC_connect' are highly skewed.
- The variable 'rev_growth_yoy' is moderately skewed.
- Most of the data's are clustered around the left tail of the distribution while the right tail of the distribution is longer.

We can classify the ***'coupon_used_for_payment'*** as the discrete variable because 87% of the coupons used by the customers are consitituted from the initial 4 values 0,1,2 and 3 itself. Also, the maximum times the coupon used by the customers to do the payment for the last 12 months is just 16.

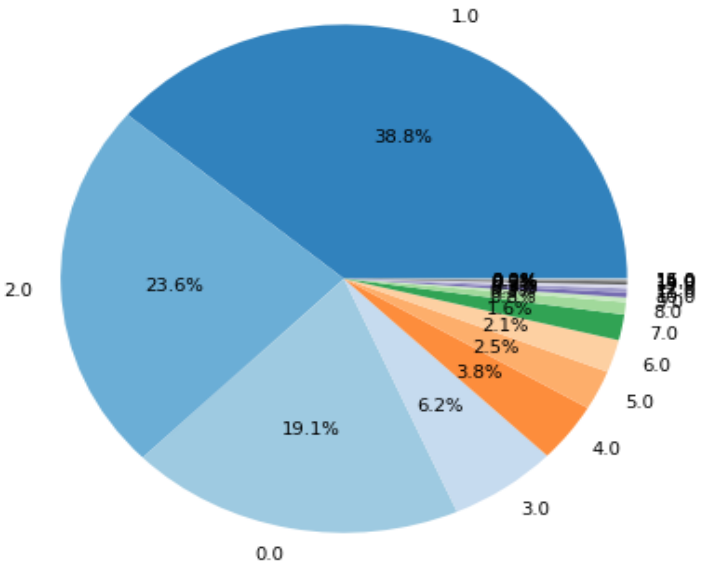

Figure 3.Proportion of coupon_used_for_payment variable

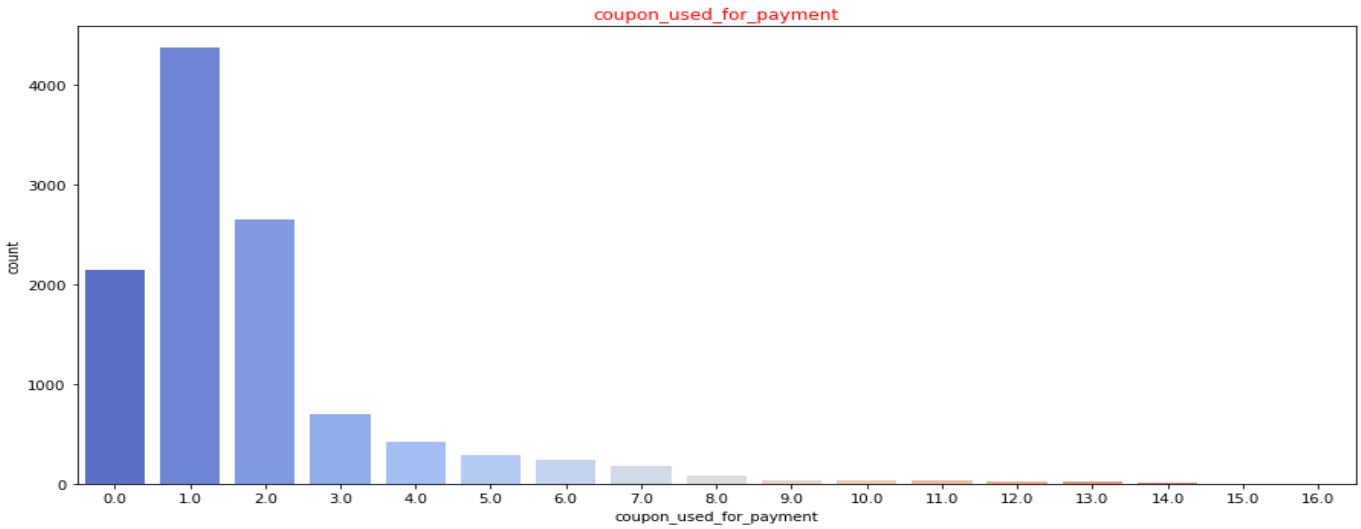

Figure 4. Distribution of coupon_used_for_payment variable

Coupon used for Payment –For the last 12 months,predominantly, the customers have used the coupons only once, twice, thrice or never used coupons to do the payment.

Discrete Variables

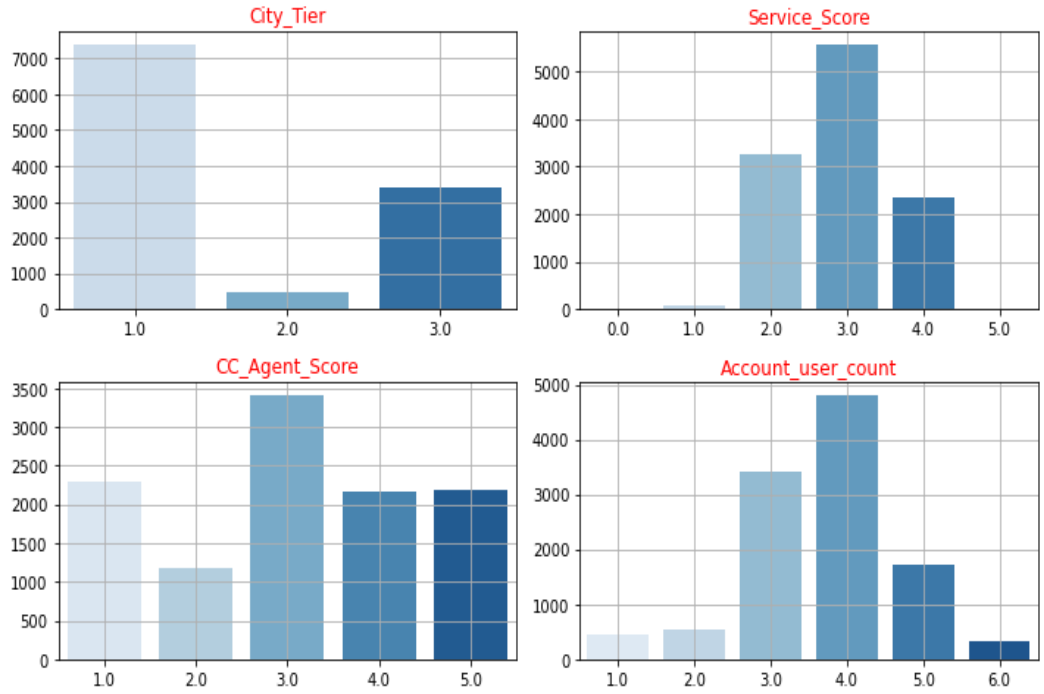

Figure 5.Countplots for Discrete Variables

- **City Tier** – The primary Customer's city of tier 1 count is high and then followed by 3. Tier 2 is very low.
- **Service Score** – Mostly, the satisfaction score given by customers of the account on service provided by the company is moderate. Customers are rarely given a very poor or very good rating.
- **CC Agent Score** – The satisfaction score given by customers of the account on customer care service provided by the company are spread out from low to high.
- **Account User Count** – Most of the times, the number of customers tagged with the account is 3,4 and 5. Rarely with 1, 2 and 6 user count.

Categorical Variables

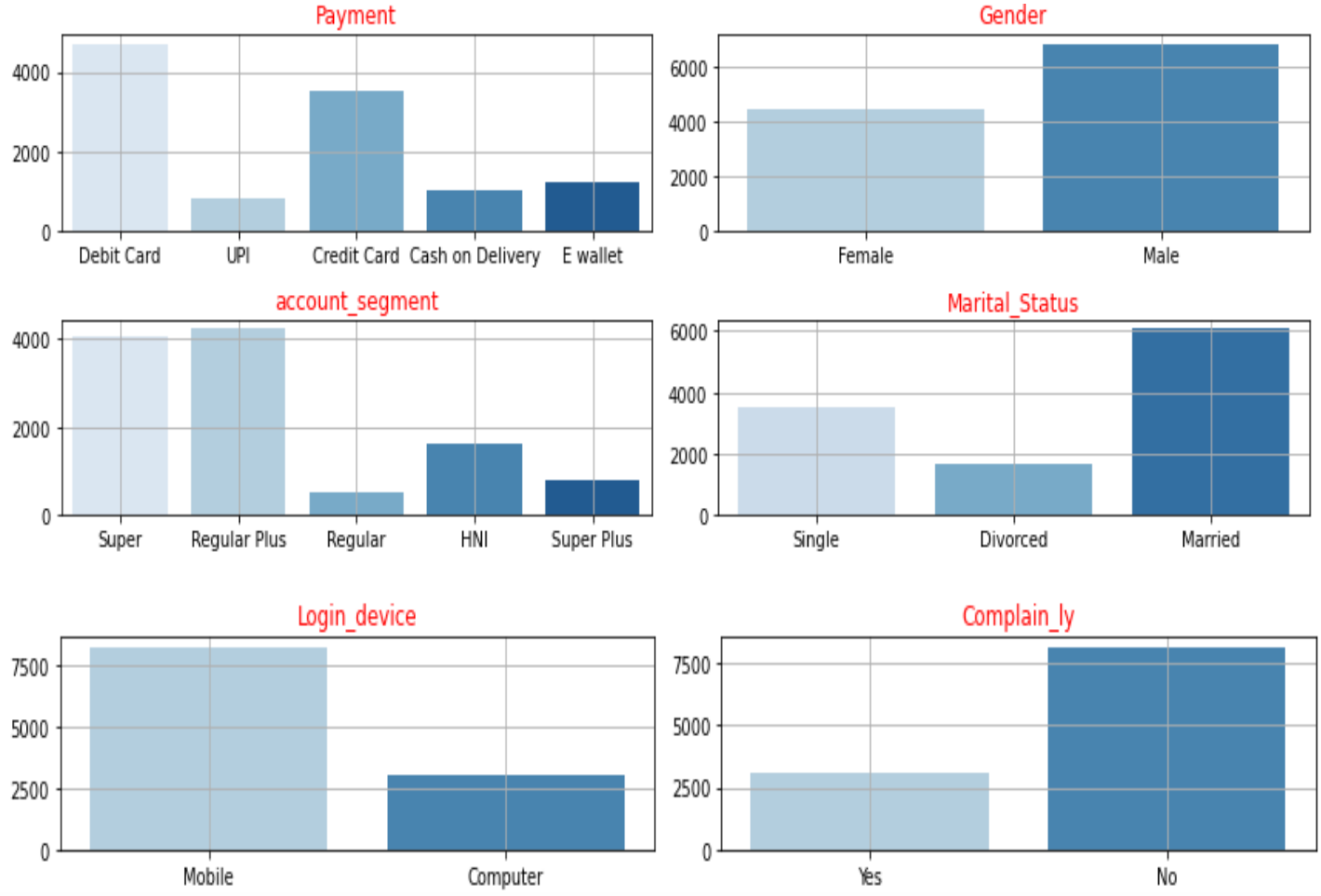

Figure 6.Countplots for Categorical Variables

- **Payment** – Customers preferred to use Debit cards and Credit Cards to do the payment.
- **Gender** – The ratio of male to female customers 60:40.

- **Account Segment** – Based on the spending, the Super and Regular plus account segments are more. We have also got a good amount of High Network Individuals.
- **Marital Status** – Primarily, Married people used the DTH the most and then followed by single and divorced.
- **Login Device** - Preferred login device of the customers in the account (Mobile) is 73% and the Computer is 27%.
- **Complain_ly** – Complaints raised by the customers for the last 12 months is 23% and not raised is 73%.

Bivariate/Multivariate analysis

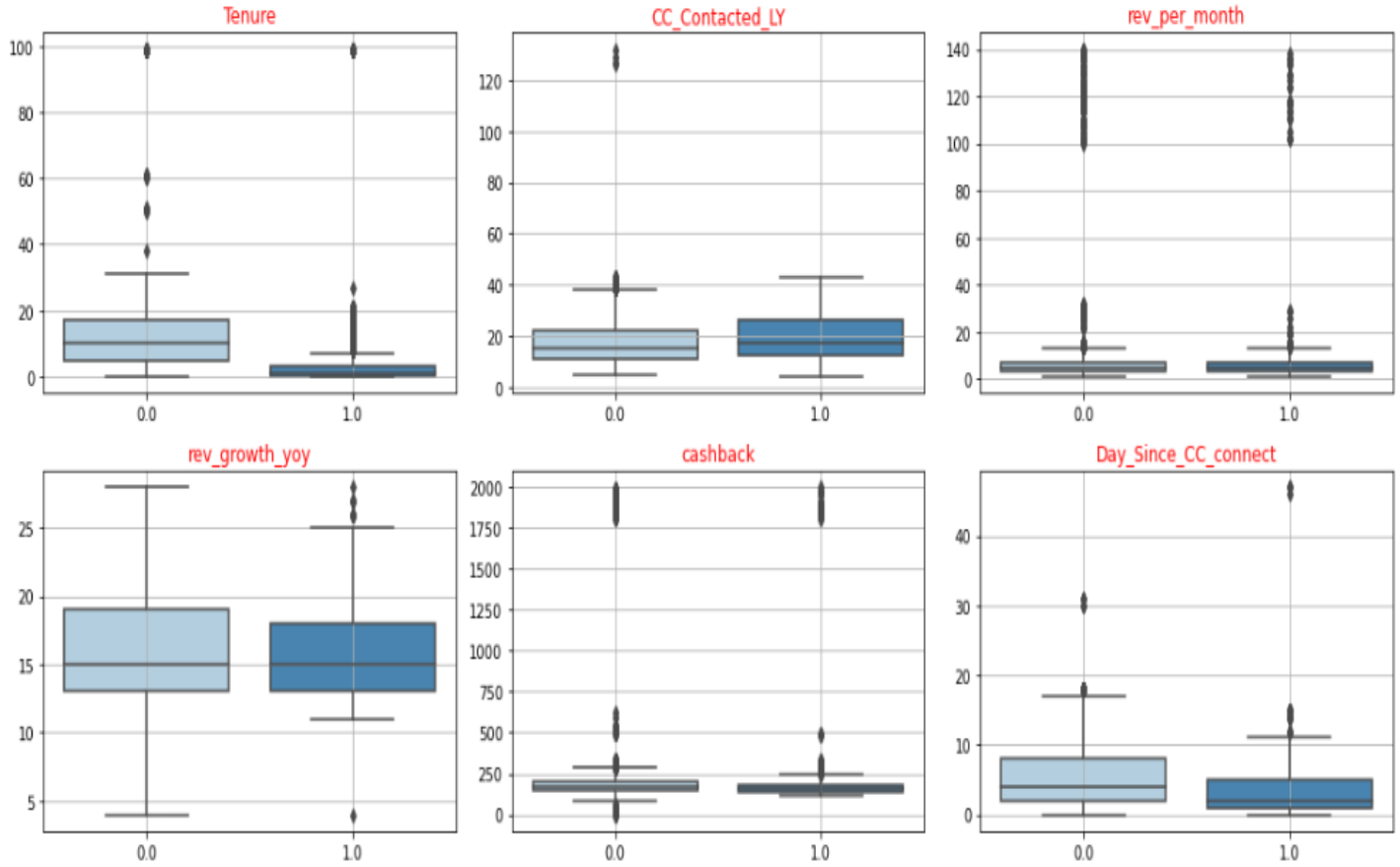

Figure 7.Churn Vs Continuous variables plots

- The customers stay longer in the contract, their churn rate is less.
- The number of times the account customers have contacted customer care in the last 12 months is increased, and their churn rate is also increasing.
- The revenue growth percentage of the non-churn customers is a bit high compared to the churn customers.

- There is almost similar revenue per month and cashback offers received by both non-churn and churn customers.

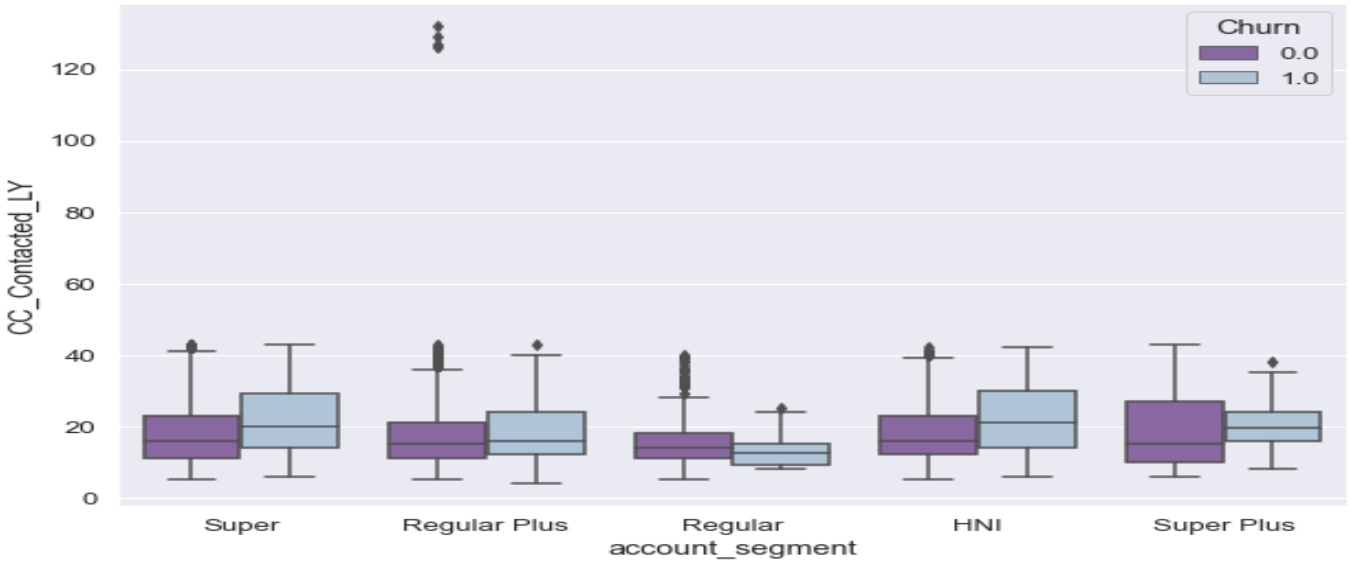

Figure 8.Plot of Account Segment Vs Customer Connect with Churn

- The number of times the account customers have contacted customer care in the last 12 months is high for HNI, Regular Plus and Super account type, and their churn rate is also high.
- For the Regular account type, the times they contacted the customer care is less and their churn rate is low.

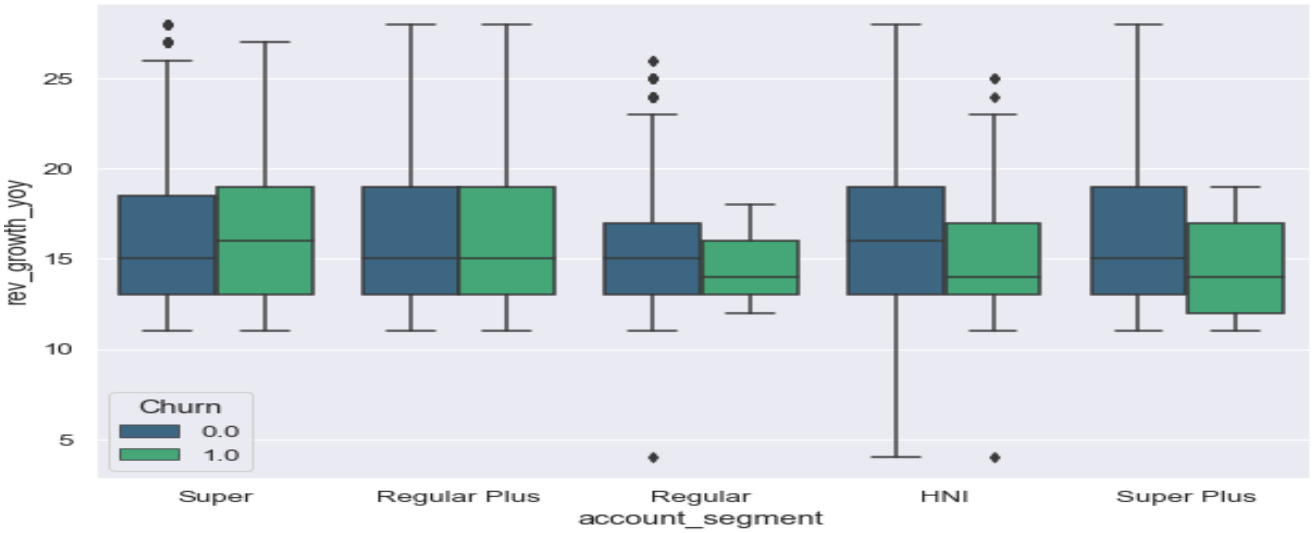

Figure 9. Plot of Account Segment Vs Revenue growth yoy with Churn

- The percentage of revenue growth of Regular Plus, Super, and Super Plus is increased, and their churn rate is also increased.

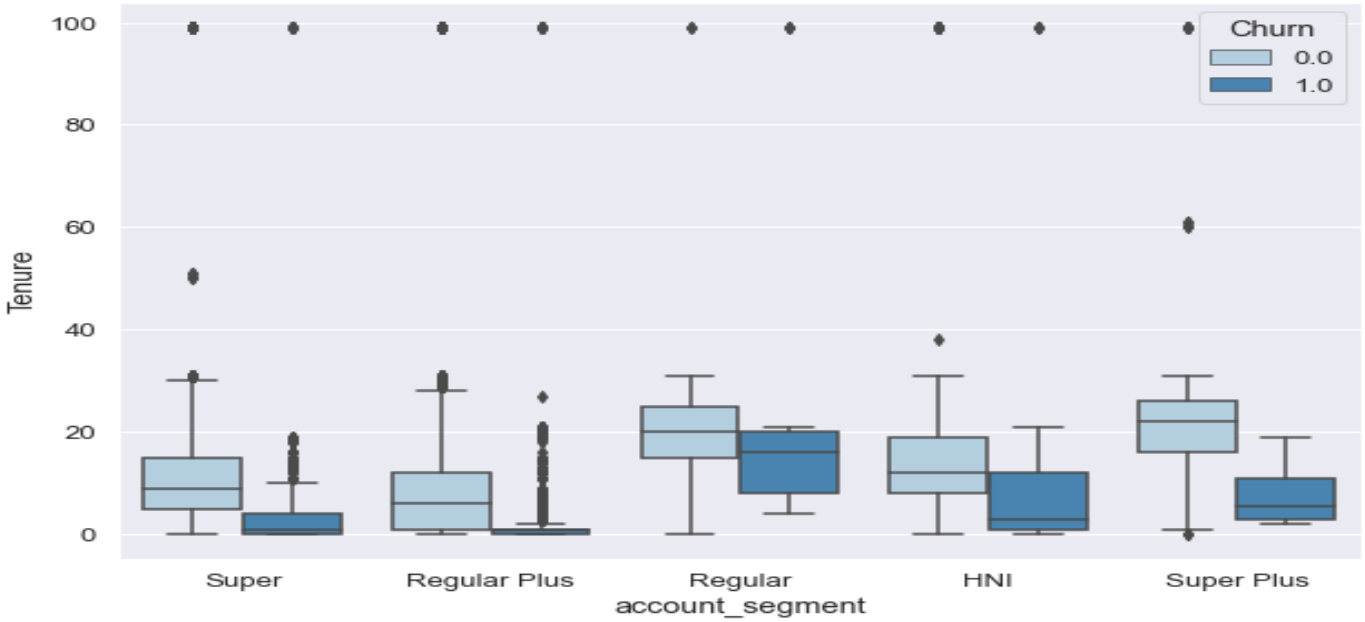

Figure 10. Plot of Account Segment Vs Tenure with Churn

- The churn rate for the 'Regular' account type is high. Interesting to note here is that, when the customers upgraded to 'Regular Plus', the average churn rate is reduced.
- We see that the churning rate of HNI is emerging when the tenure is increasing.
- The 'Super Plus' account type customers stay longer in the contract and their churn rate is also high.

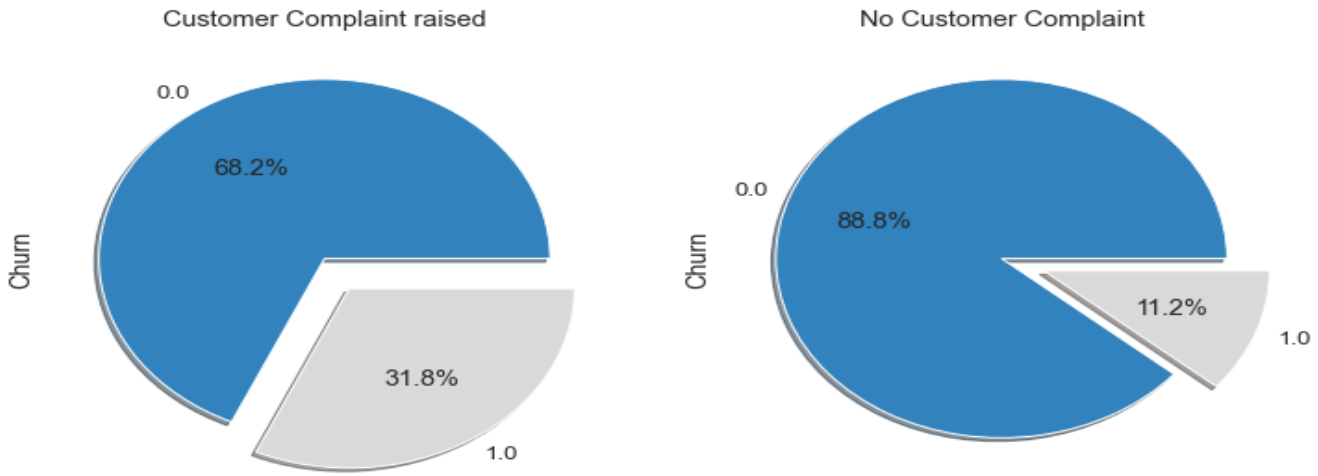

Figure 11.Proportion of Customer Complaints with Churn

- The churn rate of the customers have raised the complaint is 32% and those have not raised it just with 11%.

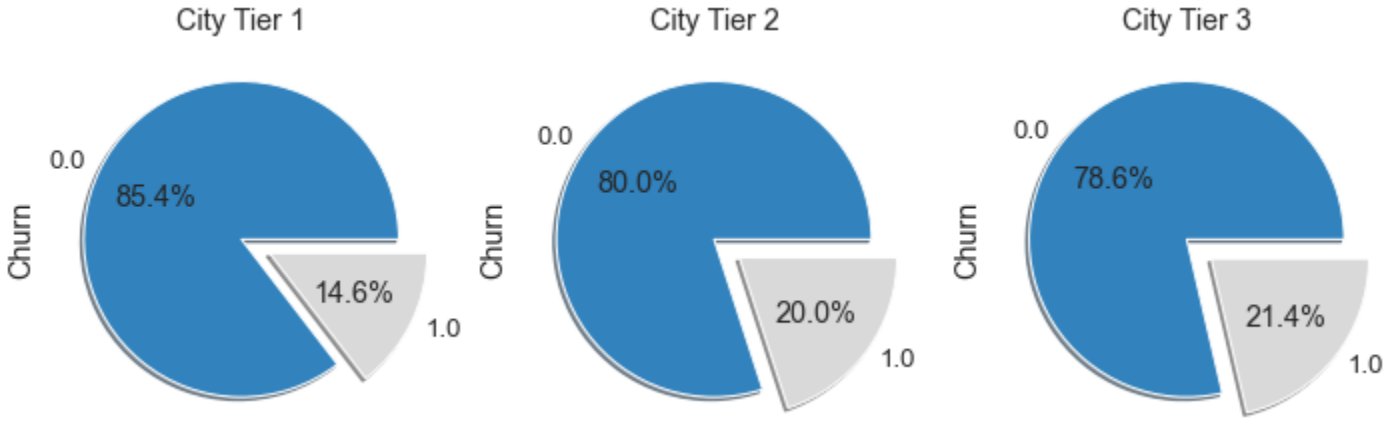

Figure 12.Proportion of City Tier with Churn

- The churn rate of City Tier is at the top with 21% and then with Tier 2 with 20% and then with Tier 1 with 15%.

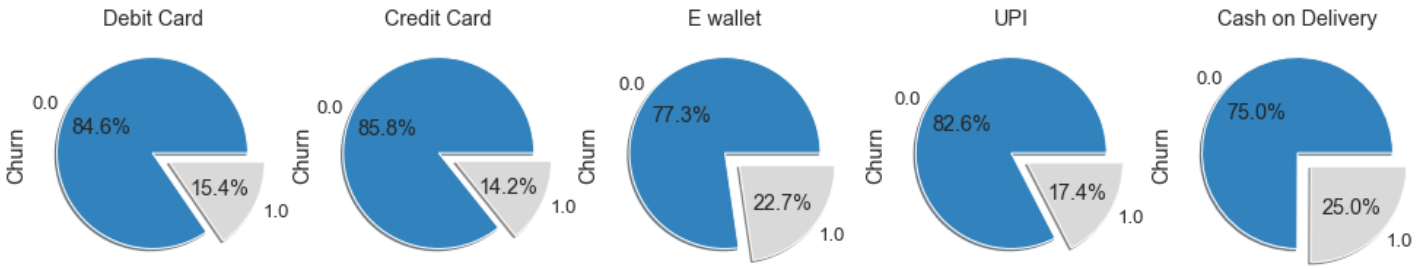

Figure 13.Proportion of Payment type with Churn

- The Churn rate of the customers using COD as the payment is 25% which is quite high and then with E-wallet with 23%, then UPI with 17.4%, then Debit & Credit card around 15%.

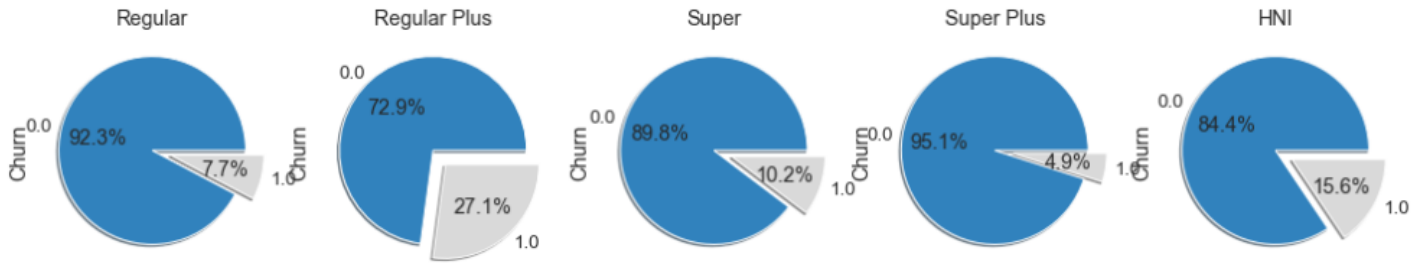

Figure 14.Proportion of Account Segment with Churn

- The churn rate of the Regular Plus account type is with 27% which is quite high and then HNI customers with 16%.Other customers are just under 10%.

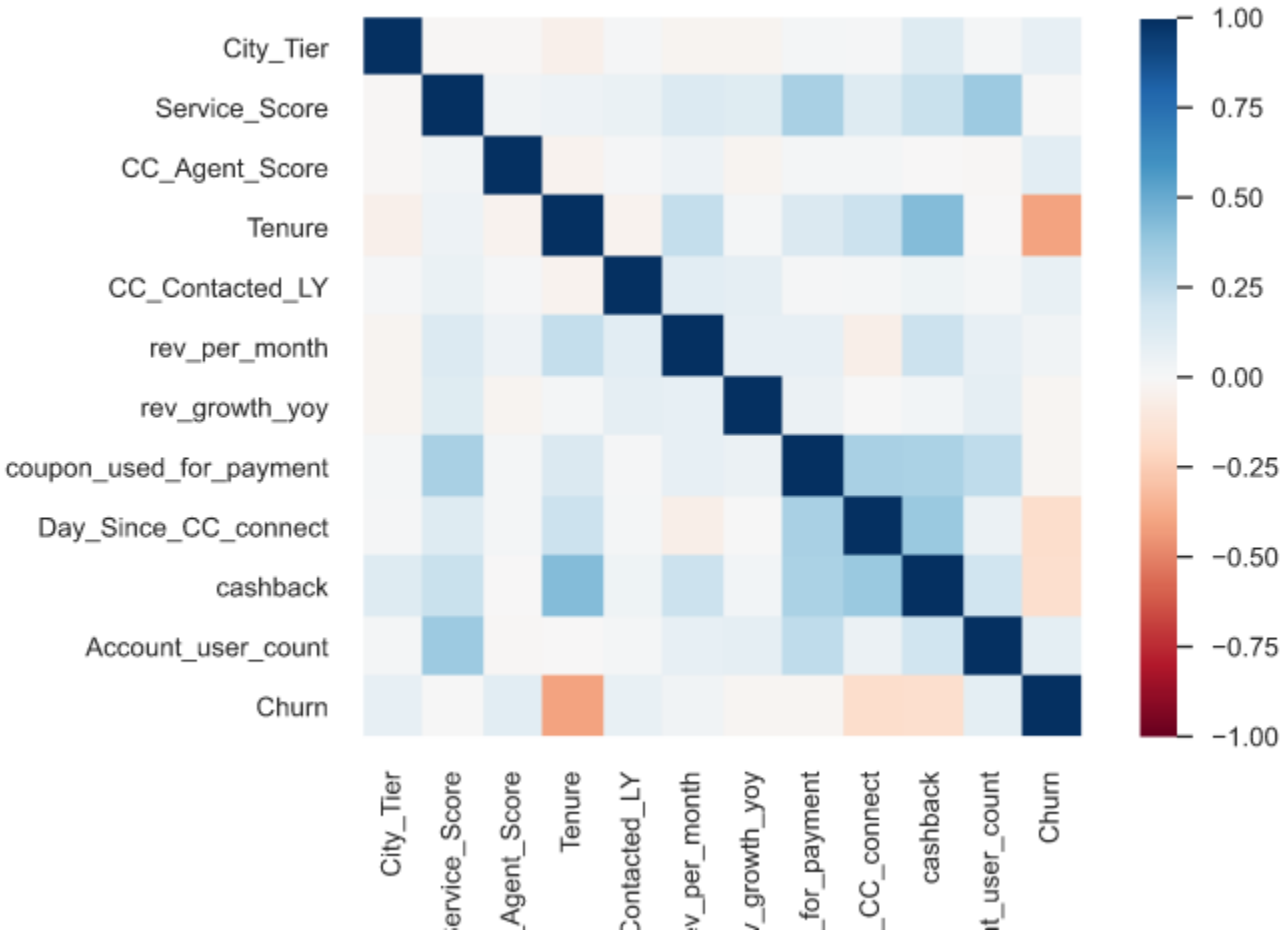

Figure 15.Correlation of Customer Churn Data

- The variable Churn and Tenure is negatively correlated. i.e., the shorter the customer stays, the higher the churn, and the longer they stay and lower the churn rate.
- DTH company provided more **cashback** offers to the customers staying for a long time. So, cashback and Tenure are somewhat highly correlated.
- The Account user count is correlated with the Service Score variable. i.e., if the user count is high in the account, the score provided to the company will be higher in numbers too.
- Day_Since_CC_connect, cashback variables are negatively correlated with the Churn variable. If they get more cashback offers, they would like to stay in the contract and if they frequently contacted customer care, their churn rate would get increased.

3. Data Cleaning and Pre-processing

Treating Data Anomalies

There are special characters like (#, +, $, @, *, &&&&) present in the dataset and can be removed with null values (np.nan). ‘Super +’ and ‘Regular +’ can be replaced with 'Super Plus' and 'Regular Plus' respectively in account segment column. 'F' and 'M' can be replaced with 'Female' and 'Male' in Gender column.

Inspect all the variables to see if there are redundant data are still present.

Variables	Unique Counts	Unique Values	Missing Value Counts
Tenure	37	[4. 0. 2. 13. 11. nan 9. 99. 19. 20. 14. 8. 26. 18. 5. 30. 7. 1. 23. 3. 29. 6. 28. 24. 25. 16. 10. 15. 22. 27. 12. 21. 17. 50. 60. 31. 51. 61.]	218
City_Tier	3	[3. 1. nan 2.]	112
CC_Contacted_LY	44	[6. 8. 30. 15. 12. 22. 11. 9. 31. 18. 13. 20. 29. 28. 26. 14. 10. 25. 27. 17. 23. 33. 19. 35. 24. 16. 32. 21. nan 34. 5. 4. 126. 7. 36. 127. 42. 38. 37. 39. 40. 41. 132. 43. 129.]	102
Payment	5	['Debit Card' 'UPI' 'Credit Card' 'Cash on Delivery' 'E wallet' nan]	109
Gender	2	['Female' 'Male' nan]	108
Service_Score	6	[3. 2. 1. nan 0. 4. 5.]	98
Account_user_count	6	[3. 4. nan 5. 2. 1. 6.]	444
account_segment	5	['Super' 'Regular Plus' 'Regular' 'HNI' nan 'Super Plus']	97
CC_Agent_Score	5	[2. 3. 5. 4. nan 1.]	116
Marital_Status	3	['Single' 'Divorced' 'Married' nan]	212
rev_per_month	58	[9. 7. 6. 8. 3. 2. 4. 10. 1. 5. nan 130. 19. 139. 102. 120. 138. 127. 123. 124. 116. 21. 126. 134. 113. 114. 108. 140. 133. 129. 107. 118. 11. 105. 20. 119. 121. 137. 110. 22. 101. 136. 125. 14. 13. 12. 115. 23. 122. 117. 131. 104. 15. 25. 135. 111. 109. 100. 103.]	791
Complain_ly	2	[1. 0. nan]	357
rev_growth_yoy	19	[11. 15. 14. 23. 22. 16. 12. 13. 17. 18. 24. 19. 20. 21. 25. 26. nan 4. 27. 28.]	3
coupon_used_for_payment	17	[1. 0. 4. 2. 9. 6. 11. 7. 12. 10. 5. 3. 13. 15. 8. nan 14. 16.]	3
Day_Since_CC_connect	23	[5. 0. 3. 7. 2. 1. 8. 6. 4. 15. nan 11. 10. 9. 13. 12. 17. 16. 14. 30. 46. 18. 31. 47.]	358
cashback	5692	[159.93 120.9 nan ... 227.36 226.91 191.42]	473
Login_device	2	['Mobile' 'Computer' nan]	760

Table 3. Dataset with unique values & missing value counts

Now all the anomalies are treated, only missing values are present in the entire dataset.

Missing Value treatment

Below the list are the missing values present in the variables. It is listed in the descending order.

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

Table 4.Missing Value counts in the Dataset

The columns 'rev_growth_yoy' and 'coupon_used_for_payment' are just with 3 missing values. It can be dropped from the analysis.

Imputing Missing Categorical Columns using Mode

The columns such as 'Payment','Gender','account_segment','Marital_Status' and 'Login_device' are categorical columns. Addition to this, the column 'City_Tier' and 'Complain_ly' can be categorized as object due to the nature of the data. So, these variables can be imputed using the ***most frequent values*** present in the respective columns.

Below table shows the frequency distribution of data before and after imputing the missing values using the mode.

<table>
<tr><th>Column Name</th><th>Distribution before imputing</th><th>Distribution after imputing</th></tr>
<tr><td>Payment</td><td>
<table>
<tr><th></th><th>Payment</th><th>Count</th><th>Percent</th></tr>
<tr><td>0</td><td>Debit Card</td><td>4585</td><td>0.41</td></tr>
<tr><td>1</td><td>Credit Card</td><td>3508</td><td>0.31</td></tr>
<tr><td>2</td><td>E wallet</td><td>1216</td><td>0.11</td></tr>
<tr><td>3</td><td>Cash on Delivery</td><td>1014</td><td>0.09</td></tr>
<tr><td>4</td><td>UPI</td><td>822</td><td>0.07</td></tr>
</table>
</td><td>
<table>
<tr><th></th><th>Payment</th><th>Count</th><th>Percent</th></tr>
<tr><td>0</td><td>Debit Card</td><td>4694</td><td>0.42</td></tr>
<tr><td>1</td><td>Credit Card</td><td>3508</td><td>0.31</td></tr>
<tr><td>2</td><td>E wallet</td><td>1216</td><td>0.11</td></tr>
<tr><td>3</td><td>Cash on Delivery</td><td>1014</td><td>0.09</td></tr>
<tr><td>4</td><td>UPI</td><td>822</td><td>0.07</td></tr>
</table>
</td></tr>
<tr><td>Gender</td><td>
<table>
<tr><th></th><th>Gender</th><th>Count</th><th>Percent</th></tr>
<tr><td>0</td><td>Male</td><td>6699</td><td>0.6</td></tr>
<tr><td>1</td><td>Female</td><td>4447</td><td>0.4</td></tr>
</table>
</td><td>
<table>
<tr><th></th><th>Gender</th><th>Count</th><th>Percent</th></tr>
<tr><td>0</td><td>Male</td><td>6807</td><td>0.6</td></tr>
<tr><td>1</td><td>Female</td><td>4447</td><td>0.4</td></tr>
</table>
</td></tr>
<tr><td>Login Device</td><td>
<table>
<tr><th></th><th>Login_device</th><th>Count</th><th>Percent</th></tr>
<tr><td>0</td><td>Mobile</td><td>7479</td><td>0.71</td></tr>
<tr><td>1</td><td>Computer</td><td>3015</td><td>0.29</td></tr>
</table>
</td><td>
<table>
<tr><th></th><th>Login_device</th><th>Count</th><th>Percent</th></tr>
<tr><td>0</td><td>Mobile</td><td>8239</td><td>0.73</td></tr>
<tr><td>1</td><td>Computer</td><td>3015</td><td>0.27</td></tr>
</table>
</td></tr>
</table>

Account Segment

	account_segment	Count	Percent
0	Regular Plus	4122	0.37
1	Super	4059	0.36
2	HNI	1639	0.15
3	Super Plus	817	0.07
4	Regular	520	0.05

	account_segment	Count	Percent
0	Regular Plus	4219	0.37
1	Super	4059	0.36
2	HNI	1639	0.15
3	Super Plus	817	0.07
4	Regular	520	0.05

Marital Status

	Marital_Status	Count	Percent
0	Married	5859	0.53
1	Single	3515	0.32
2	Divorced	1668	0.15

	Marital_Status	Count	Percent
0	Married	6071	0.54
1	Single	3515	0.31
2	Divorced	1668	0.15

City Tier

	City_Tier	Count	Percent
0	1.0	7259	0.65
1	3.0	3403	0.31
2	2.0	480	0.04

	City_Tier	Count	Percent
0	1.0	7371	0.65
1	3.0	3403	0.30
2	2.0	480	0.04

Complain_ly

	Complain_ly	Count	Percent
0	0.0	7786	0.71
1	1.0	3111	0.29

	Complain_ly	Count	Percent
0	0.0	8143	0.72
1	1.0	3111	0.28

Table 5. Frequency distribution of data before and after imputing the missing values

There must not be much change in the distribution because of the imputation. If there is a significant change, then probably the imputation logic is not correct. Except for the Login device variable, all the other variables identically hold similar frequency distribution percentage. As there is not much difference in frequency distribution before and after imputation, we may assume **the imputation has happened correctly**.

At this point, let's encode the categorical columns based on their ordinality, value counts and nature of the data.

Categorical Variable	Categorical Values	Encoding
account_segment	Regular, Regular Plus, Super, Super Plus, HNI	0,1,2,3,4
Marital_Status	Single, Married, Divorced	0,1,2
Payment	Debit Card, Credit Card, E wallet, Cash on Delivery, UPI	0,1,2,3,4
Gender	Female, Male	0,1
Login_device	Mobile, Computer	0,1

Table 6.Encoding Categorical Variables

greatlearning
Power Ahead

Imputing Missing Numerical Columns using KNN Imputer

We can use **'KNNImputer'** imported from sklearn library. Each sample's missing values are imputed using the mean value from (n_neighbors) nearest neighbors found in the dataset. Based on the data, assuming 5 nearest neighbouring values to impute the missing values.

Missing values of column counts are shown below.

```
rev_per_month           791
cashback                473
Account_user_count      444
Day_Since_CC_connect    358
Tenure                  218
CC_Agent_Score          116
CC_Contacted_LY         102
Service_Score            98
```

After imputing the missing values using the KNN Imputer method, there are no missing values present in the dataset.

Our original dataset has 11260 records with 19 columns whereas now we have 11254 records with 18 columns present in the dataset in our analysis. We just have dropped the 6 missing records of '*rev_growth_yoy*' and '*coupon_used_for_payment'* variables and '*AccountID*' column from our analysis.

Let us round up the imputed mean(*decimals*) values of discrete and continuous variables except for the *cashback* variable which is originally provided in the decimal values, we do this just to be consistent with the original data format even though it would not impact our machine learining(*supervised*) modelling process.

Variables	Unique Counts	Unique Values
Tenure	38	[4. 0. 2. 13. 11. 9. 3. 99. 19. 20. 14. 8. 26. 18. 5. 30. 7. 1. 23. 29. 6. 28. 24. 25. 16. 10. 15. 22. 17. 27. 12. 21. 50. 60. 31. 51. 61. 38.]
City_Tier	3	[3. 1. 2.]
CC_Contacted_LY	44	[6. 8. 30. 15. 12. 22. 11. 9. 31. 18. 13. 20. 29. 28. 26. 14. 10. 25. 27. 17. 23. 33. 19. 35. 24. 16. 32. 21. 34. 5. 4. 126. 7. 36. 127. 42. 38. 37. 39. 40. 41. 132. 43. 129.]
Payment	5	[0. 4. 1. 3. 2.]
Gender	2	[0. 1.]
Service_Score	6	[3. 2. 1. 0. 4. 5.]
Account_user_count	6	[3. 4. 5. 2. 1. 6.]
account_segment	5	[2. 1. 0. 4. 3.]
CC_Agent_Score	5	[2. 3. 5. 4. 1.]
Marital_Status	3	[0. 2. 1.]

rev_per_month	66	[9. 7. 6. 8. 3. 2. 4. 10. 1. 5. 130. 19. 139. 102. 120. 138. 127. 123. 124. 116. 21. 31. 126. 134. 113. 32. 114. 108. 140. 29. 133. 129. 107. 118. 11. 105. 20. 119. 121. 137. 110. 27. 22. 101. 28. 136. 125. 14. 13. 12. 115. 23. 122. 117. 30. 131. 104. 15. 25. 135. 111. 109. 26. 24. 100. 103.]
Complain_ly	2	[1. 0.]
rev_growth_yoy	19	[11. 15. 14. 23. 22. 16. 12. 13. 17. 18. 24. 19. 20. 21. 25. 26. 4. 27. 28.]
coupon_used_for_payment	17	[1. 0. 4. 2. 9. 6. 11. 7. 12. 10. 5. 3. 13. 15. 8. 14. 16.]
Day_Since_CC_connect	23	[5. 0. 3. 7. 2. 1. 8. 6. 4. 15. 11. 10. 9. 13. 12. 17. 16. 14. 30. 46. 18. 31. 47.]
cashback	6129	[159.93 120.9 141.708 ... 227.36 226.91 91.42]
Login_device	2	[0. 1.]

Table 7.Dataset with unique values & unique value counts

Now, the data looks clean and ready for model building.

Outlier treatment using IQR method

Outliers are nothing but data points that differ significantly from other observations. They are the points that lie outside the overall distribution of the dataset. Let us plot the box plots to understand the outliers.

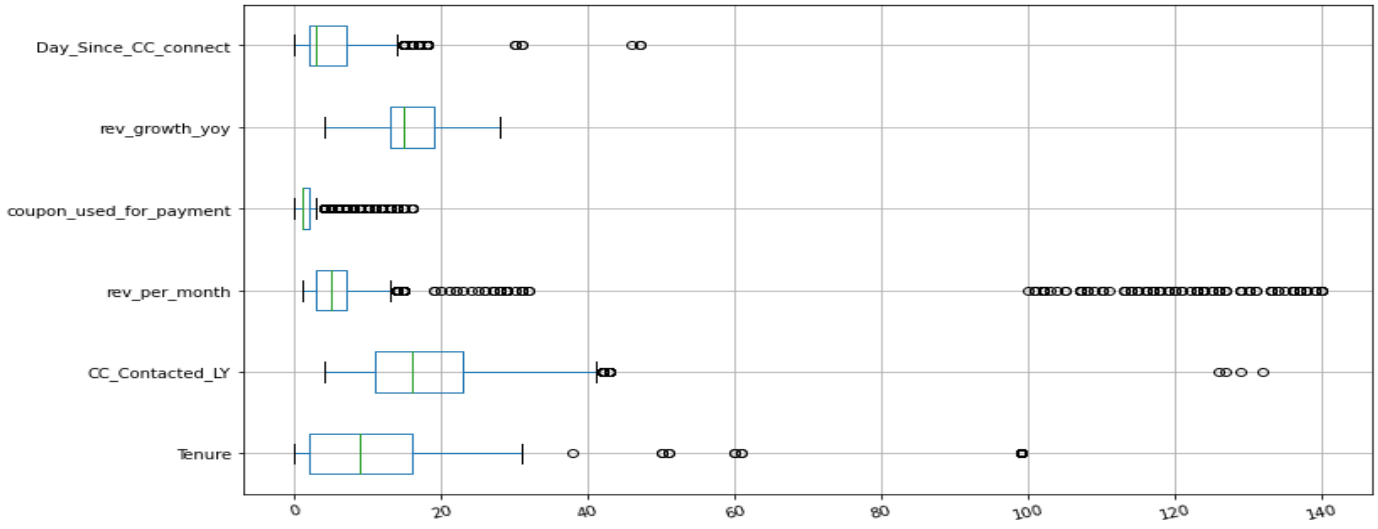

Figure 16.Boxplot to verify Outliers

- The variables Day_Since_CC_connect, CC_Contacted_LY, and Tenure have very minimal outliers.
- *Revenue growth percentage* variable has no outliers present. Additionally, by looking at the whiskers, seem to have a data is normally distributed.
- The variable coupon_used_for_payment contains lot of outliers.

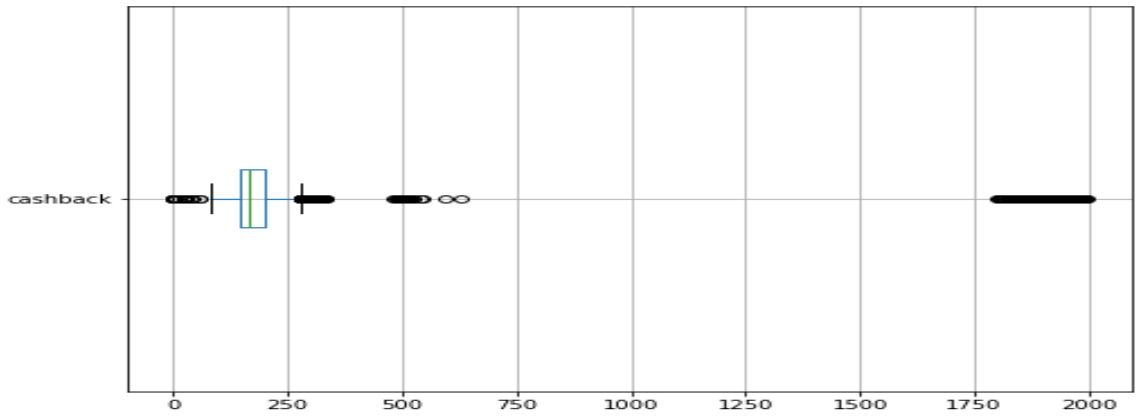

Figure 17.Boxplot for Cashback variable

- The variable rev_per_month and cashback has extreme outliers present.

Why do we need to treat Outliers?

The machine learning models such as Logistic Regression, KNN and Neural Networks have issues with outliers. Outliers, if not treated, can cause serious problems in statistical analyses. Let us plot the box plots to detect the outliers.

Assuming that treating the outliers using ***Inter Quartile Range method*** may improve the model performance. However, we will cross-validate the results before and after treating the outliers to see the percentage of accuracy increase in the model. Otherwise, we would keep the orignial records without removing the outliers.

Below table contains the list of outlier counts and its proportions before treating the outliers.

Variables	Outlier Counts	Outliers Percentage
Tenure	140	1.24%
CC_Contacted_LY	42	0.37%
rev_per_month	207	1.89%
coupon_used_for_payment	1380	12.26%
rev_growth_yoy	0	0%
Day_Since_CC_connect	131	1.16%
cashback	917	8.14%

Table 7.Outlier proportions before treatment

Q1,Q3=np.percentile(col,[25,75])
IQR=Q3-Q1
*lower_range= Q1-(1.5 * IQR)*
*upper_range= Q3+(1.5 * IQR)*

IQR Formula

By looking at the whiskers of all variables except **rev_growth_yoy**, we can say that the datas are not normally distributed. Additionally, using the **Shapiro-Wilk's Test, the** p-values of all the variables 0.0 which is less than α. (where α = 0.05). We can conclude that all the variables are NOT coming from the normal distribution.

To remove the outliers, let us calculate the Inter Quartile range between 0.25 and 0.75 and treat the lower and upper range values. Refer IQR formula above.

After treating the outliers, box plots of the numerical variables look neat.

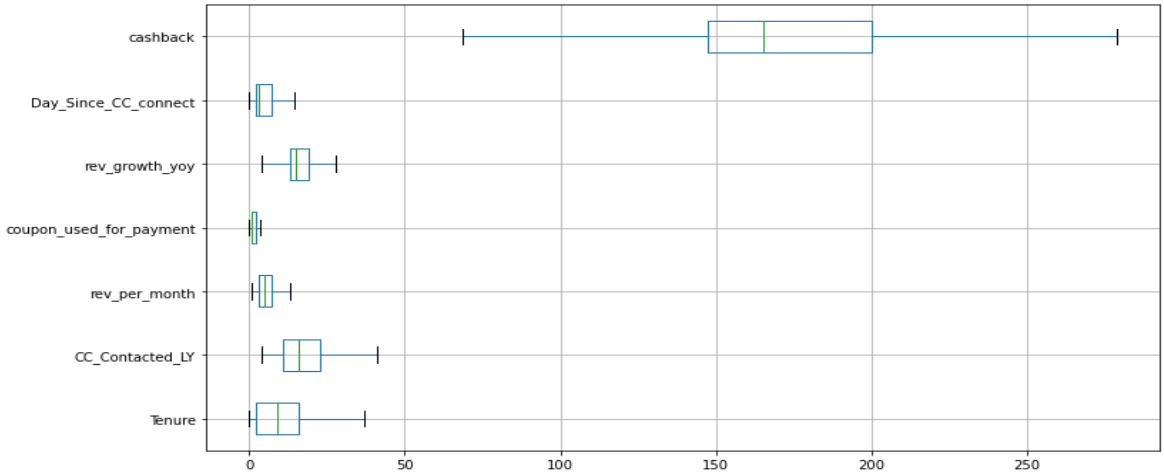

Figure 18.Boxplot after treating the outliers

Variable transformation

The variable **rev_per_month** and **cashback** has extreme outliers present. We will normalize the variables through log transformation method. For the cashback variable, there are 0 values present, so normal log transformation would result in infinite errors. So, perform log(x+1) transformation (*in python we use np.log1p method)*. This is the best way to avoid errors created by log transformation.

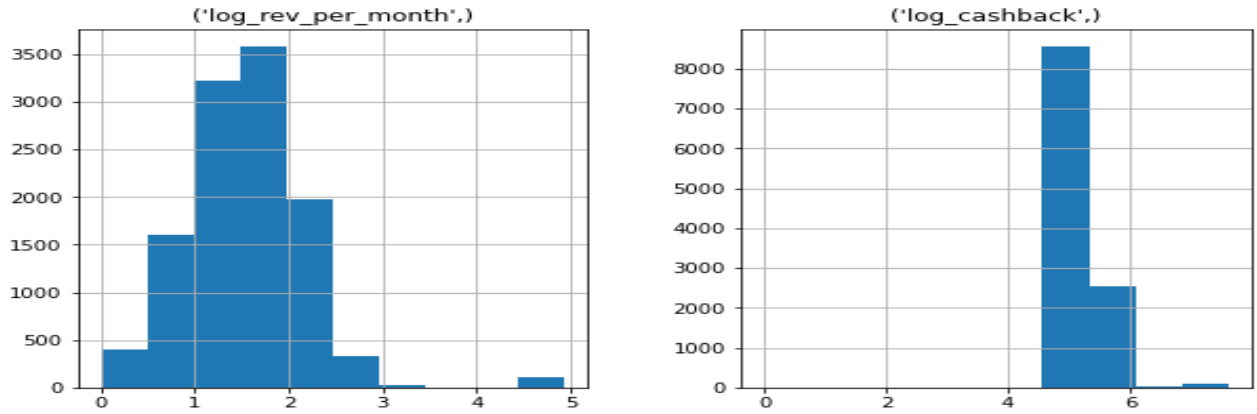

Figure 19.Log transformation of rev_per_month and cashback variables

Even after the normalization of variables, we could still observe the skewness. Based on the histograms, the models' accuracy would not get improve if we build the models with normalization variables. So, let us conclude that we will build the model without transformation of data.

Variables removed or added

All the provided variables seem to contribute some value to the Machine learning model. *AccountID* is the unique identifier field which can be excluded from our analysis. We do not see a need to remove any other variables at this point. However, using the ***Variation Inflation Factor Technique*** to treat the multi-collinearity, we can remove the features having the VIF values greater than 5 one after the other. Please see VIF section for detail explanation.

Let us group the records using the binning technique; it will help us make the dataset more manageable. We will run the dataset (*contains binning of few variables*) with the base models and analyze the outcome. We have grouped the variables of service/cc agent score, account type and cash back as shown below.

Variables	Values	Binned values
Service_Score	0,1	Poor
	2,3	Fair
	4,5	Good
account_segment	Regular, Regular Plus	Regular
	Super	Super
	Super Plus, HNI	HNI
CC_Agent_Score	1,2	Poor
	3,4	Satisfactory
	5	Excellent
cashback	0-100	Low-range cash back account
	100-200	Mid-Range cash back account
	>200	Highly generated cash back account

Table 8.Outlier proportions before treatment

With the binned data, below are the observations for all the models.

	Logistic Regression	LDA	Naive Bayes	CART	Random Forest	Gradient Boost	KNN	ANN
Base Model Recall	45.69	41.12	56.24	84.18	84.89	59.75	89.63	84.01
Base Model AUC	84	83.5	80.11	90.44	99.35	93.59	96.36	98.13
Recall after Binning	46.22	43.41	55.01	82.07	86.47	59.4	91.04	79.09
AUC after Binning	84.96	83.94	80.4	89.27	99.36	92.9	96.5	98.19

Table 9.Base Model comparison with binning data

- There is a slight improvement in the Logistic Regression, LDA and Naïve Bayes model recall score.
- The scores are slightly declined for Naïve Bayes, CART, Gradient Boosting and ANN model.
- There is a 2% increase in 'recall' score for KNN and Random Forest model.

Overall, binning does not make much improvement in the model results score. So, let's run the models without binning technique.

4. Model building

Train and Test Split

We build and run the models with the training data ratio of 70% and test set ratio of 30% and random state=1 to be consistent across the results. Using the 'stratify' option; will split equally among the train and test records according to the target classification variable (Churn).

Proportion of Target variable Churn

	Original Dataset
0	0.831527
1	0.168473

	Training Dataset
0	0.831535
1	0.168465

	Test Dataset
0	0.831507
1	0.168493

Ratio of split between original dataset, train dataset and test dataset is nominal.

Base Models

For our Chrun prediction business case problem let us consider the few Machine Learning classification algorithm from linear, non-linear and ensemble model techniques. For our DTH account churn prediction case, we are less careful about the '*account not churned*' and more care about the '*account churned*'. How many of the actual accounts are churning that we were able to predict correctly with our model. So, we aim for high '**recall'** in this case.

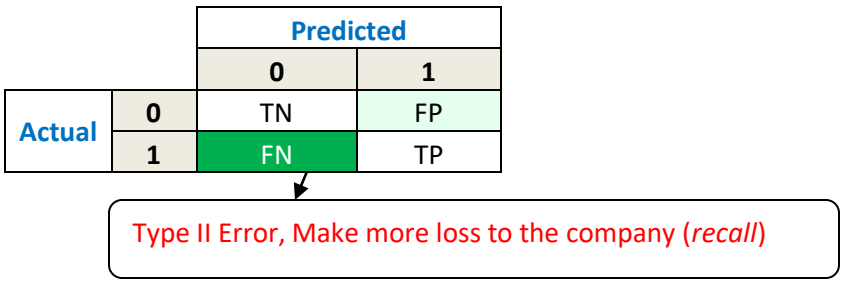

Figure 20.Log transformation of rev_per_month and cashback variables

So, the performance metrics mentioned below for all the models are only for 'account churned' (class 1). Identifying churners from non-churners is more valuable (or) more important than classifying non-churners to churners.

Multiple models within each type are built and optimal model is selected for comparison

- **Logistic Regression/LDA:** Very Poor metrics scores.
- **KNN:** 7 variables are selected as significant variables with the optimum K value as 2.
- **ANN:** Only Tenure has been selected as the significant variable.
- **CART:** 11 variables are selected as significant variables.
- **Random Forest:** 12 variables are selected as significant variables.
- **Gradient Boosting:** 7 variables are selected as significant variables.

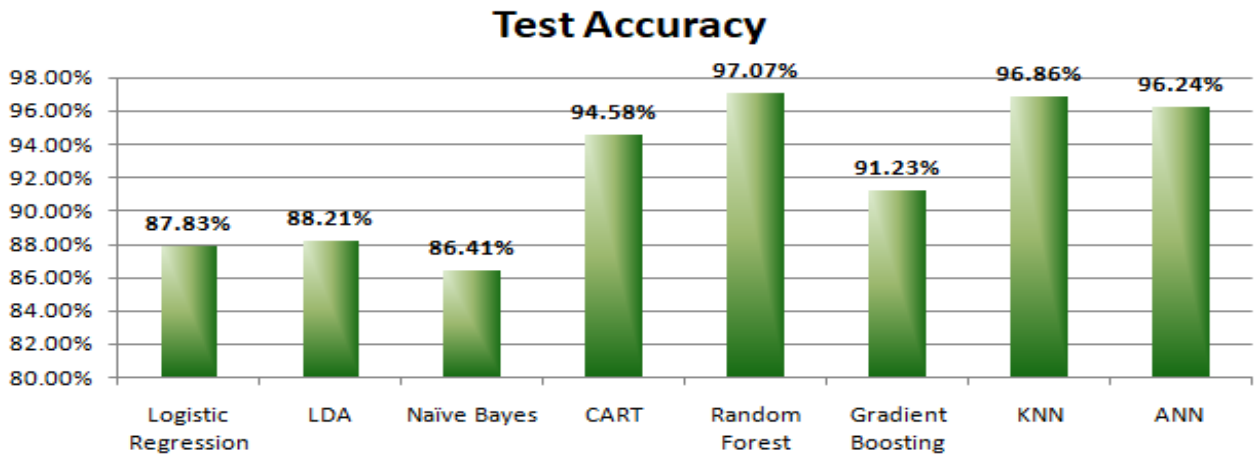

Figure 21.Accuracy score of all models

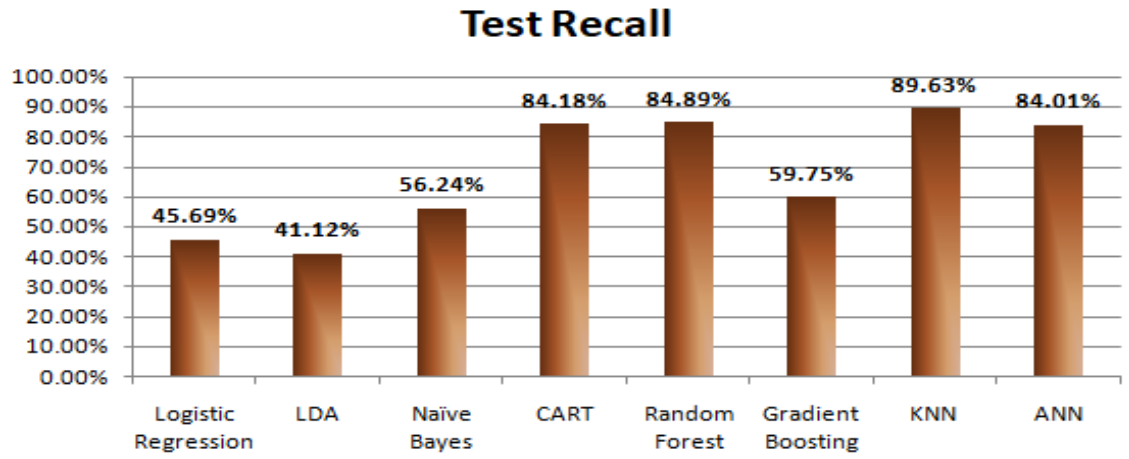

Figure 22.Recall Score of all models

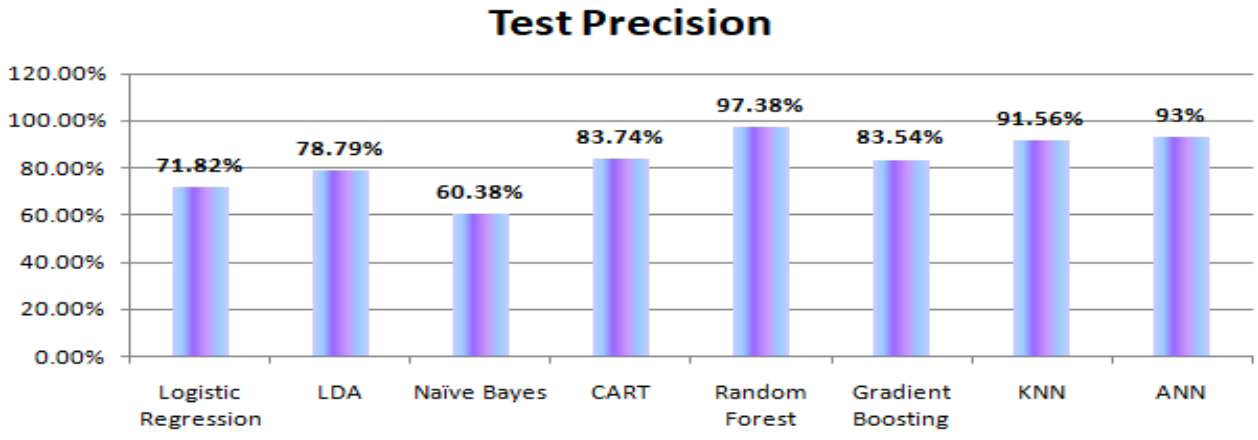

Figure 23.Precision score of all models

By looking at the charts, Model's accuracy, recall and precision scores are good for KNN, ANN, Random Forest, Gradient Boosting and CART Models.

- In general, tree based models works well with non-linear dataset.
- Random Forest can automatically balance data sets when a class is more infrequent than other classes in the dataset.
- Random Forest is easier to train & test, whereas ANN & GB models are computationally expensive.
- Random Forest does not require scaling, whereas KNN & ANN require the records in common scale & prone to outliers.
- Random Forest explains which features are more important.
- Random Forest model utilized most of the variables to make an account churn prediction which is lagging in KNN & ANN models.

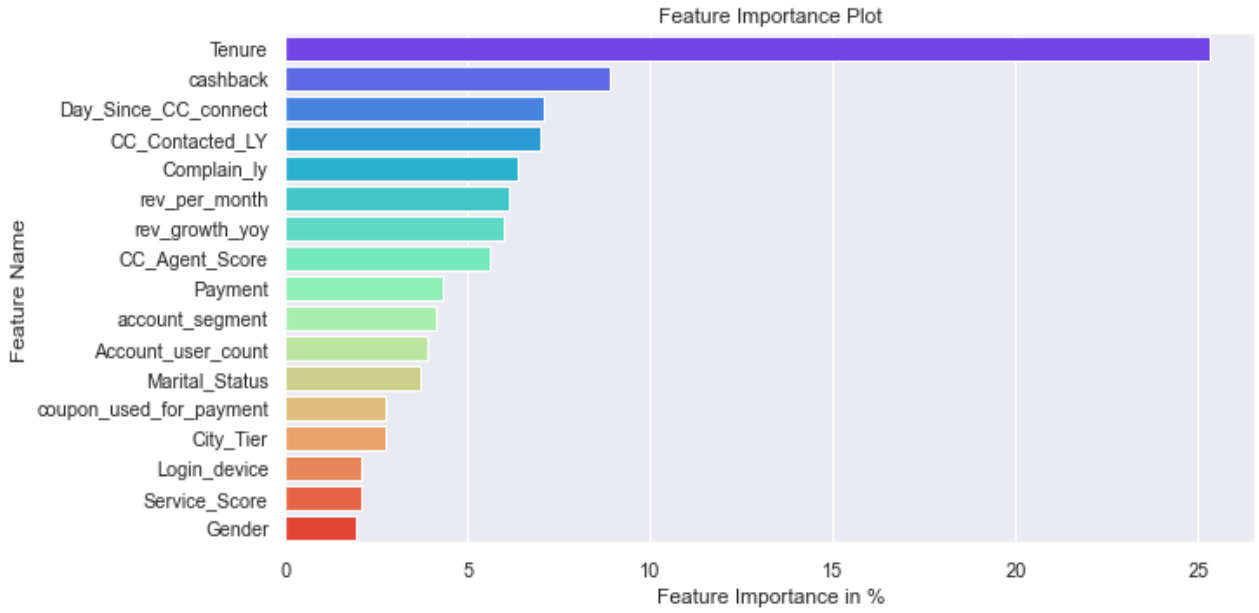

Figure 24.Random Forest Model Feature Importances

- RF makes the prediction utilizing 'Tenure' variable with ~25% & cumulative considerations of all the other continuous and categorical variables, which indicate that it, work well in the production data.
- The variables such as *Tenure, cashback, Day_Since_CC_connect, CC_Contacted_LY, Complain_ly, revenue, and CC_Agent_score* are the most important features to predict the churn.

Considering the above points, we can conclude that "**Random Forest is the efficient model for our DTH churn prediction classification problem**".

Effort to improve model performance

GridSearchCV Hyper Tuning Technique on Random Forest Model

Using the GridSearchCV function, we train the model with the below parameter combinations

```
param_grid = {'n_estimators':[250,300,350],'max_features': [12,14,16]}
```

The Random Forest classification model estimated the best parameters as below with the 'recall' as the scoring parameter with the cross-validation on the full data as 3. Cutting down the trees with *minimum split, maximum depth* parameters brought down the scores, let it grow as much as it can to classify the '*account churn or not*.'

After fine tuning further combinations, we attain the maximum recall/precision with the below hyper parameters

```
{max_features=14, n_estimators=300}
```

max_depth - Decision nodes ends maximum at 15, 16 or 20 in CART, let's choose around the same. Even in the Random Forest classifier model, the process will be building the multiple DTs and considering the maximum outcomes from the multiple DTs.

n_estimators - the number of trees you want to build before taking the maximum voting or averages of predictions.

max_features - These are the maximum number of features Random Forest is allowed to try in individual tree.

	Accuracy	Precision	Recall	F1 Score
Base Model	0.97	0.97	0.84	0.91
Hyper Tuning using GridSearchCV	0.97	0.95	**0.87**	0.9

Table 10.Base Model Vs Hyper Tuning Technique on Random Forest Model

After tuning the model, 'recall' score has improved from 84% to 87%.

Threshold Cutoff Technique on Random Forest Model

Model Names	Test Recall	Test Threshold Recall	Test F1 Score	Test F1 Threshold	Test AUC Score	Test AUC Threshold
KNN	0.89	-	0.9	-	0.96	-
ANN	0.91	-	0.92	-	0.99	-
Random Forest	0.87	**0.94**	0.9	**0.89**	0.99	**0.95**
Gradient Boost	0.85	0.92	0.88	0.87	0.99	0.94

Table 11.Threshold cutoff tuning on Random Forest Model

- When we run the Random Forest model with the cutoff value of 0.3 and a slight compromise on the F1 score of 4%, have attained an increase in the 'recall' score of about 7% (87% to 94%). This is the best 'recall' score compared to any other models.
- Additionally, the model performance results out with 95%, which is good AUC score.
- The cost implications are always there when we have more churners.

SMOTE Techinque on Random Forest Model

The Churn target data seems to be highly imbalanced with only 17% of 1's and 83% of 0's in the target variable.

One approach to addressing imbalanced datasets is to oversample the minority class. The simplest approach involves duplicating examples in the minority class, although these examples don't add any new information to the model. Instead, new examples can be synthesized from the existing examples. This is a type of data augmentation for the minority class and is referred to as the **Synthetic Minority Oversampling Technique**, or **SMOTE** for short.

We build and run the models with the SMOTE training data ratio of 70% and SMOTE test set ratio of 30% and random state=1 to be consistent across the results.

- Number of train records after SMOTE: 13101
- Number of test records after SMOTE: 5615

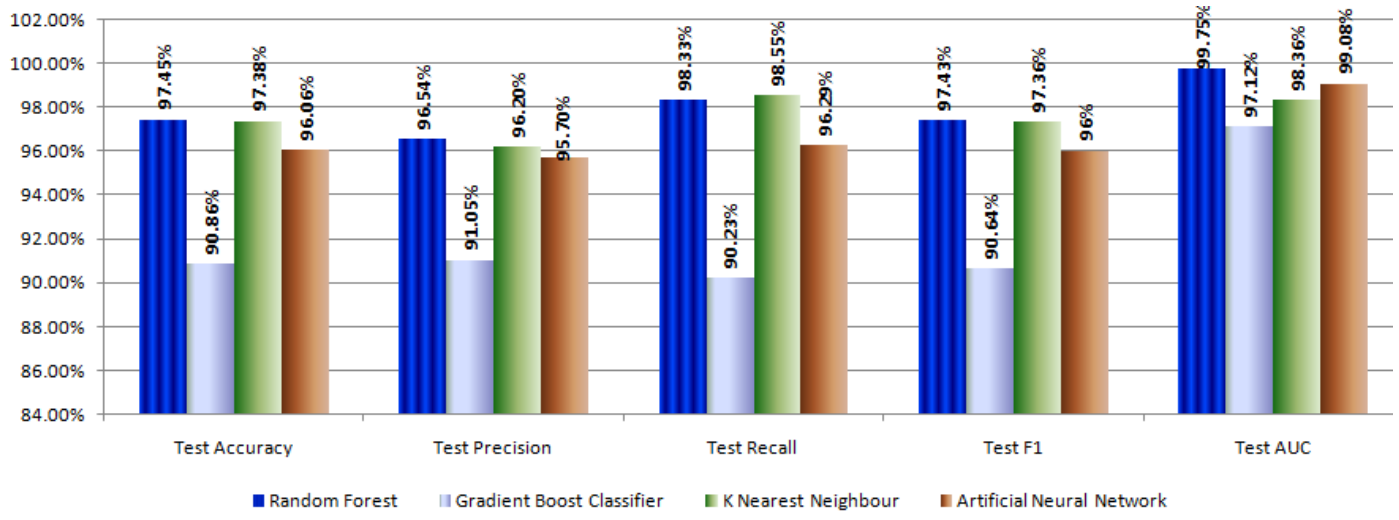

Figure 25.SMOTE tuning technique on efficient models

When we have balanced set of classes, Random Forest (*in dark blue bar*) outperforms all the other models in all aspects. Even, the model performance is close to 100%.

Variance Inflation Factor Technique on Random Forest Model

We will use VIF, to check if there is multicollinearity in the data.

Features having a VIF score >5 will be dropped/treated one after the other till all the features have a VIF score <5. The variables to be dropped are *cashback, Service_Score, rev_growth_yoy* and *Account_user_count*.

	Accuracy	Precision	Recall	F1 Score
Base Model	0.97	0.97	**0.84**	0.91
VIF Technique	0.97	0.96	**0.84**	0.9

Table 12.VIF technique on Random Forest Model

Even after dropping the above variables, the metric scores and model performance remains almost same. So, these columns might not be useful in predicting the DTH account churn.

Considering the results output using the tuning mechanisms, we can conclude that Random Forest is the optimum model for our DTH churn prediction problem.

5. Model validation

Let us look at the performance metrics of the Machine Learning models.

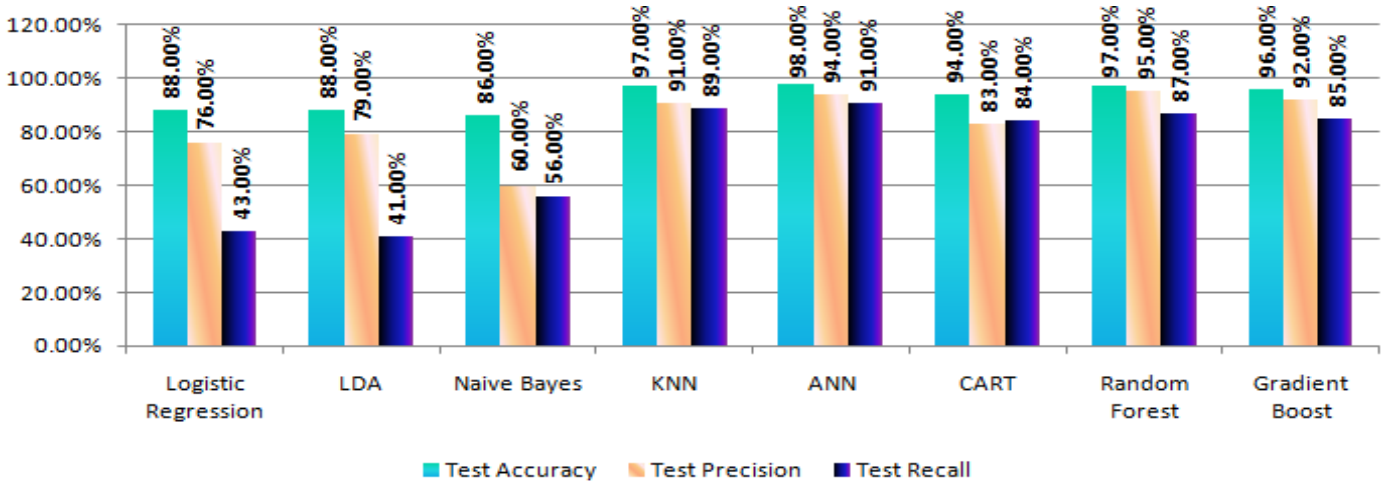

Figure 26.Performance metrics of all the models

By comparing the Accuracy, Precision and Recall metrics of different models, ANN, KNN, Random Forest and Gradient Boosting results out with good metrics scores.

Machine Learning models can be best validated using AUC score and ROC Curve. ROC Curve is the plot between False Positive Rate (X-axis) and True Positive Rate (Y-axis).

False Positive Rate: The Ratio of False Positives to the actual number of negatives. In the context of our model, it is a measure for how many cases did the model predicts that the account has been churned from all the accounts who actually did not churn.

True Positive Rate: The True positive rate (TPR) gives the proportion of correct predictions in predictions of positive class. In this context, it is a measure for how many cases did the model predicts that the account has been churned from all the accounts who actually churned.

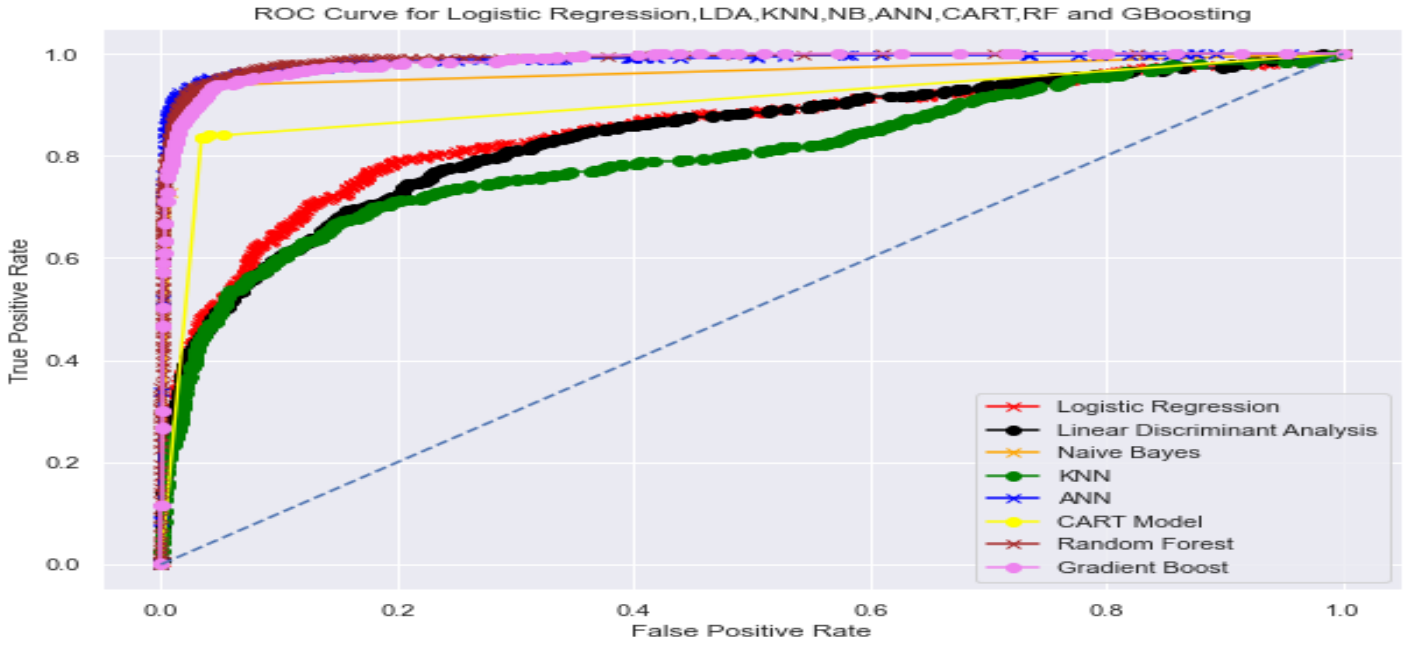

Figure 27.ROC Curve comparisons for all the models

- AUC Score of KNN is 96.36%
- AUC Score of Neural Networks is 98.49%
- AUC Score of Random Forest Model is 98.97%
- AUC Score of Gradient Boosting Model is 98.51%

Logistic Regression, LDA, Naïve Bayes, KNN have the poor ROC curve. CART is not better than ANN & ensemble models.

Although, ANN, Gradient Boosting and Random Forest models ROC curve is superimpose over one another, considering the performance metrics score, model evaluation and all other observations with the other models in the previous inferences section, the way the tree based models works for non-linear dataset, we can conclude that '**Random Forest**' is the best optimum model and will work well in the production environment for our DTH account churn prediction problem.

6. Final interpretation / recommendation

Interpretations

- Churn is observed for the accounts contacted the customer care frequently during the 1st 2 years in the Tenure.
- High Churn is observed for the account types Regular and HNI even the account stays in the tenure more than 5 years.
- 32% of the churn of accounts occurred when the complaint has been raised.
- 25% of the churn of accounts occurred when the payment is COD, so implicitly indicate that customers are not treated well when step into the store.
- Even the account counts of Regular Plus and Super is almost similar, we observed that 27% of the churn occurred with the account type as Regular Plus but when they switch onto the next level (i.e., Super), the churn rate is reduced to 10%.
- Offers and plans to be revisited between these two account types.

Recommendations

- Improve the services in the store when customers approach to pay (COD). Set up a token-based system to pay their monthly charges. Provide them a feedback form to get the ratings, & take appropriate actions.
- Keep your new customers happy with regular follow-ups (Quarterly once at least for 2 years) to get their feedback about the services, and whether the chosen plan meets their expectations or not; because, the more churn occurred during the first 2 years in the contract.
- Since the Super account type is just the next level to Regular Plus account type, the churn rate has come down to 10% from 27%. Offers and plans need to be revisited between these two account types.
- Don't provide more offers to the Regular & HNI account types; because they are ready to cancel the service regardless of the tenure.
- The churn rate for the accounts that raised the complaint is 32%, whereas when they have not risen it is just 11%. So, inform the customer service to give more importance to the cases that have raised the complaint, keeping your customer a service issue to a minimum is the key.
- The times the customers of the account contacted customer care are high then the churn rate is also increased. Customer service has always affected the customer satisfaction. From issues reaching the call centers to the actual resolution of the customer's problem. Take necessary action to improve the customer service experience.

THE END!