

Capstone Project on Customer Churn Final Report

Nandha Keshore Utti

PG-DSBA Online

March' 22

Date: 19/03/2023

Contents

1. Introduction	6
- Brief introduction about the problem statement and the need of solving it.	6
2. EDA - Data Cleaning and Pre-processing	7
- Approach used for identifying and treating missing values and outlier treatment (and why)	8
- Need for variable transformation (if any)	10
- Variables removed or added and why (if any)	8
3. EDA - Data visualization and Business implications	11
- Uni-variate / Bi-variate / multi-variate analysis to understand relationship b/w variables. How your analysis is impacting the business?	11, 17, 19
- Both visual and non-visual understanding of the data.	11, 17, 19
4. Model building	25
- Clear on why was a particular model(s) chosen.	25
- Effort to improve model performance.	41
5. Model validation	54
- How was the model validated? Just accuracy, or anything else too?	54
6. Final interpretation / recommendation	57
- Detailed recommendations for the management/client based on the analysis done.	57

List of figures

Fig.1 – Dataset columns’ list	8
Fig.2 – ‘Account_segment’ variable categories list before anomaly treatment	09
Fig.3 – ‘Account_segment’ variable categories list after anomaly treatment	09
Fig.4 – ‘Gender’ variable categories list before anomaly treatment	10
Fig.5 – ‘Gender’ variable categories list after anomaly treatment	10
Fig.6 – Hist plots of all numeric variables	11
Fig.7 – Count plot of all categorical variables	13
Fig.8 – Count plots of all discrete variables	14
Fig.9 – Box plots before outlier treatment	15
Fig.10 – Box plots after outlier treatment	16
Fig.11 – Pair plot	17
Fig.12 – Heat map	18
Fig.13 – Churn vs Account_segment plot	19
Fig.14 – Churn vs CC_Agent_score	19
Fig.15 – Churn vs City_Tier	20
Fig.16 – Churn vs Gender & Marital status	20
Fig.17 – Churn vs Account_segment & CC_Contacted_LY	21
Fig.18 – Revenue vs Account_user_count	21
Fig.19 – Account_segment vs Tenure	22
Fig.20 – Account_segment vs Complaint_ly	22
Fig.21 – Churn vs Account_segment & Coupon_used_for_payment	23
Fig.22 – Churn variable ratios	23
Fig.23 – Train and Test data set shapes before SMOTE	25
Fig.24 – Train and Test data set split percentages after splitting	26
Fig.25 – Train and Test data set shapes after SMOTE	26
Fig.26 – Target variable ‘Churn’ split ratio after SMOTE	26
Fig.27 – CART model after fitting on the train data	27
Fig.28 – Confusion matrix of train data set from CART model	27
Fig.29 – ROC curve of train data set from CART model	28
Fig.30 – Confusion matrix of test data set from CART model	28
Fig.31 – ROC curve of test data set from CART model	29
Fig.32 – Random-Forest model after fitting on the train data	30
Fig.33 – Confusion matrix of train data set from RF model	30
Fig.34 – ROC curve of train data set from RF model	31
Fig.35 – Confusion matrix of test data set from RF model	31
Fig.36 – ROC curve of test data set from RF model	32
Fig.37 – Logistic Regression model after fitting on the train data	33
Fig.38 – Confusion matrix of train data set from Logistic Regression model	33
Fig.39 – ROC curve of train data set from Logistic Regression model	34
Fig.40 – Confusion matrix of test data set from Logistic Regression model	34
Fig.41 – ROC curve of test data set from Logistic Regression model	35
Fig.42 – Statsmodel-Logistic Regression model parameters before removing insignificant variables	36
Fig. 43 – Statsmodel-Logistic Regression model parameters after removing insignificant variables	37

Fig.44 – LDA model after fitting on the train data	37
Fig.45 – Confusion matrix of train data set from LDA model	38
Fig.46 – ROC curve of train data set from LDA model	38
Fig.47 – Confusion matrix of test data set from LDA model	38
Fig.48 – ROC curve of test data set from LDA model	39
Fig.49 – KNN model after fitting on the train data	39
Fig.50 – Confusion matrix of train data set from KNN model	40
Fig.51 – ROC curve of train data set from KNN model	40
Fig.52 – Confusion matrix of test data set from KNN model	41
Fig.53 – ROC curve of test data set from KNN model	41
Fig.54 – GridSearchCV CART model after fitting on the train data	41
Fig.55 – Best parameters of GridSearchCV CART model	42
Fig.56 – Confusion matrix of train data set from GridSearchCV CART model	42
Fig.57 – ROC curve of train data set from GridSearchCV CART model	43
Fig.58 – Confusion matrix of test data set from GridSearchCV CART model	43
Fig.59 – ROC curve of test data set from GridSearchCV CART model	43
Fig.60 – GridSearchCV RF model after fitting on the train data	44
Fig.61 – Best parameters of GridSearchCV RF model	44
Fig.62 – Confusion matrix of train data set from GridSearchCV RF model	44
Fig.63 – ROC curve of train data set from GridSearchCV RF model	45
Fig.64 – Confusion matrix of test data set from GridSearchCV RF model	45
Fig.65 – ROC curve of test data set from GridSearchCV RF model	45
Fig.66 – GridSearchCV KNN model after fitting on the train data	46
Fig.67 – Best parameters of GridSearchCV KNN model	46
Fig.68 – Confusion matrix of train data set from GridSearchCV KNN model	46
Fig.69 – ROC curve of train data set from GridSearchCV KNN model	47
Fig.70 – Confusion matrix of test data set from GridSearchCV KNN model	47
Fig.71 – ROC curve of test data set from GridSearchCV KNN model	48
Fig.72 – Bagging Classifier model after fitting on the train data	48
Fig.73 – Confusion matrix of train data set from Bagging Classifier model	49
Fig.74 – ROC curve of train data set from Bagging Classifier model	49
Fig.75 – Confusion matrix of test data set from Bagging Classifier model	49
Fig.76 – ROC curve of test data set from KNN model	50
Fig.77 – Adaboost Classifier model after fitting on the train data	50
Fig.78 – Confusion matrix of train data set from Adaboost Classifier model	51
Fig.79 – ROC curve of train data set from Adaboost Classifier model	51
Fig.80 – Confusion matrix of test data set from Adaboost Classifier model	52
Fig.81 – ROC curve of test data set from Adaboost model	52
Fig.82 – Gradient Boosting Classifier model after fitting on the train data	52
Fig.83 – Confusion matrix of train data set from Gradient Boosting Classifier model ...	53
Fig.84 – ROC curve of train data set from Gradient Boosting Classifier model	53
Fig.85 – Confusion matrix of test data set from Gradient Boosting Classifier model	54
Fig.86 – ROC curve of test data set from Gradient Boosting model	54

List of tables

Table.1 – Data frame with first sample of 5 rows	7
Table.2 – Dataset information	8

Table.3 – Dataset null information	8
Table.4 – Sample Data frame after dropping ‘AccountID’ variable	9
Table.5 – Dataset variables after treating null values	10
Table.6 – Data types after treating the anomalies	11
Table.7 – Skewness table	12
Table.8 – Sample Dataset description table	25
Table.9 – Sample Data frame after Data Encoding	26
Table.10 – Sample Train Data set after Data Encoding	27
Table.11 – Sample Test Data set after Data Encoding	28
Table.12 – Classification report of train dataset of CART model	29
Table.13 – Classification report of test dataset of CART model	30
Table.14 – Sample Feature importance table of CART model	30
Table.15 – Classification report of train dataset of RF model	32
Table.16 – Classification report of test dataset of RF model	33
Table.17 – Sample Feature importance table of RF model	33
Table.18 – Classification report of train dataset of Logistic Regression model	34
Table.19 – Classification report of test dataset of Logistic Regression model	35
Table.20 – Statsmodel-Logistic Regression model results before removing insignificant variables	36
Table.21 – Statsmodel-Logistic Regression model results after removing insignificant variables	37
Table.22 – Classification report of test dataset of LDA model	39
Table.23 – Classification report of test dataset of LDA model	40
Table.24 – Classification report of train dataset of KNN model	41
Table.25 – Classification report of test dataset of KNN model	42
Table.26 – Classification report of train dataset of GridSearchCV CART model	43
Table.27 – Classification report of test dataset of GridSearchCV CART model	44
Table.28 – Classification report of train dataset of GridSearchCV RF model	45
Table.29 – Classification report of test dataset of GridSearchCV RF model	46
Table.30 – Classification report of train dataset of GridSearchCV KNN model	47
Table.31 – Classification report of test dataset of GridSearchCV KNN model	48
Table.32 – Classification report of train dataset of Bagging Classifier model	50
Table.33 – Classification report of test dataset of Bagging Classifier model	51
Table.34 – Classification report of train dataset of Adaboost Classifier model	52
Table.35 – Classification report of test dataset of Adaboost Classifier model	53
Table.36 – Classification report of train dataset of Gradient Boosting model	54
Table.37 – Classification report of test dataset of Gradient Boosting model	55
Table.38 – Summary of performance metrics of all the classification models	56
Table.39 – Summary of performance metrics of the Ensemble models	56
Table.40 – Summary of performance metrics of the GridSearchCV tuned models in terms of misclassifications	57
Table.41 – Summary of performance metrics of the GridSearchCV tuned models	57

Customer Churn

1. Introduction

Defining problem statement:

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

You have been assigned to develop a churn prediction model for the company and provide business recommendations on the campaign.

Your campaign suggestion should be unique and be very clear on the campaign offer because your recommendation will go through the revenue assurance team. If they find that you are giving a lot of free (or subsidized) stuff thereby making a loss to the company; they are not going to approve your recommendation. Hence be very careful while providing campaign recommendation.

Need of the study/project:

- Churn analysis is the evaluation of a company's customer loss rate in order to reduce it and it can be minimized by assessing your product and how people react to it on various factors.
- It's a fact acquiring new customers is a costly affair but losing the existing customers will cost even more for the business or the organization.
- The competition in any market is on a rise and this encourages organizations to focus not only on new business but also on retaining existing customers.
- A customer's intention to stop using a particular product/service may always be a decision formed over time. There are various factors that lead to this decision and it's important for organizations to understand each and every factor so that customers can be convinced to stay and keep making purchases.
- So, there is a need to understand customers behavioural analysis by understanding various factors considering into account such as their buying pattern, revenue generated by the customers, tenure of the customer, activity of the customer towards the purchases etc.
- And this can be done by constantly conducting customer satisfaction surveys and analysing the received feedback.

Ref: <https://www.paddle.com/resources/customer-churn-analysis>

Understanding business/social opportunity:

- *Reducing Risk of the Business:* Analysis of customer churn prediction and retaining them is more important than acquiring a new customer in business perspective. Customer churn indicates a direct loss to the business. Selling a new product/service to an existing customer will be much easier than selling it to a new customer. Thus, customer churn can be harmful to the growth of the business.

- *Gain information for improvement:* Dissatisfied customers are a source of constructive feedback for an organization's betterment. An organization will gain information about aspects that need to be improved while implementing strategies to prevent customer churn.
- *Understand the target market:* Constantly working towards the reduction of customer churn will uncover layers of the market which were otherwise unknown. Survey focus groups and other such activities can be carried out to know the target market in a better manner and in turn reduce customer churn.
- *Build a competitive advantage in the market:* In a world where there is constant competition to attain new customers and retain existing ones, having an edge over the competition is important. In the process of reducing customer churn, not only do customers know unknown aspects of a business but also build a competitive advantage over the others in the market.

Ref: <https://www.questionpro.com/blog/customer-churn/>

2. EDA - Data Cleaning and Pre-processing

Understanding how data was collected in terms of time, frequency and methodology:

- If we look at the feature 'rev_growth_yoy', i.e., revenue growth percentage of the account (last 12 months vs last 24 to 13 month), we can say data collected is of ~ 2 years.
- Data collection is mainly focused on customers' personal data, revenue generated by the customers of the account, feedback from the customers.
- Customers' accounts also segmented, can be performed some cluster analysis if required based on the available data.

Visual inspection of data (rows, columns, descriptive details):

Data frame with sample of first 5 rows:

	AccountID	Churn	Tenure	City_Tier	CC_Contacted_LY	Payment	Gender	Service_Score	Account_user_count	account_segment	CC_Agent_Score
0	20000	1	4	3.0	6.0	Debit Card	Female	3.0	3	Super	2.0
1	20001	1	0	1.0	8.0	UPI	Male	3.0	4	Regular Plus	3.0
2	20002	1	0	1.0	30.0	Debit Card	Male	2.0	4	Regular Plus	3.0
3	20003	1	0	3.0	15.0	Debit Card	Male	2.0	4	Super	5.0
4	20004	1	0	1.0	12.0	Credit Card	Male	2.0	3	Regular Plus	5.0

Table. 01

- Data set contains 11260 records and 19 features.

Data set columns' list:

```
Index(['Churn', 'Tenure', 'City_Tier', 'CC_Contacted_LY', 'Payment', 'Gender',
      'Service_Score', 'Account_user_count', 'account_segment',
      'CC_Agent_Score', 'Marital_Status', 'rev_per_month', 'Complain_ly',
      'rev_growth_yoy', 'coupon_used_for_payment', 'Day_Since_CC_connect',
      'cashback', 'Login_device'],
      dtype='object')
```

Fig. 01

- No discrepancies found in columns' names, so renaming of the columns is not required.

Understanding of attributes (variable info):

Dataset information:

- Given data set contains 12 object, 7 numerical variables.
- Some of the numerical variables should be shown as numerical variable, but it is showing as object type.
e.g., 'rev_growth_yoy', 'cashback'

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11260 entries, 0 to 11259
Data columns (total 19 columns):
#   Column                Non-Null Count  Dtype
---  -
0   AccountID             11260 non-null  int64
1   Churn                 11260 non-null  int64
2   Tenure                11158 non-null  object
3   City_Tier             11148 non-null  float64
4   CC_Contacted_LY      11158 non-null  float64
5   Payment               11151 non-null  object
6   Gender                11152 non-null  object
7   Service_Score         11162 non-null  float64
8   Account_user_count    11148 non-null  object
9   account_segment       11163 non-null  object
10  CC_Agent_Score        11144 non-null  float64
11  Marital_Status        11048 non-null  object
12  rev_per_month         11158 non-null  object
13  Complain_ly           10903 non-null  float64
14  rev_growth_yoy        11260 non-null  object
15  coupon_used_for_payment 11260 non-null  object
16  Day_Since_CC_connect  10903 non-null  object
17  cashback              10789 non-null  object
18  Login_device          11039 non-null  object
dtypes: float64(5), int64(2), object(12)
memory usage: 1.6+ MB
```

Table. 02

Data set null information:

- Null values are present in all the variables except 'AccountID', 'Churn', 'rev_growth_yoy', 'coupon_used_for_payment'.
- There are no duplicated records in the data set.

```
AccountID           0
Churn               0
Tenure             102
City_Tier          112
CC_Contacted_LY    102
Payment            109
Gender             108
Service_Score      98
Account_user_count 112
account_segment     97
CC_Agent_Score     116
Marital_Status     212
rev_per_month      102
Complain_ly        357
rev_growth_yoy      0
coupon_used_for_payment 0
Day_Since_CC_connect 357
cashback           471
Login_device       221
dtype: int64
```

Table. 03

Removal of unwanted variables:

- 'AccountID' is not a significant variable for the model building.

Sample Data frame after dropping 'AccountID' variable:

	Churn	Tenure	City_Tier	CC_Contacted_LY	Payment	Gender	Service_Score	Account_user_count	account_segment	CC_Agent_Score	Marital_Status
0	1	4	3.0	6.0	Debit Card	Female	3.0	3	Super	2.0	Single
1	1	0	1.0	8.0	UPI	Male	3.0	4	Regular +	3.0	Single
2	1	0	1.0	30.0	Debit Card	Male	2.0	4	Regular +	3.0	Single
3	1	0	3.0	15.0	Debit Card	Male	2.0	4	Super	5.0	Single
4	1	0	1.0	12.0	Credit Card	Male	2.0	3	Regular +	5.0	Single

Table. 04

- Now, Data set is of 11260 records and 18 features.

Missing Value treatment:

- All the variables null values are treated by using mode method except for the variable 'cashback'.
- As 'cashback' variable is continuous in nature, its nulls are treated by mean method. Remaining all the variables have nulls are more of categorical in nature, so these are treated by mode method.

Data set variables after treating null values:

```
Churn          0
Tenure         0
City_Tier      0
CC_Contacted_LY  0
Payment        0
Gender         0
Service_Score  0
Account_user_count  0
account_segment  0
CC_Agent_Score  0
Marital_Status  0
rev_per_month  0
Complain_ly    0
rev_growth_yoy  0
coupon_used_for_payment  0
Day_Since_CC_connect  0
cashback       0
Login_device   0
dtype: int64
```

Table. 05

Anomalies treatment:

- Some variables have some special characters in their entries like shown below:

Tenure: #

Gender: F for Female and M for Male

Account_user_count: @
 Account_segment: 'Regular Plus & Regular +' ; 'Super Plus & Super +'
 rev_per_month: +
 rev_growth_yoy: \$
 coupon_used_for_payment: #, \$, *
 Days_Since_CC_connect: \$
 Cashback: \$
 Login_Device: &&&&

- 'Tenure', 'Account_user_count', 'rev_per_month', 'rev_growth_yoy', 'coupon_used_for_payment', 'Days_Since_CC_connect', 'cashback', 'Login_Device' variables have special characters. So, anomalies in these variables are treated by using mode method.
- 'Account_segment' has two different entries for the same kind of segment i.e., for 'Regular +' and 'Super +' like shown below:

```
array(['Super', 'Regular Plus', 'Regular', 'HNI', 'Regular +', nan,
      'Super Plus', 'Super +'], dtype=object)
```

Fig. 02

These anomalies are treated by replacing them with the one name like shown below:

```
array(['Super', 'Regular +', 'Regular', 'HNI', 'Super +'], dtype=object)
```

Fig. 03

- 'Gender' also has two different entries for the same kind of gender i.e., F for Female and M for Male.

```
array(['Female', 'Male', 'F', nan, 'M'], dtype=object)
```

Fig. 04

These anomalies are treated by replacing them with the one name like shown below:

```
array(['Female', 'Male'], dtype=object)
```

Fig. 05

Data types after treating the anomalies:

- It can be seen that data types showing correctly after treating the anomalies.

Churn	int64
Tenure	float64
City_Tier	float64
CC_Contacted_LY	float64
Payment	object
Gender	object
Service_Score	float64
Account_user_count	int64
account_segment	object
CC_Agent_Score	float64
Marital_Status	object
rev_per_month	float64
Complain_ly	float64
rev_growth_yoy	float64
coupon_used_for_payment	float64
Day_Since_CC_connect	float64
cashback	float64
Login_device	object
dtype:	object

Table. 06

3. EDA – Data Visualization and Business Implication

Univariate analysis:

Hist plots of all the numerical variables:

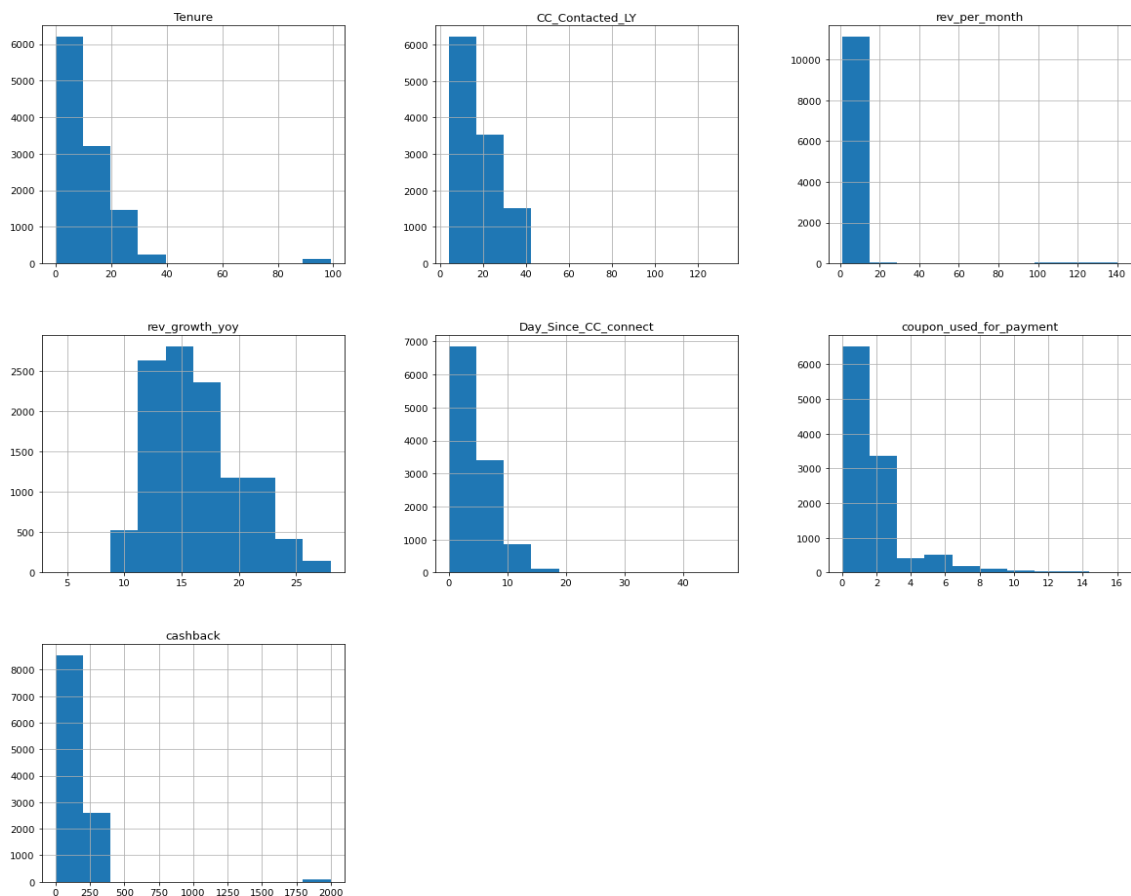


Fig. 06

- None of the variables are symmetrical in nature.
- Remaining all the variables have some skewness in their distributions.
- Let us check the distribution clearly by checking skewness values:

Tenure	3.901903
CC_Contacted_LY	1.436919
rev_per_month	9.412240
rev_growth_yoy	0.752886
Day_Since_CC_connect	1.293829
coupon_used_for_payment	2.575680
cashback	8.966070
dtype:	float64

Table. 07

Interpretations:

- Normally distributed variables: None
- Highly right skewed variables: 'Tenure', 'rev_per_month', 'cashback'
- Moderately right skewed variables: 'CC_Contacted_LY', 'rev_growth_yoy', 'Day_since_CC_connect', 'coupon_used_for_payment'

Count plots of all categorical variables:

- Customers are most preferred mode of payment is through debit and credit cards, least by COD.
- Majority of the accounts primely owned by males.
- 'Super' and 'Regular +' account segments customers are the most valuable.
- Majority of account owner's marital status is married.
- Customers are preferring mobile login over computer logins.

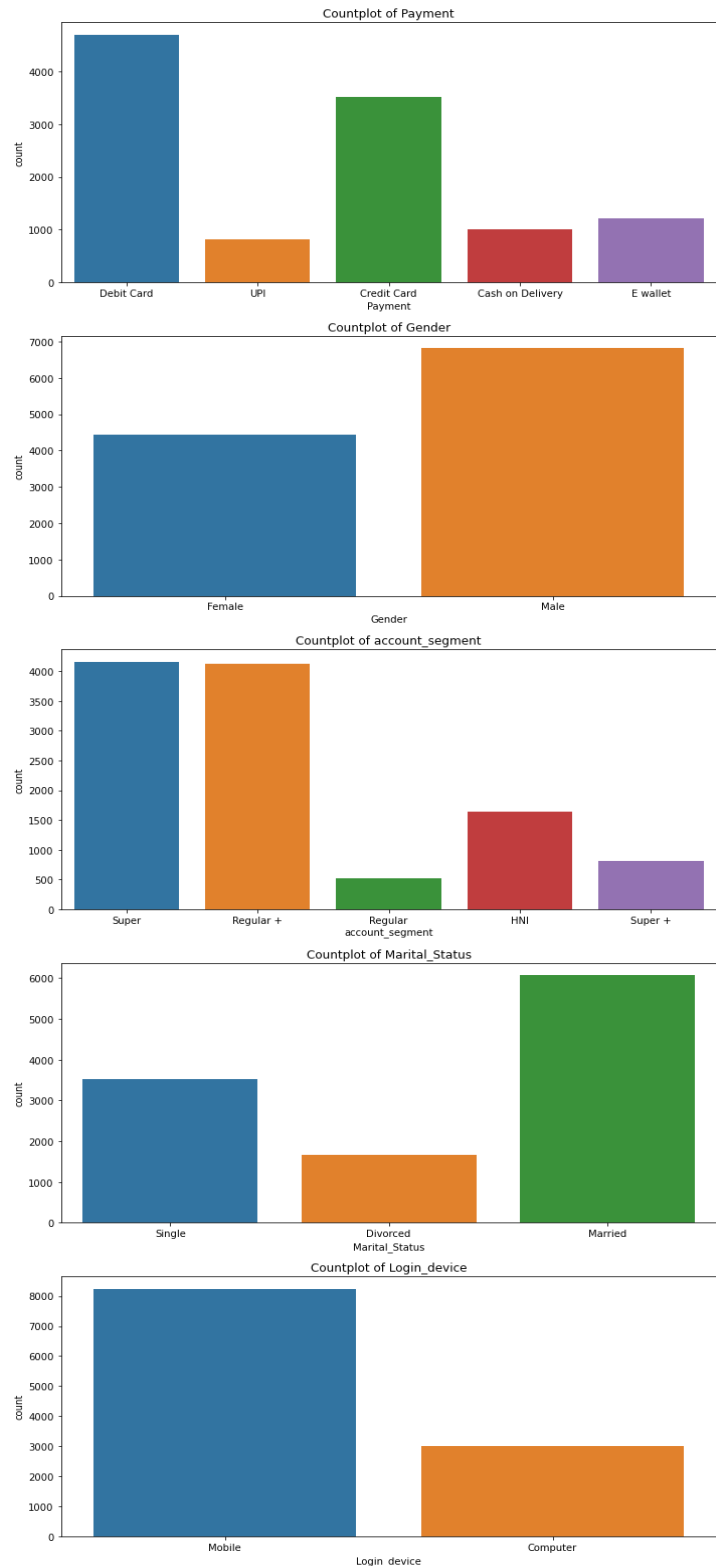


Fig. 07

Count plots of all discrete categorical variables:

- Non-churn customers are on majority.
- Customers from Tier-01 city are on majority.
- Customers of the account have given average rating i.e., 3 towards the company's service and same is the trend for towards customer service team also.
- Majority of the accounts have 4 customers users per the account.
- All the customers have not raised complaints towards the services.

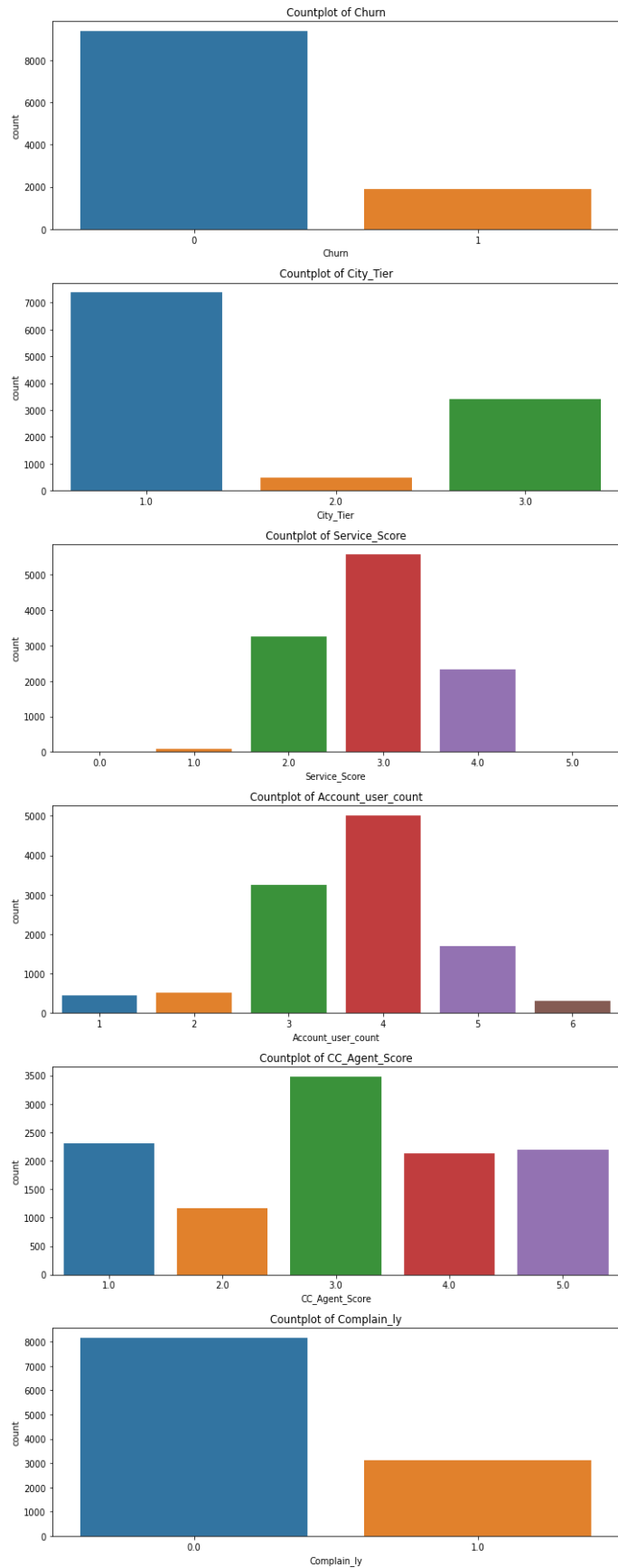


Fig. 08

Outlier treatment:

Box plots before outlier treatment (for continuous numerical variables only):

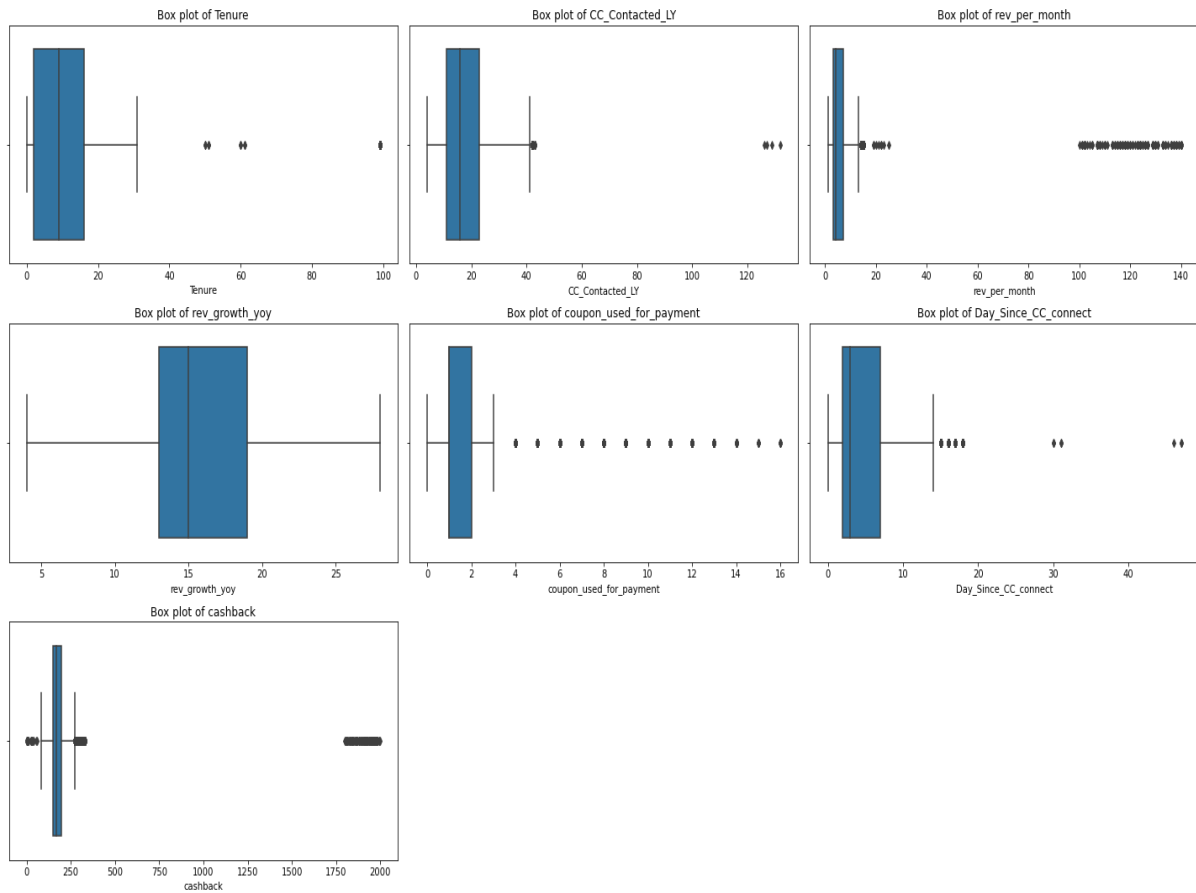


Fig. 09

- All the variables have outliers except 'rev_growth_yoy'.
- Let us treat the outlier by using IQR method.

Box plots after outlier treatment (for continuous numerical variables only):

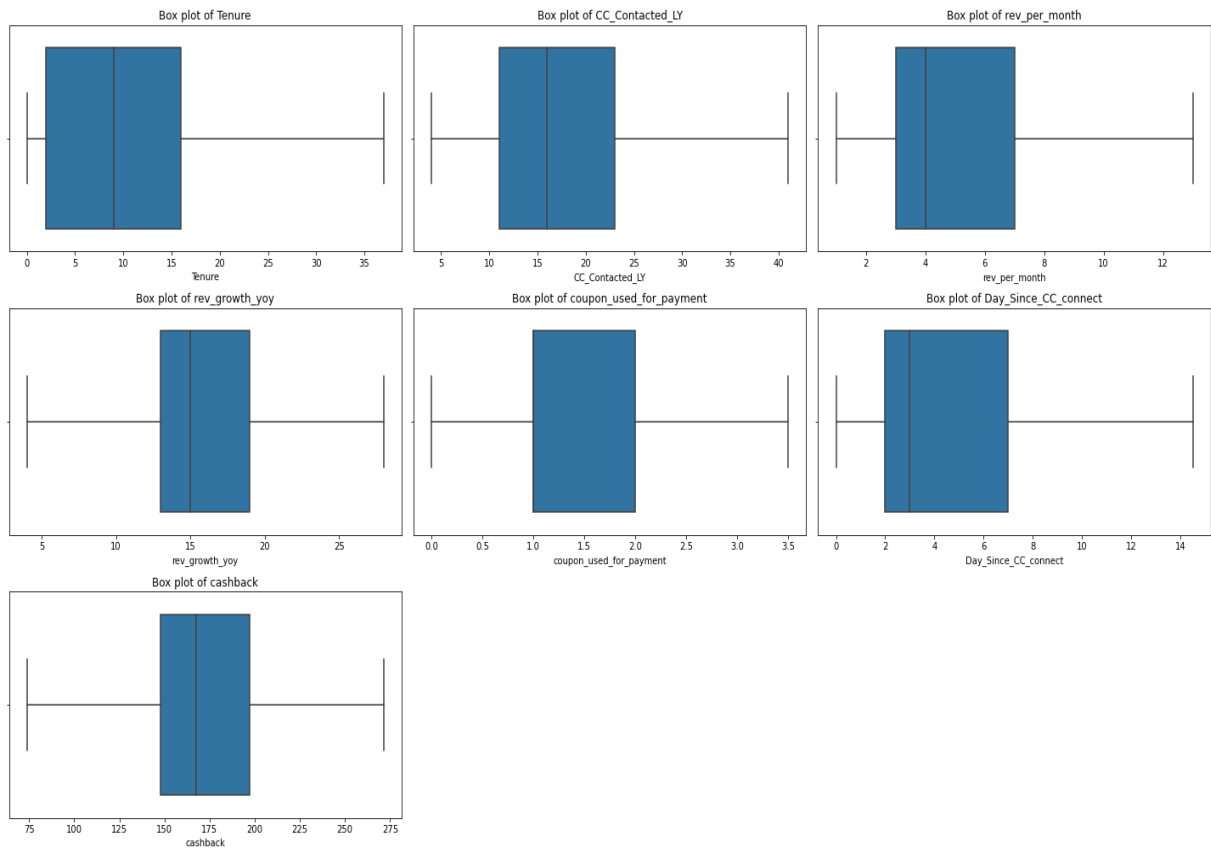


Fig. 10

- Outliers treated successfully.

Bivariate analysis:

Pair plot:

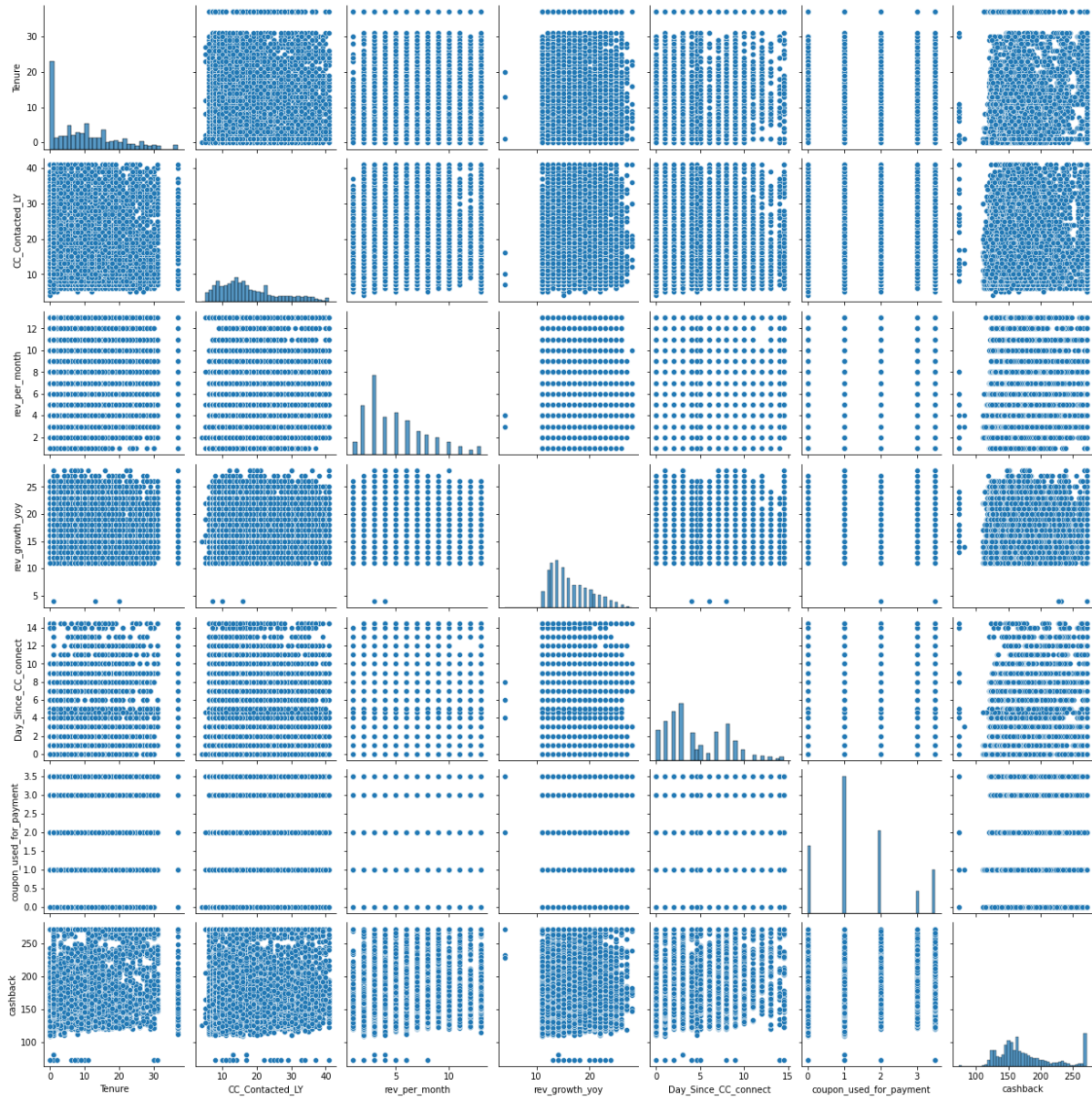


Fig. 11

- There are no correlations and patterns observed among the variables.

Heatmap:

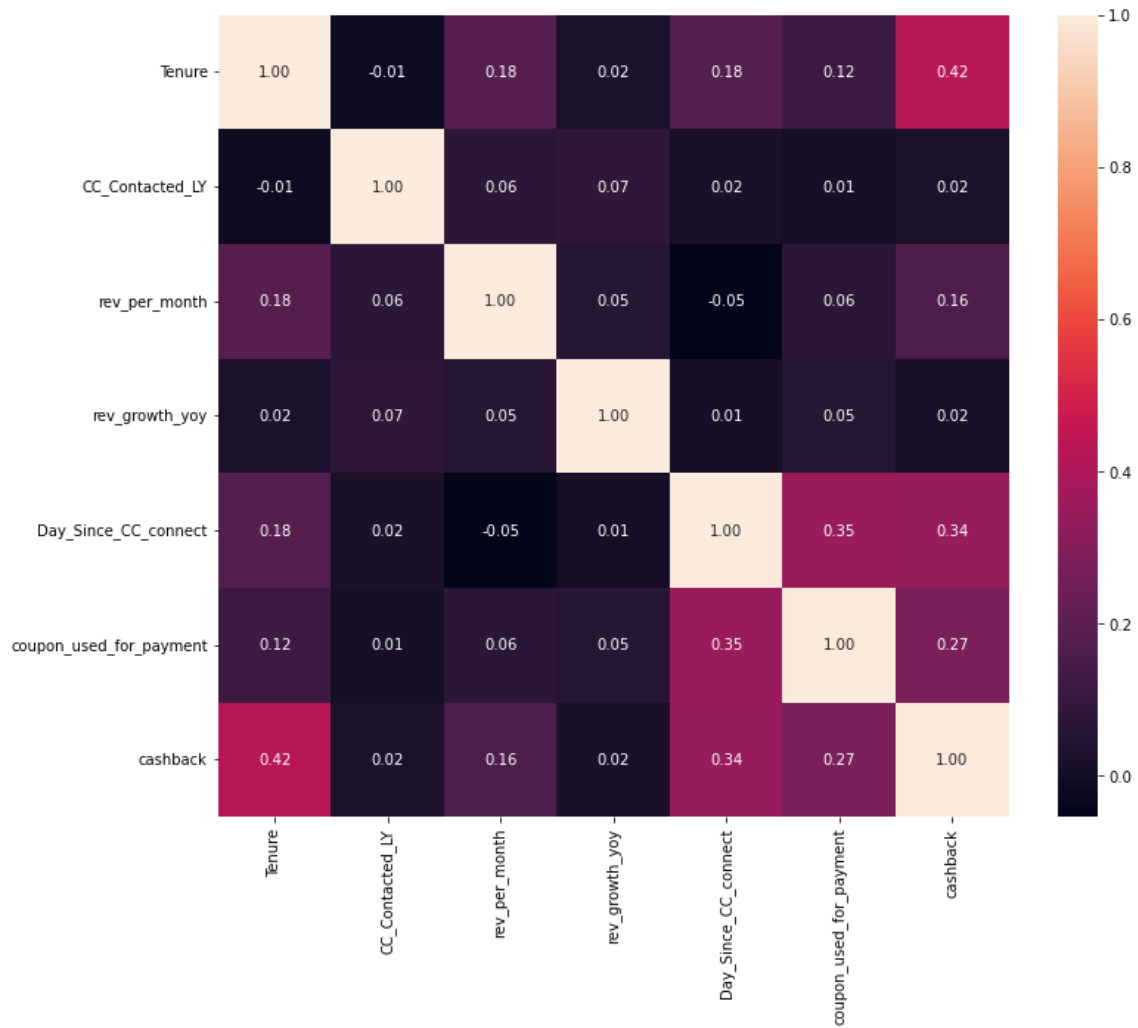


Fig. 12

- All the variables have very poor correlation among them.
- 0.42 is the highest correlation observed between 'cashback' & 'Tenure'

Multivariate Analysis:

Let us check the interaction b/w some important variables visually.

Churn vs Account_segment:

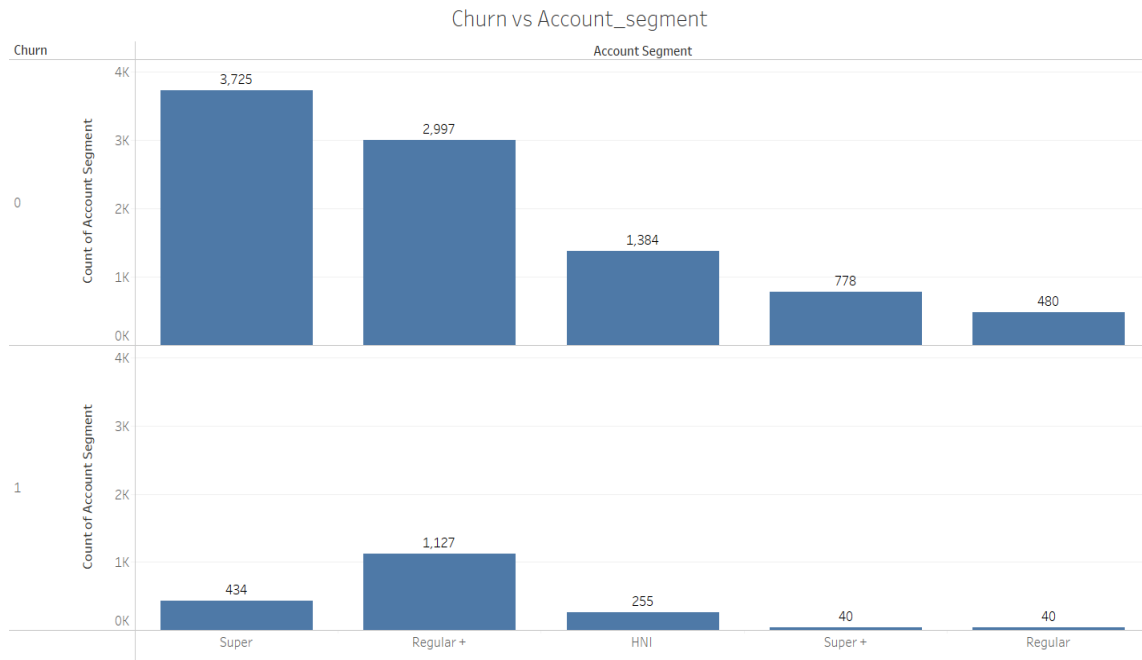


Fig. 13

- Customers are churning more from 'Regular +' account segment.

Churn vs CC_Agent_score:

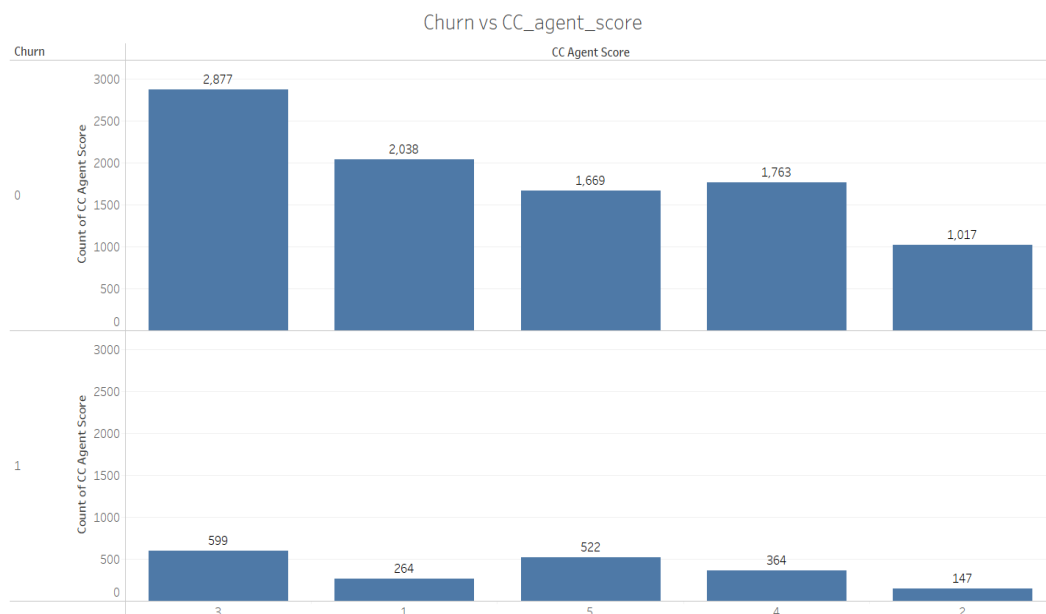


Fig. 14

- It is interesting to see that customers given 5 rating to customer care service are churning more compared to customers who given less rating.

Churn vs City_Tier:

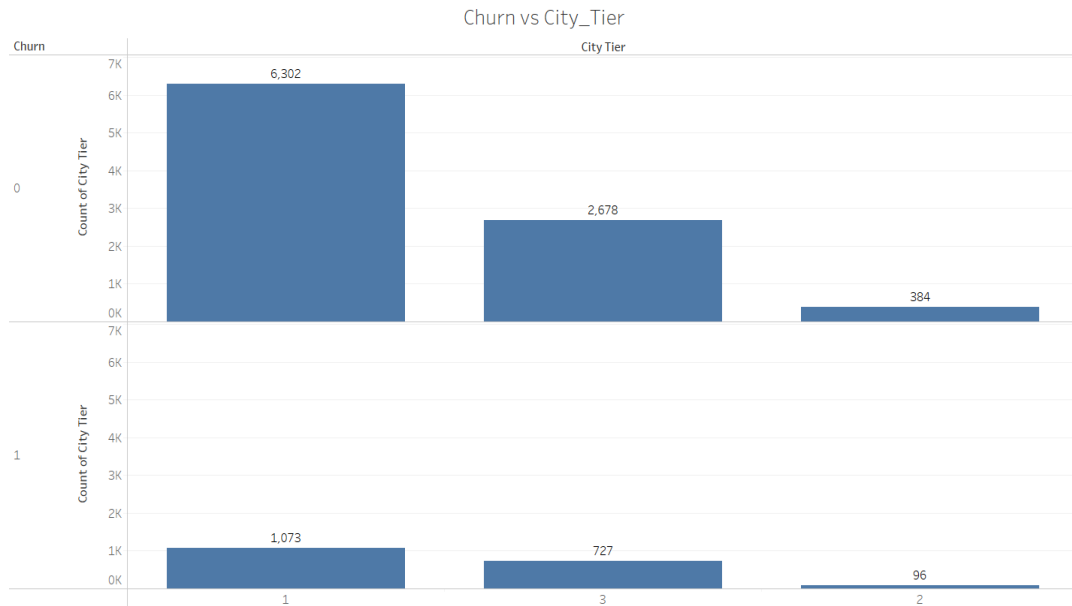


Fig. 15

- Customers are churning more from Tier-1 & 3 cities.

Churn vs Gender & Marital status:

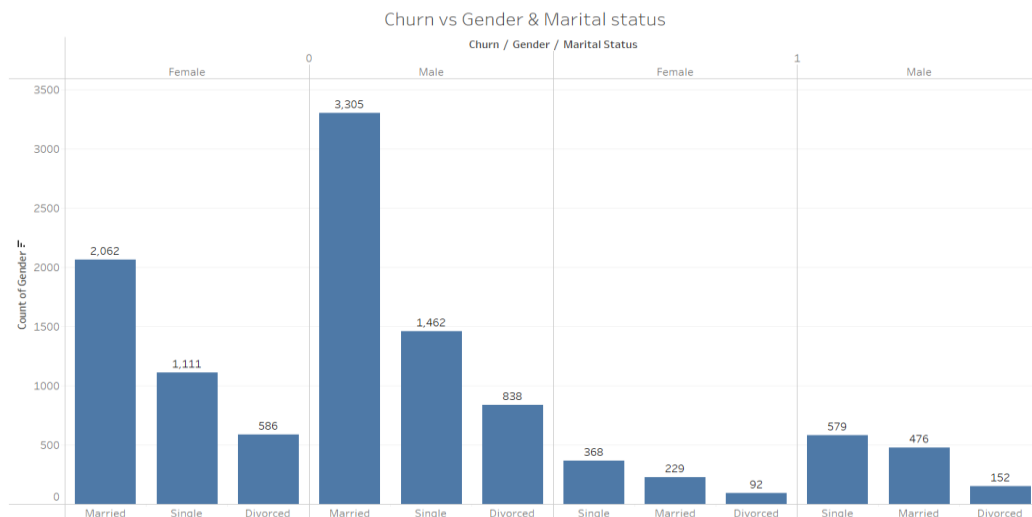


Fig. 16

- Customers with single marital status are churning more compared to married customers.
- But, married customers are the majority on non-churners side.

Churn vs Account_segment & CC_Contacted_LY:

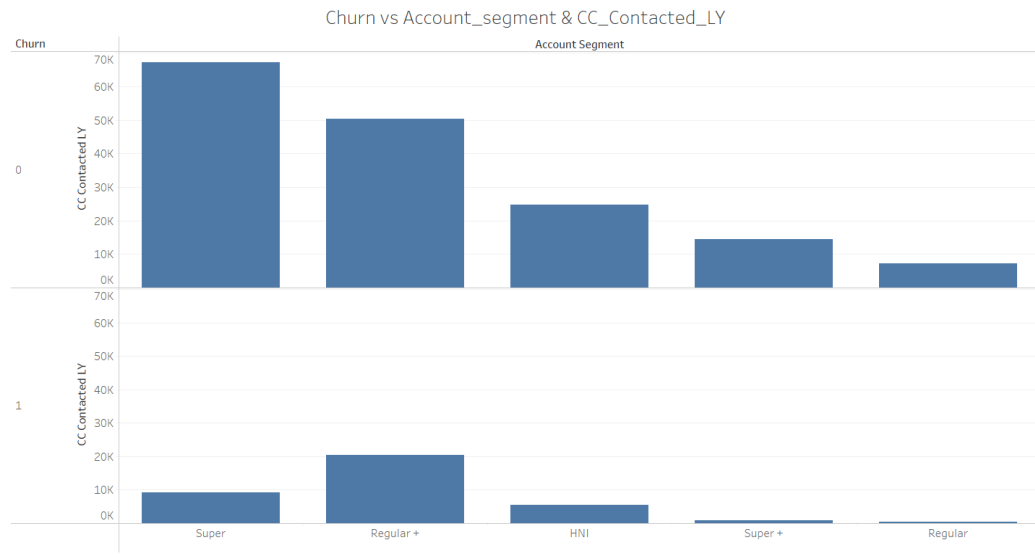


Fig. 17

- It looks like 'Regular +' segment customers are dissatisfied even though they contacted the customer care more times than others.

Revenue vs Account_user_count:

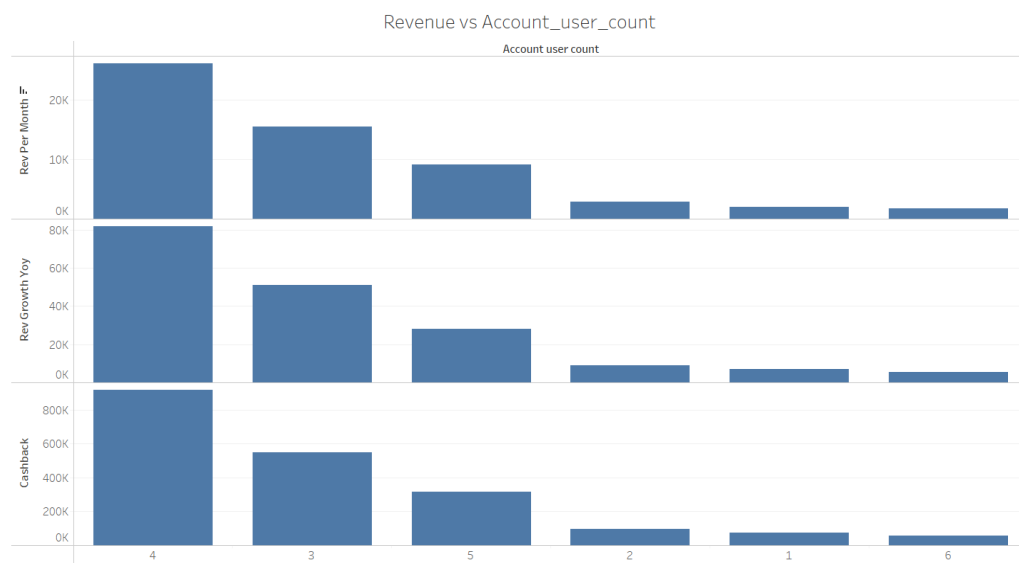


Fig. 18

- Accounts having 4 users are giving the highest revenue compared to other accounts.
- Revenue is low for the accounts even with 6 users.

Account_segment vs Tenure:

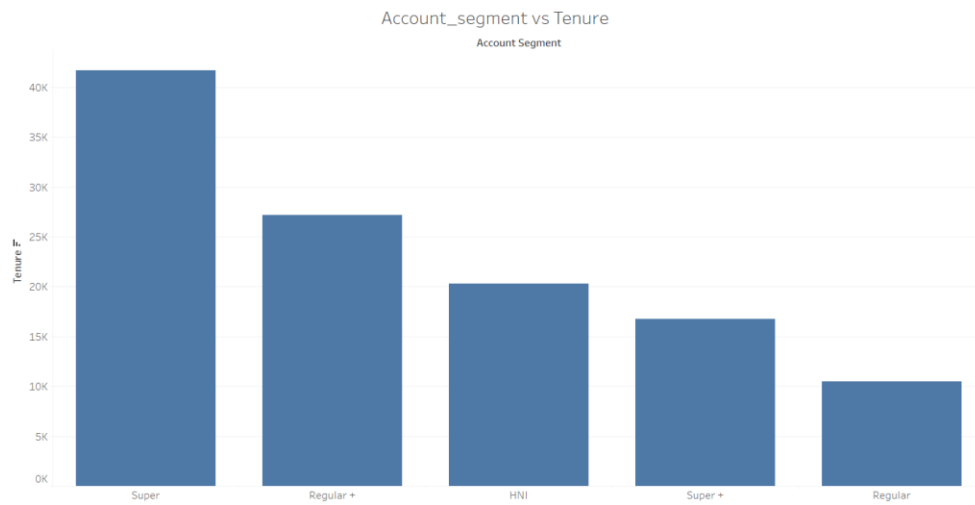


Fig. 19

- 'Super' and 'Regular +' segment accounts have long relationship with the company compared to other segments.

Account_segment vs Complaint_ly:

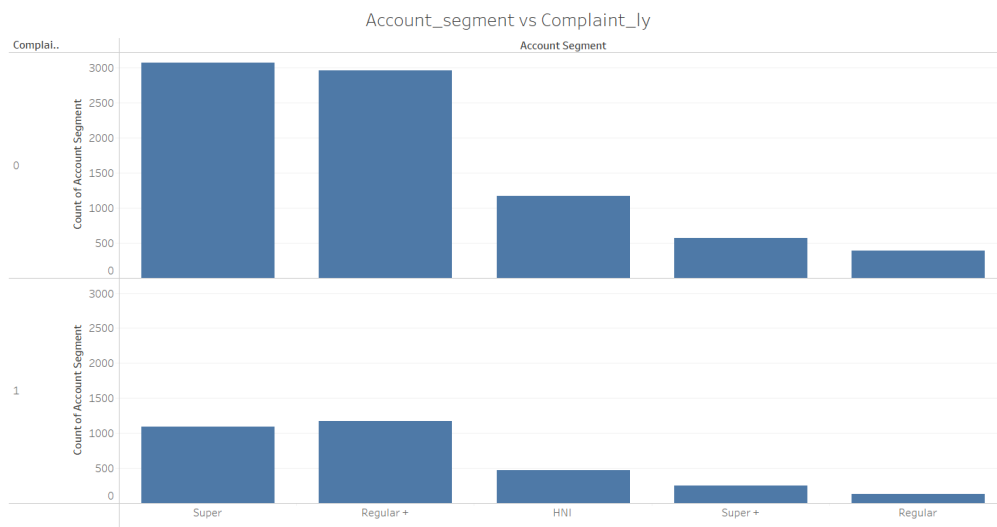


Fig. 20

- 'Super' and 'Regular +' segments account customers have raised the most complaints compared to others.
- 'Super' and 'Regular +' segments account customers are the most active customers both in terms of the generation of the revenue and raising complains to the company.

Churn vs Account_segment & Coupon_used_for_payment:

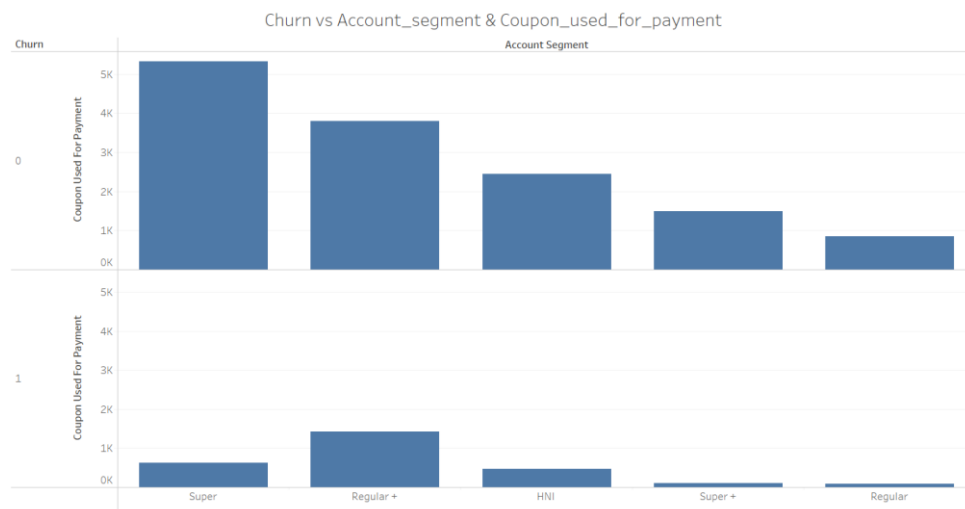


Fig. 21

- Being using most coupons for the payment, 'Regular +' customers are churning more compared to others.

Business insights:

Clustering and its business insights:

- Clustering done by k-Means
- 3 main clusters identified by WSS plot
- We divide these clusters based on 'Tenure':
Cluster-1: ~16 months (Old customers)
Cluster-2: ~9 months (Old to New customers)
Cluster-3: ~6 months (New customers)
- Old customers are gained more cashback compared to other segment customers.
- Old customers given good rating and satisfied towards the company services.
- Old customers only contributing more revenue per month to the company on an average compared to the other segment customers.

Data imbalance and its business context:

- We can have 3 degrees of data imbalance: Mild (20–40%), Moderate (1–20%), and Extreme (<1%)
- Let us check for the given data set's target variable data balance. Ratio of churn (1) and non-churn (0) are as shown below:

```
0    0.83
1    0.17
Name: Churn, dtype: float64
```

Fig. 22

- We have 83:17 ratio of data. So, the data is moderately imbalanced.

- Because of this, we lose potentially important information about churners required for the model building.
- Models which built may have accuracy issues which in turn will affect the prediction of churn.
- Incorrect predictions may create loss of the customers to the company which in turn will highly affect the revenue and profits.
- Ideal balanced dataset should be of 70:30 or at least 75:25.
- So, during model building if our current 83:17 ratio creates low accuracy scores. We can do over sample the target variable by using SMOTE technique and accuracy scores, prediction power of the model can be increased.

Other business insights:

Sample Data set description table:

	Churn	Tenure	City_Tier	CC_Contacted_LY	Service_Score	Account_user_count	CC_Agent_Score	rev_per_month	Complain_ly
count	11260.000000	11260.000000	11260.000000	11260.000000	11260.000000	11260.000000	11260.000000	11260.000000	11260.000000
mean	0.168384	10.343783	1.647425	17.796892	2.903375	3.704973	3.065808	5.110302	0.276288
std	0.374223	9.054847	0.912763	8.570074	0.722476	1.004383	1.372663	2.936656	0.447181
min	0.000000	0.000000	1.000000	4.000000	0.000000	1.000000	1.000000	1.000000	0.000000
25%	0.000000	2.000000	1.000000	11.000000	2.000000	3.000000	2.000000	3.000000	0.000000
50%	0.000000	9.000000	1.000000	16.000000	3.000000	4.000000	3.000000	4.000000	0.000000
75%	0.000000	16.000000	3.000000	23.000000	3.000000	4.000000	4.000000	7.000000	1.000000
max	1.000000	37.000000	3.000000	41.000000	5.000000	6.000000	5.000000	13.000000	1.000000

Table. 08

- ~11% of the accounts are having 0 months of the tenure. We can consider them as new customers' accounts.
- Mean tenure of the customers' accounts is ~11 months.
- 75% of the customers' accounts have tenure less than or equal to 16 months.
- Out of 3 tiers of cities available, ~65% of the customers are from the Tier-1 cities.
- Customers have called the company ~18 times in the last 12 months on an average.
- ~90% of the customers are preferring online payment mode, out of which, ~42% of the people doing payment via debit card only.
- On a rate of 1 to 5, Mean satisfaction score is 3, indicating that company services are meeting the customers expectations.
- Outstanding ratings (5) are very less, only 3 of the customers have the 5 rating.
- 4 users are there per account on an average.
- ~37% of the customers belong to Super, Regular + accounts' segments.
- HNI accounts' segment comprises of ~14% of the customers.
- On a rate of 1 to 5, 3 is mean rating given by the customers of the account for the company customer care service. ~70% of the customers are on meets expectations to exceed expectations side towards customer care service.
- Mean revenue per month is ~6.1.
- On an average, ~28% of the customers have reached to customer care in the last 12 months.
- Mean revenue growth percentage is ~16% (last 12 months vs last 24 to 13 month)
- On an average, customers have used coupons ~2 times to make the payment.

- Average monthly cashback per account is ~196.

4. Model building

Model building approach:

- Problem identification: It is a binary classification problem –whether an account will churn or not
- Pre-requisite EDA such as Data encoding, Data splitting, SMOTE for the train data set, Data scaling
- Building various classification models such as CART, Random Forest, Logistic Regression, LDA, KNN
- Interpretation of the performance metrics of the built classification models
- Tuning of the hyper-parameters for the above built classification models
- Interpretation of the performance metrics of the tuned classification models
- Ensemble modelling such as Bagging, AdaBoosting, GradientBoosting
- Interpretation of the performance metrics of the ensemble models
- Choosing the best and optimum model with based on performance metrics with maximum deviation of 10% between train and test results, focusing particularly on recall value
- Getting insights, Analysis of the important features and providing business recommendations to the company

➤ Pre-requisite EDA before model building:

Data Encoding:

- Dataset categorical variables encoded by using get.dummies method.

Sample data frame after encoding:

account_segment_Regular	account_segment_Regular +	account_segment_Super	account_segment_Super +	Marital_Status_Married	Marital_Status_Single
0	0	1	0	0	1
0	1	0	0	0	1
0	1	0	0	0	1
0	0	1	0	0	1
0	1	0	0	0	1

Table. 09

Data Splitting:

- Let us split the data into train and test set in 70:30 ratio.
- Shapes of splitted train and test sets are as shown below:

```
Shape of X1_train is (7882, 24)
Shape of X_test is (3378, 24)
Shape of y1_train is (7882, 1)
Shape of y_test is (3378, 1)
```

Fig. 23

- Let us check ratio of data after splitting

```
Split percentage of X1_train is 70.0
Split percentage of X_test is 30.0
Split percentage of y1_train is 70.0
Split percentage of y_test is 30.0
```

Fig. 24

- We can see that data successfully split into 70:30 train/test ratio.

SMOTE:

- Our data set is imbalanced with 83:17 non-churn (0)/churn (1) ratio. So, let us balance this ration by performing SMOTE technique on the train dataset before the model building.
- Shapes of spitted train and test sets after performing the SMOTE on train dataset:

```
Shape of X_train is (13110, 24)
Shape of X_test is (3378, 24)
Shape of y_train is (13110, 1)
Shape of y_test is (3378, 1)
```

Fig. 25

- We can see that train dataset shape increased from 7882 records to 13110.
- Let us check for the target variable's data balance after performing the SMOTE. Ratio of churn (1) and non-churn (0) are as shown below:

```
Churn
0      0.5
1      0.5
dtype: float64
```

Fig. 26

- We can see that we created extra records synthetically and balanced the dataset which is good for building the proper model with good performance metrics.

Data Scaling:

- Let us scale the latest data for the models such as KNN which is distance-based model. It is necessary to scale the data as the feature with a higher value range starts dominating when calculating distances.
- Sample train dataset after scaling:

	Tenure	City_Tier	CC_Contacted_LY	Service_Score	Account_user_count	CC_Agent_Score	rev_per_month	Complain_ly	rev_growth_yoy
0	0.079109	1.0	-0.745637	3.0	3	3.0	-1.441199	1.0	-0.558963
1	-0.626034	3.0	0.066090	2.0	3	4.0	-0.756489	0.0	-0.288212
2	0.549205	3.0	0.877818	3.0	3	5.0	-0.071779	0.0	-0.829715
3	2.194540	3.0	-0.977559	3.0	4	5.0	-0.756489	0.0	-1.100467
4	-0.743558	3.0	-0.281793	3.0	4	1.0	-0.756489	0.0	-0.288212

Table. 10

- Sample test dataset after scaling:

	Tenure	City_Tier	CC_Contacted_LY	Service_Score	Account_user_count	CC_Agent_Score	rev_per_month	Complain_ly	rev_growth_yoy
6888	-0.124840	1.0	0.261623	3.0	1	4.0	-1.073557	0.0	2.074920
467	1.095895	1.0	-1.132506	2.0	3	2.0	-1.073557	1.0	-1.394129
2347	-0.235816	1.0	0.726332	2.0	3	1.0	-0.729717	0.0	-1.394129
1794	0.208088	1.0	-0.435441	2.0	3	3.0	-1.073557	0.0	-1.394129
3125	-1.012647	1.0	0.493977	3.0	4	3.0	1.333319	1.0	-0.860429

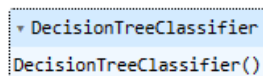
Table. 11

➤ Building various classification models:

Let us build the CART, Random Forest, KNN, Logistic Regression, LDA models

Decision Tree Classifier (CART):

- Let us fit the train dataset by using basic CART model.



```
from sklearn.tree import DecisionTreeClassifier
```

Fig. 27

- Hyperparameters used by the basic model are as shown below:

```
criterion='gini',
splitter='best',
max_depth=None,
min_samples_split=2,
min_samples_leaf=1,
```

Accuracy scores:

- For train dataset, 100%
- For test dataset, 92.65%

Prediction and Model evaluation:

- Train and test datasets are predicted using basic CART model.
- Performance metrics and Model evaluation are shown below:

Train dataset:

- Confusion matrix:

```
array([[6555,    0],
       [    0, 6555]], dtype=int64)
```

Fig. 28

- Classification report table:

	precision	recall	f1-score	support
0	1.0	1.0	1.0	6555.0
1	1.0	1.0	1.0	6555.0
accuracy	1.0	1.0	1.0	1.0
macro avg	1.0	1.0	1.0	13110.0
weighted avg	1.0	1.0	1.0	13110.0

Table. 12

- Accuracy score is 100.00%
- ROC_AUC score is 100.00%
- ROC curve:

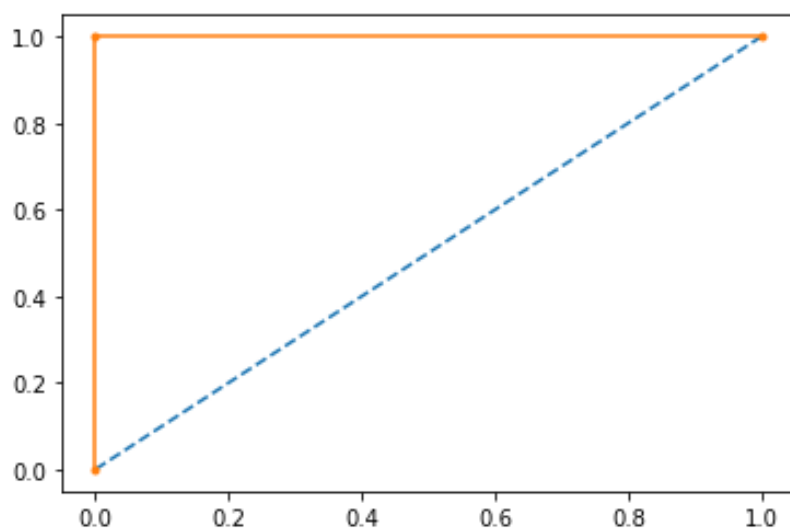


Fig. 29

Test dataset:

- Confusion matrix:

```
array([[2654, 155],
       [ 93, 476]], dtype=int64)
```

Fig. 30

- Classification report:

	precision	recall	f1-score	support
0	0.966145	0.944820	0.955364	2809.000000
1	0.754358	0.836555	0.793333	569.000000
accuracy	0.926584	0.926584	0.926584	0.926584
macro avg	0.860252	0.890688	0.874348	3378.000000
weighted avg	0.930471	0.926584	0.928071	3378.000000

Table. 13

- Accuracy score is 92.65%
- ROC_AUC score is 0.891
- ROC curve:

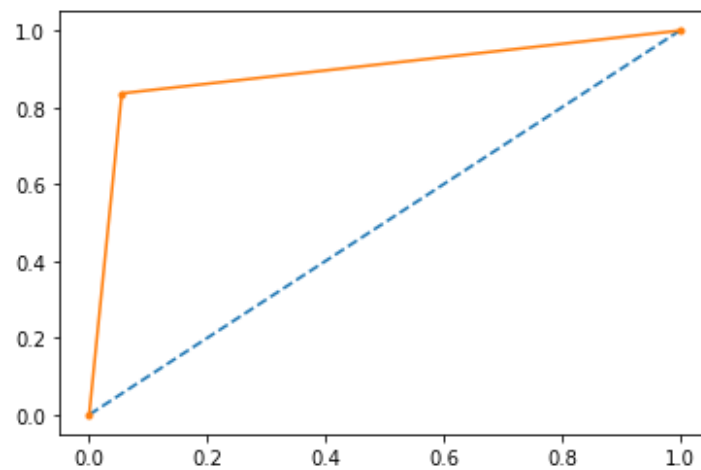


Fig. 31

Feature importance:

- Top important features for CART building are as shown below:

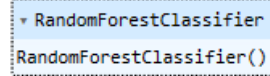
Tenure	38.992702
Complain_ly	14.273146
CC_Agent_Score	6.169920
CC_Contacted_LY	4.924384
rev_per_month	3.930984
cashback	3.815659
Day_Since_CC_connect	3.670967
City_Tier	3.542022
Marital_Status_Married	3.529643

Table. 14

- Tenure and Complain_ly are the top two features helping in predicting the churning of the customers.

Random Forest:

- Let us fit the train dataset by using basic Random Forest model.



```
from sklearn.ensemble import  
RandomForestClassifier()
```

Fig. 32

- Hyperparameters used by the basic model are as shown below:

```
n_estimators=100,  
criterion='gini',  
max_depth=None,  
min_samples_split=2,  
min_samples_leaf=1,  
max_features='sqrt',  
bootstrap=True,  
oob_score=False
```

Accuracy scores:

- For train dataset, 100%
- For test dataset, 97.18%

Prediction and Model evaluation:

- Train and test datasets are predicted using basic RF model.
- Performance metrics and Model evaluation are shown below:

Train dataset:

- Confusion matrix:

```
array([[6555,    0],  
       [    0, 6555]], dtype=int64)
```

Fig. 33

- Classification report table:

	precision	recall	f1-score	support
0	1.0	1.0	1.0	6555.0
1	1.0	1.0	1.0	6555.0
accuracy	1.0	1.0	1.0	1.0
macro avg	1.0	1.0	1.0	13110.0
weighted avg	1.0	1.0	1.0	13110.0

Table. 15

- Accuracy score is 100.00%
- ROC_AUC score is 100.00%
- ROC curve:

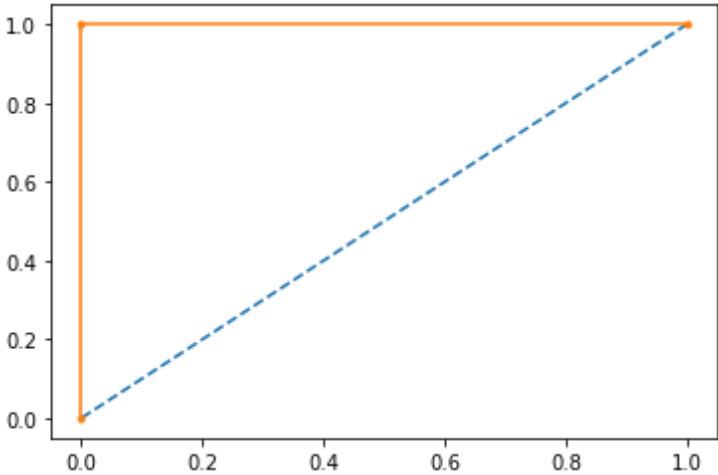


Fig. 34

Test dataset:

- Confusion matrix:

```
array([[2783,  26],
       [  69, 500]], dtype=int64)
```

Fig. 35

- Classification report:

	precision	recall	f1-score	support
0	0.975808	0.990744	0.983219	2809.000000
1	0.950570	0.878735	0.913242	569.000000
accuracy	0.971877	0.971877	0.971877	0.971877
macro avg	0.963188	0.934739	0.948230	3378.000000
weighted avg	0.971556	0.971877	0.971431	3378.000000

Table. 16

- Accuracy score is 97.18%
- ROC_AUC score is 0.991
- ROC curve:

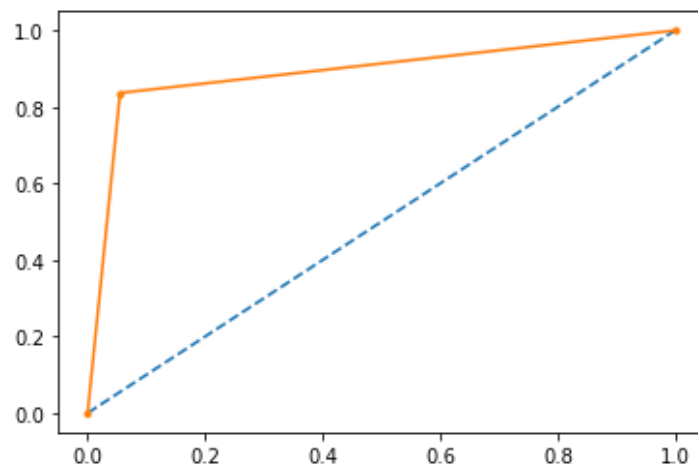


Fig. 36

Feature importance:

- Top important features for RF building are as shown below:

Tenure	26.292520
Complain_ly	11.524112
cashback	6.009295
CC_Agent_Score	5.839571
Day_Since_CC_connect	5.562774
Marital_Status_Married	5.390393
CC_Contacted_LY	4.622098
rev_per_month	4.512067
rev_growth_yoy	4.221254

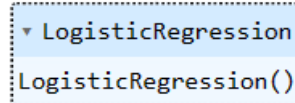
Table. 17

- Tenure and Complain_ly are the top two features helping in predicting the churning of the customers which is same as CART model.

Logistic Regression:

Scikit model:

- Let us fit the train dataset by using basic Logistic Regression model.



```
LogisticRegression()  
LogisticRegression()
```

Fig. 37

- Hyperparameters used by the basic model are as shown below:

```
penalty='l2',  
tol=0.0001,  
C=1.0,  
max_iter=100
```

Accuracy scores:

- For train dataset, 84.79%
- For test dataset, 81.40%

Prediction and Model evaluation:

- Train and test datasets are predicted using basic Logistic Regression model.
- Performance metrics and Model evaluation are shown below:

Train dataset:

- Confusion matrix:

```
array([[5635,  920],  
       [ 930, 5625]], dtype=int64)
```

Fig. 38

- Classification report table:

	precision	recall	f1-score	support
0	0.858340	0.859649	0.858994	6555.000000
1	0.859435	0.858124	0.858779	6555.000000
accuracy	0.858886	0.858886	0.858886	0.858886
macro avg	0.858887	0.858886	0.858886	13110.000000
weighted avg	0.858887	0.858886	0.858886	13110.000000

Table. 18

- Accuracy score is 85.88%
- ROC_AUC score is 0.918%
- ROC curve:

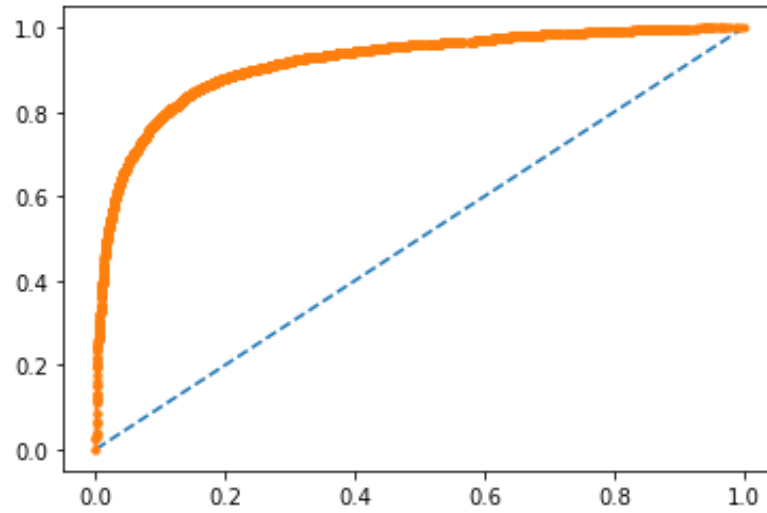


Fig. 39

Test dataset:

- Confusion matrix:

```
array([[2414, 395],
       [ 171, 398]], dtype=int64)
```

Fig. 40

- Classification report:

	precision	recall	f1-score	support
0	0.933849	0.859381	0.895069	2809.000000
1	0.501892	0.699473	0.584435	569.000000
accuracy	0.832445	0.832445	0.832445	0.832445
macro avg	0.717870	0.779427	0.739752	3378.000000
weighted avg	0.861089	0.832445	0.842745	3378.000000

Table. 19

- Accuracy score is 83.24%
- ROC_AUC score is 0.852
- ROC curve:

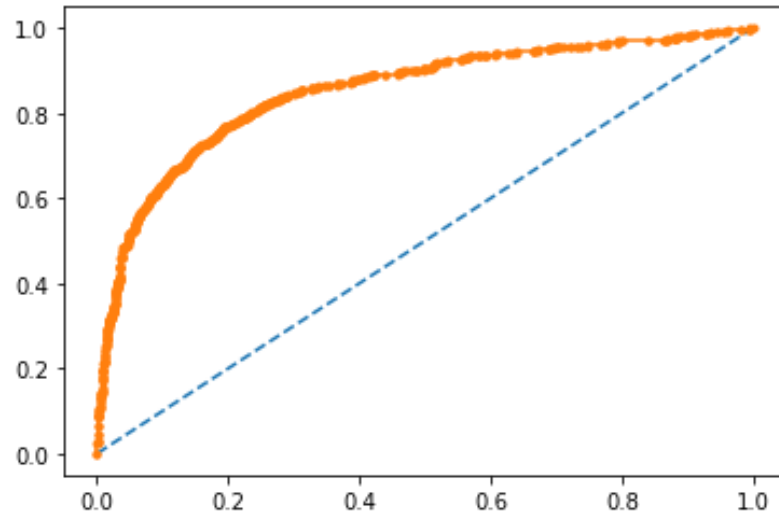


Fig. 41

Statsmodel:

- Statsmodel is built on the train dataset.
- Logistic model parameters are as shown below:

Intercept	0.786024
Tenure	-0.019031
City_Tier	0.051699
CC_Contacted_LY	0.004386
Service_Score	0.002501
Account_user_count	0.033832
CC_Agent_Score	0.029427
rev_per_month	0.017629
Complain_ly	0.215888
rev_growth_yoy	-0.003671
coupon_used_for_payment	0.016666
Day_Since_CC_connect	-0.009912
cashback	-0.001177
Payment_Credit_Card	-0.294027
Payment_Debit_Card	-0.260163
Payment_E_wallet	-0.205826
Payment_UPI	-0.275297
Gender_Male	-0.004039
account_segment_Regular	0.103544
account_segment_Regular_plus	-0.078688
account_segment_Super	-0.255078
account_segment_Super_plus	-0.024911
Marital_Status_Married	-0.197377
Marital_Status_Single	-0.045726
Login_device_Mobile	-0.079083
dtype:	float64

Table. 20

- Logistic model regression results are as shown below:

OLS Regression Results						
Dep. Variable:	Churn	R-squared:	0.529			
Model:	OLS	Adj. R-squared:	0.528			
Method:	Least Squares	F-statistic:	613.2			
Date:	Fri, 17 Mar 2023	Prob (F-statistic):	0.00			
Time:	06:49:56	Log-Likelihood:	-4575.0			
No. Observations:	13110	AIC:	9200.			
Df Residuals:	13085	BIC:	9387.			
Df Model:	24					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
Intercept	0.7860	0.033	24.133	0.000	0.722	0.850
Tenure	-0.0190	0.000	-45.010	0.000	-0.020	-0.018
City_Tier	0.0517	0.004	12.941	0.000	0.044	0.060
CC_Contacted_LY	0.0044	0.000	12.302	0.000	0.004	0.005
Service_Score	0.0025	0.005	0.500	0.617	-0.007	0.012
Account_user_count	0.0338	0.003	10.021	0.000	0.027	0.040
CC_Agent_Score	0.0294	0.002	12.566	0.000	0.025	0.034
rev_per_month	0.0176	0.001	16.439	0.000	0.016	0.020
Complain_ly	0.2159	0.007	32.161	0.000	0.203	0.229
rev_growth_yoy	-0.0037	0.001	-4.437	0.000	-0.005	-0.002
coupon_used_for_payment	0.0167	0.004	4.521	0.000	0.009	0.024
Day_Since_CC_connect	-0.0099	0.001	-9.377	0.000	-0.012	-0.008
cashback	-0.0012	0.000	-8.278	0.000	-0.001	-0.001
Payment_Credit_Card	-0.2940	0.009	-32.342	0.000	-0.312	-0.276
Payment_Debit_Card	-0.2602	0.008	-31.673	0.000	-0.276	-0.244
Payment_E_wallet	-0.2058	0.013	-16.145	0.000	-0.231	-0.181
Payment_UPI	-0.2753	0.014	-19.158	0.000	-0.303	-0.247
Gender_Male	-0.0040	0.006	-0.657	0.511	-0.016	0.008
account_segment_Regular	0.1035	0.020	5.058	0.000	0.063	0.144
account_segment_Regular_plus	-0.0787	0.011	-6.845	0.000	-0.101	-0.056
account_segment_Super	-0.2551	0.010	-25.677	0.000	-0.275	-0.236
account_segment_Super_plus	-0.0249	0.018	-1.410	0.158	-0.060	0.010
Marital_Status_Married	-0.1974	0.008	-25.072	0.000	-0.213	-0.182
Marital_Status_Single	-0.0457	0.008	-5.694	0.000	-0.061	-0.030
Login_device_Mobile	-0.0791	0.006	-12.225	0.000	-0.092	-0.066
Omnibus:	75.774	Durbin-Watson:	1.545			
Prob(Omnibus):	0.000	Jarque-Bera (JB):	68.532			
Skew:	0.136	Prob(JB):	1.31e-15			
Kurtosis:	2.773	Cond. No.	2.05e+03			

Fig. 42

- It can be seen that 'Service_Score', 'Gender_Male', 'account_segment_Super_plus' variables have p-value greater than 0.05, indicating these as insignificant variables.
- We will remove the these variables and built the model as next step.
- Logistic model parameters after removing the insignificant variables are as shown below:

Intercept	0.798361
Tenure	-0.019153
City_Tier	0.051604
CC_Contacted_LY	0.004402
Account_user_count	0.034878
CC_Agent_Score	0.029438
rev_per_month	0.017696
Complain_ly	0.215800
rev_growth_yoy	-0.003635
coupon_used_for_payment	0.017391
Day_Since_CC_connect	-0.009907
cashback	-0.001253
Payment_Credit_Card	-0.295287
Payment_Debit_Card	-0.261415
Payment_E_wallet	-0.206592
Payment_UPI	-0.277229
account_segment_Regular	0.113704
account_segment_Regular_plus	-0.079194
account_segment_Super	-0.252898
Marital_Status_Married	-0.198002
Marital_Status_Single	-0.046439
Login_device_Mobile	-0.078814
dtype:	float64

Table. 21

- Logistic model regression results after removing the insignificant variables are as shown below:

OLS Regression Results						
Dep. Variable:	Churn	R-squared:	0.529			
Model:	OLS	Adj. R-squared:	0.528			
Method:	Least Squares	F-statistic:	700.7			
Date:	Fri, 17 Mar 2023	Prob (F-statistic):	0.00			
Time:	06:56:43	Log-Likelihood:	-4576.5			
No. Observations:	13110	AIC:	9197.			
Df Residuals:	13088	BIC:	9361.			
Df Model:	21					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
Intercept	0.7984	0.031	25.445	0.000	0.737	0.860
Tenure	-0.0192	0.000	-46.167	0.000	-0.020	-0.018
City_Tier	0.0516	0.004	12.927	0.000	0.044	0.059
CC_Contacted_LY	0.0044	0.000	12.364	0.000	0.004	0.005
Account_user_count	0.0349	0.003	10.739	0.000	0.029	0.041
CC_Agent_Score	0.0294	0.002	12.610	0.000	0.025	0.034
rev_per_month	0.0177	0.001	16.562	0.000	0.016	0.020
Complain_ly	0.2158	0.007	32.163	0.000	0.203	0.229
rev_growth_yoy	-0.0036	0.001	-4.409	0.000	-0.005	-0.002
coupon_used_for_payment	0.0174	0.004	4.830	0.000	0.010	0.024
Day_Since_CC_connect	-0.0099	0.001	-9.373	0.000	-0.012	-0.008
cashback	-0.0013	0.000	-9.965	0.000	-0.002	-0.001
Payment_Credit_Card	-0.2953	0.009	-32.612	0.000	-0.313	-0.278
Payment_Debit_Card	-0.2614	0.008	-31.982	0.000	-0.277	-0.245
Payment_E_wallet	-0.2066	0.013	-16.215	0.000	-0.232	-0.182
Payment_UPI	-0.2772	0.014	-19.381	0.000	-0.305	-0.249
account_segment_Regular	0.1137	0.019	6.035	0.000	0.077	0.151
account_segment_Regular_plus	-0.0792	0.011	-7.019	0.000	-0.101	-0.057
account_segment_Super	-0.2529	0.010	-25.875	0.000	-0.272	-0.234
Marital_Status_Married	-0.1980	0.008	-25.221	0.000	-0.213	-0.183
Marital_Status_Single	-0.0464	0.008	-5.794	0.000	-0.062	-0.031
Login_device_Mobile	-0.0788	0.006	-12.202	0.000	-0.091	-0.066
Omnibus:	74.834	Durbin-Watson:	1.544			
Prob(Omnibus):	0.000	Jarque-Bera (JB):	68.378			
Skew:	0.138	Prob(JB):	1.42e-15			
Kurtosis:	2.780	Cond. No.	1.95e+03			

Fig. 43

- It can be seen that all the variables have p-value less than 0.05.
- We can say that all the variables are significant for the logistic model except 'Service_Score', 'Gender_Male', 'account_segment_Super_plus'.

Linear Discriminant Analysis:

- Let us fit the train dataset by using basic LDA model.

```
LinearDiscriminantAnalysis()
LinearDiscriminantAnalysis()
```

Fig. 44

- Hyperparameters used by the basic model are as shown below:

```
solver='svd',
tol=0.0001
```

Accuracy scores:

- For train dataset, 85.74%
- For test dataset, 83.42%

Prediction and Model evaluation:

- Train and test datasets are predicted using basic LDA model.

- Performance metrics and Model evaluation are shown below:

Train dataset:

- Confusion matrix:

```
array([[5626,  929],
       [ 940, 5615]], dtype=int64)
```

Fig. 45

- Classification report table:

	precision	recall	f1-score	support
0	0.856838	0.858276	0.857557	6555.000000
1	0.858038	0.856598	0.857317	6555.000000
accuracy	0.857437	0.857437	0.857437	0.857437
macro avg	0.857438	0.857437	0.857437	13110.000000
weighted avg	0.857438	0.857437	0.857437	13110.000000

Table. 22

- Accuracy score is 85.74%
- ROC_AUC score is 0.930%
- ROC curve:

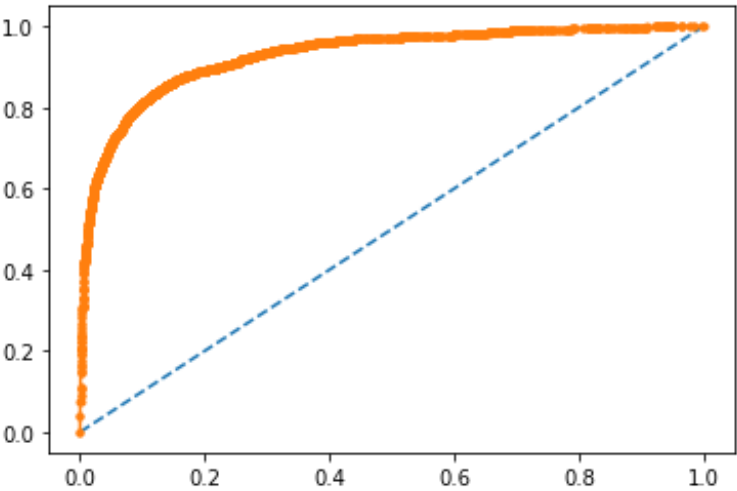


Fig. 46

Test dataset:

- Confusion matrix:

```
array([[2426,  383],
       [ 177,  392]], dtype=int64)
```

Fig. 47

- Classification report:

	precision	recall	f1-score	support
0	0.932002	0.863653	0.896526	2809.000000
1	0.505806	0.688928	0.583333	569.000000
accuracy	0.834221	0.834221	0.834221	0.834221
macro avg	0.718904	0.776290	0.739930	3378.000000
weighted avg	0.860212	0.834221	0.843771	3378.000000

Table. 23

- Accuracy score is 83.42%
- ROC_AUC score is 0.854
- ROC curve:

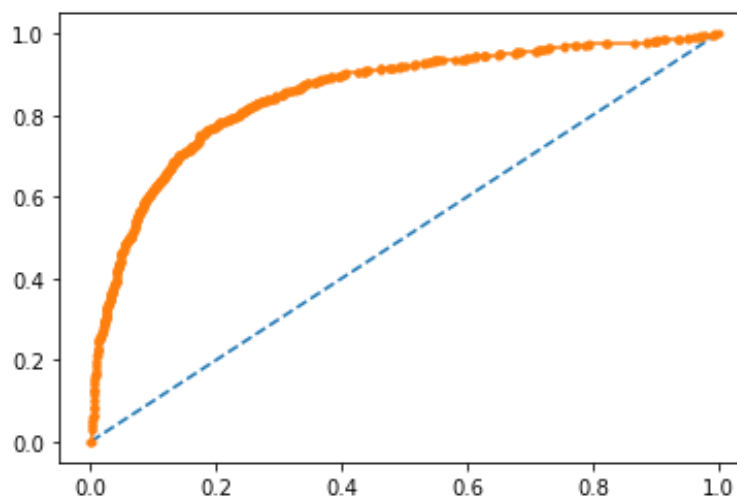


Fig. 48

K-Nearest Neighbors:

- Let us fit the train dataset by using basic KNN model.

```
KNeighborsClassifier
KNeighborsClassifier(n_neighbors=2)
```

Fig. 49

- Hyperparameters used by the basic model are as shown below:

```
n_neighbors=2,
weights='uniform',
algorithm='auto',
leaf_size=30,
p=2,
metric='minkowski'
```

Accuracy scores:

- For train dataset, 99.87%
- For test dataset, 96.50%

Prediction and Model evaluation:

- Train and test datasets are predicted using KNN model.
- Performance metrics and Model evaluation are shown below:

Train dataset:

- Confusion matrix:

```
array([[6555,    0],
       [   17, 6538]], dtype=int64)
```

Fig. 50

- Classification report table:

	precision	recall	f1-score	support
0	0.997413	1.000000	0.998705	6555.000000
1	1.000000	0.997407	0.998702	6555.000000
accuracy	0.998703	0.998703	0.998703	0.998703
macro avg	0.998707	0.998703	0.998703	13110.000000
weighted avg	0.998707	0.998703	0.998703	13110.000000

Table. 24

- Accuracy score is 99.87%
- ROC_AUC score is 100.00%
- ROC curve:

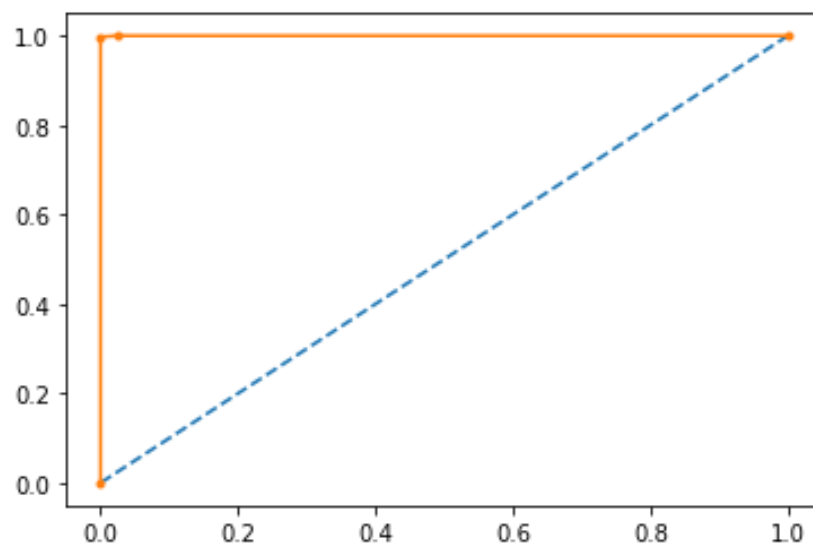


Fig. 51

Test dataset:

- Confusion matrix:

```
array([[2746, 63],  
       [ 55, 514]], dtype=int64)
```

Fig. 52

- Classification report:

	precision	recall	f1-score	support
0	0.980364	0.977572	0.978966	2809.000000
1	0.890815	0.903339	0.897033	569.000000
accuracy	0.965068	0.965068	0.965068	0.965068
macro avg	0.935589	0.940456	0.938000	3378.000000
weighted avg	0.965280	0.965068	0.965165	3378.000000

Table. 25

- Accuracy score is 96.50%
- ROC_AUC score is 0.960
- ROC curve:

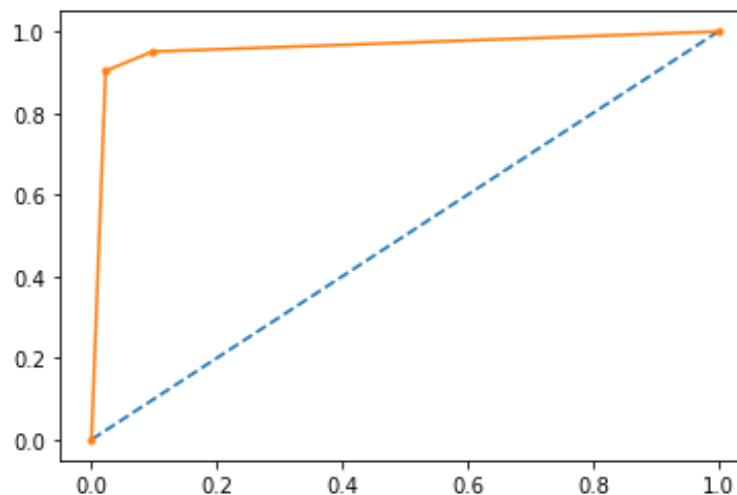


Fig. 53

➤ **Tuning of the hyper-parameters for the above built classification models**

CART:

- Let us do the grid search on the train dataset.

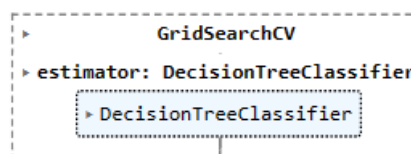


Fig. 54

- Best parameters and estimator after grid search are as below:

```
{'class_weight': 'balanced', 'max_leaf_nodes': 3000, 'min_samples_leaf': 1, 'min_samples_split': 3}
```

Fig. 55

Accuracy scores and validity after grid search:

- For train dataset, 99.88%
- For test dataset, 92.59%
- Accuracy score is not improved for both the train and test datasets after tuning the model using grid search.

Prediction and Model evaluation:

- Train and test datasets are predicted using tuned CART model.
- Performance metrics and Model evaluation are shown below:

Train dataset:

- Confusion matrix:

```
array([[6555,    0],
       [    0, 6555]], dtype=int64)
```

Fig. 56

- Classification report table:

	precision	recall	f1-score	support
0	1.0	1.0	1.0	6555.0
1	1.0	1.0	1.0	6555.0
accuracy	1.0	1.0	1.0	1.0
macro avg	1.0	1.0	1.0	13110.0
weighted avg	1.0	1.0	1.0	13110.0

Table. 26

- Accuracy score is 100.00%
- ROC_AUC score is 100.00%
- ROC curve:

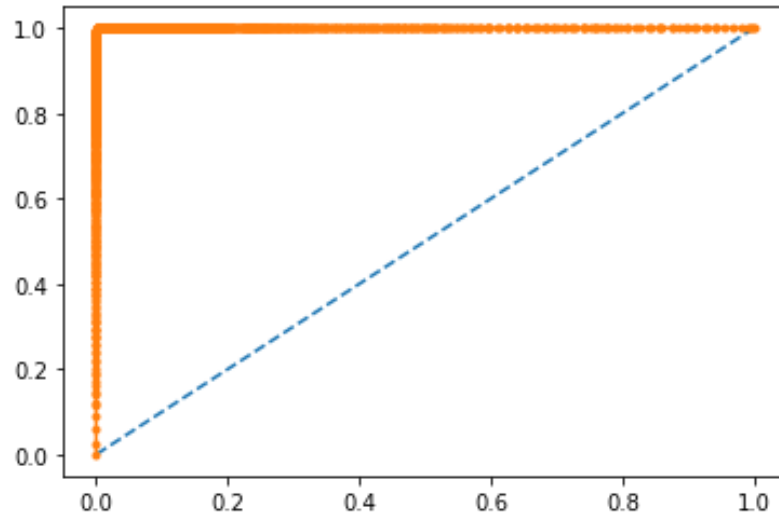


Fig. 57

Test dataset:

- Confusion matrix:

```
array([[2633, 176],
       [ 139, 430]], dtype=int64)
```

Fig. 58

- Classification report:

	precision	recall	f1-score	support
0	0.949856	0.937344	0.943559	2809.00000
1	0.709571	0.755712	0.731915	569.00000
accuracy	0.906750	0.906750	0.906750	0.90675
macro avg	0.829713	0.846528	0.837737	3378.00000
weighted avg	0.909381	0.906750	0.907909	3378.00000

Table. 27

- Accuracy score is 90.67%
- ROC_AUC score is 0.888
- ROC curve:

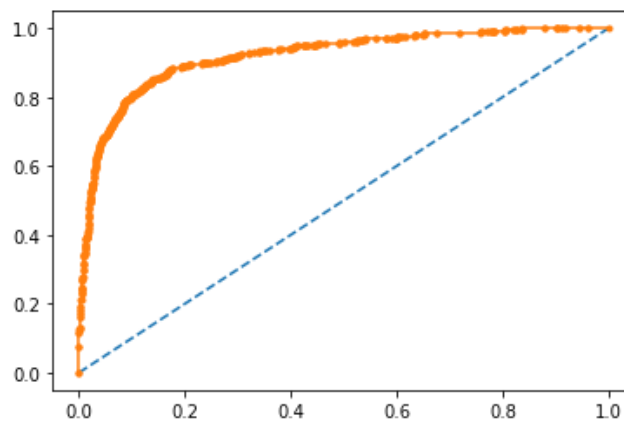


Fig. 59

Random Forest:

- Let us do the grid search on the train dataset.

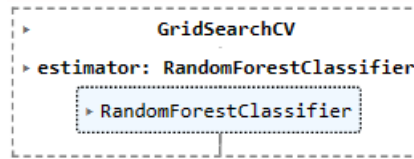


Fig. 60

- Best parameters and estimator after grid search are as below:

```
{'class_weight': 'balanced', 'criterion': 'log_loss', 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 300}
```

Fig. 61

Accuracy scores and validity after grid search:

- For train dataset, 100.00%
- For test dataset, 97.18%
- Accuracy score is not improved for both the train and test datasets after tuning the model using grid search.

Prediction and Model evaluation:

- Train and test datasets are predicted using tuned RF model.
- Performance metrics and Model evaluation are shown below:

Train dataset:

- Confusion matrix:

```
array([[6555,    0],
       [    0, 6555]], dtype=int64)
```

Fig. 62

- Classification report table:

	precision	recall	f1-score	support
0	1.0	1.0	1.0	6555.0
1	1.0	1.0	1.0	6555.0
accuracy	1.0	1.0	1.0	1.0
macro avg	1.0	1.0	1.0	13110.0
weighted avg	1.0	1.0	1.0	13110.0

Table. 28

- Accuracy score is 100.00%
- ROC_AUC score is 100.00%

- ROC curve:

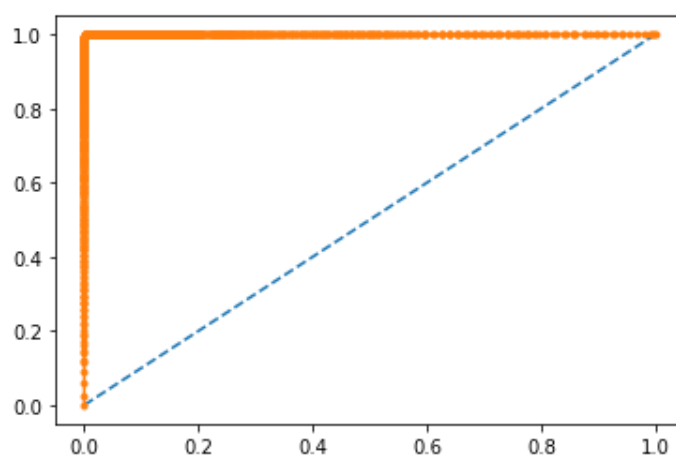


Fig. 63

Test dataset:

- Confusion matrix:

```
array([[2787,  22],
       [ 73, 496]], dtype=int64)
```

Fig. 64

- Classification report:

	precision	recall	f1-score	support
0	0.974476	0.992168	0.983242	2809.000000
1	0.957529	0.871705	0.912603	569.000000
accuracy	0.971877	0.971877	0.971877	0.971877
macro avg	0.966002	0.931936	0.947923	3378.000000
weighted avg	0.971621	0.971877	0.971344	3378.000000

Table. 29

- Accuracy score is 97.18%
- ROC_AUC score is 0.991
- ROC curve:

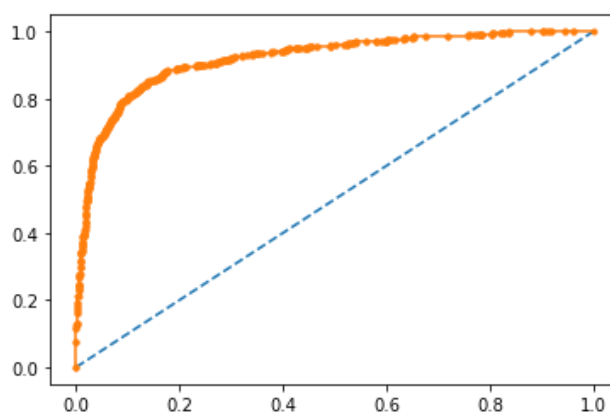


Fig. 65

Logistic Regression:

- Model tuning done by cut-off probability method.
- None of the cut-off has improved the accuracy of the model compared to the basic model.
- Maximum accuracy attained is 0.85 with 0.5 probability which is same as basic model

LDA:

- Model tuning done by cut-off probability method.
- None of the cut-off has improved the accuracy of the model compared to the basic model.
- Maximum accuracy attained is 0.86 with 0.5 probability which is same as basic model

K Nearest Neighbors:

- Let us do the grid search on the train dataset.

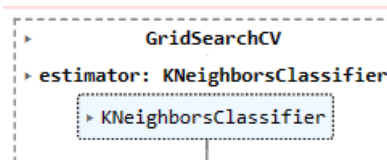


Fig. 66

- Best parameters and estimator after grid search are as below:

```
{'algorithm': 'auto', 'leaf_size': 10, 'n_neighbors': 2, 'p': 1, 'weights': 'distance'}
```

Fig. 67

Accuracy scores and validity after grid search:

- For train dataset, 100.00%
- For test dataset, 97.89%
- Accuracy score is improved for both the train and test datasets after tuning the model using grid search.

Prediction and Model evaluation:

- Train and test datasets are predicted using tuned RF model.
- Performance metrics and Model evaluation are shown below:

Train dataset:

- Confusion matrix:

```
array([[6555,    0],
       [    0, 6555]], dtype=int64)
```

Fig. 68

- Classification report table:

	precision	recall	f1-score	support
0	1.0	1.0	1.0	6555.0
1	1.0	1.0	1.0	6555.0
accuracy	1.0	1.0	1.0	1.0
macro avg	1.0	1.0	1.0	13110.0
weighted avg	1.0	1.0	1.0	13110.0

Table. 30

- Accuracy score is 100.00%
- ROC_AUC score is 100.00%
- ROC curve:

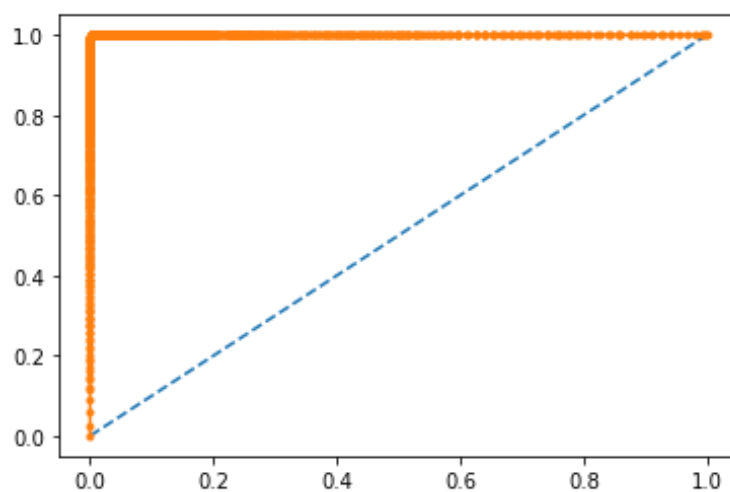


Fig. 69

Test dataset:

- Confusion matrix:

```
array([[2762,  47],
       [ 24, 545]], dtype=int64)
```

Fig. 70

- Classification report:

	precision	recall	f1-score	support
0	0.991385	0.983268	0.987310	2809.000000
1	0.920608	0.957821	0.938846	569.000000
accuracy	0.978982	0.978982	0.978982	0.978982
macro avg	0.955997	0.970544	0.963078	3378.000000
weighted avg	0.979464	0.978982	0.979147	3378.000000

Table. 31

- Accuracy score is 97.89%

- ROC_AUC score is 0.981
- ROC curve:

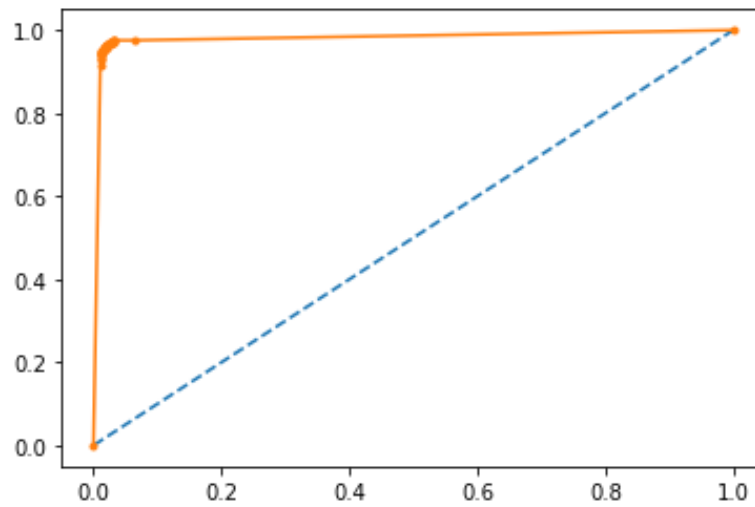


Fig. 71

➤ Ensemble Modelling

Bagging Classifier:

- Let us fit the train dataset by using Bagging Classifier model.

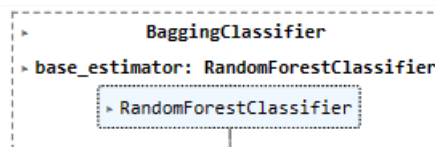


Fig. 72

- Hyperparameters used by the basic model are as shown below:

```
base_estimator=None,
n_estimators=10,
max_samples=1.0,
max_features=1.0
```

Accuracy scores:

- For train dataset, 99.83%
- For test dataset, 96.50%

Prediction and Model evaluation:

- Train and test datasets are predicted using Bagging Classifier model.
- Performance metrics and Model evaluation are shown below:

Train dataset:

- Confusion matrix:

```
array([[6549, 6],
       [ 15, 6540]], dtype=int64)
```

Fig. 73

- Classification report table:

	precision	recall	f1-score	support
0	0.997715	0.999085	0.998399	6555.000000
1	0.999083	0.997712	0.998397	6555.000000
accuracy	0.998398	0.998398	0.998398	0.998398
macro avg	0.998399	0.998398	0.998398	13110.000000
weighted avg	0.998399	0.998398	0.998398	13110.000000

Table. 32

- Accuracy score is 99.83%
- ROC_AUC score is 100.00%
- ROC curve:

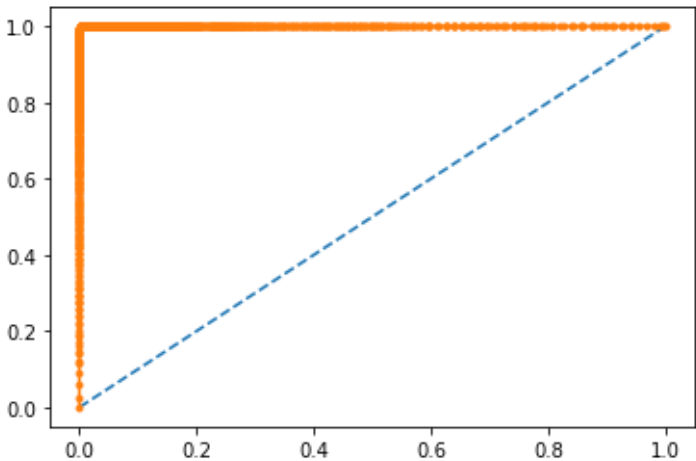


Fig. 74

Test dataset:

- Confusion matrix:

```
array([[2769, 40],
       [ 78, 491]], dtype=int64)
```

Fig. 75

- Classification report:

	precision	recall	f1-score	support
0	0.972603	0.985760	0.979137	2809.000000
1	0.924670	0.862917	0.892727	569.000000
accuracy	0.965068	0.965068	0.965068	0.965068
macro avg	0.948637	0.924339	0.935932	3378.000000
weighted avg	0.964529	0.965068	0.964582	3378.000000

Table. 33

- Accuracy score is 96.50%
- ROC_AUC score is 0.987
- ROC curve:

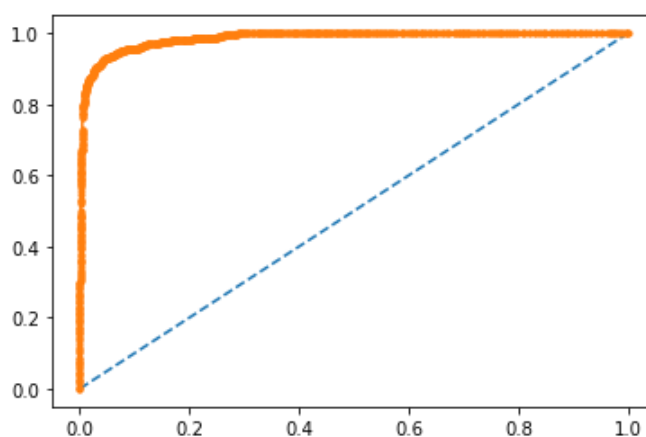


Fig. 76

Adaboost Classifier:

- Let us fit the train dataset by using Adaboost Classifier model.

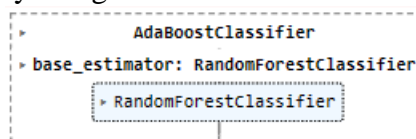


Fig. 77

- Hyperparameters used by the basic model are as shown below:

```
base_estimator=rf,
n_estimators=50,
learning_rate=1.0,
algorithm='SAMME.R',
random_state=1
```

Accuracy scores:

- For train dataset, 100.00%
- For test dataset, 97.15%

Prediction and Model evaluation:

- Train and test datasets are predicted using Adaboost Classifier model.
- Performance metrics and Model evaluation are shown below:

Train dataset:

- Confusion matrix:

```
array([[6555,  0],  
       [ 0, 6555]], dtype=int64)
```

Fig. 78

- Classification report table:

	precision	recall	f1-score	support
0	1.0	1.0	1.0	6555.0
1	1.0	1.0	1.0	6555.0
accuracy	1.0	1.0	1.0	1.0
macro avg	1.0	1.0	1.0	13110.0
weighted avg	1.0	1.0	1.0	13110.0

Table. 34

- Accuracy score is 100.00%
- ROC_AUC score is 100.00%
- ROC curve:

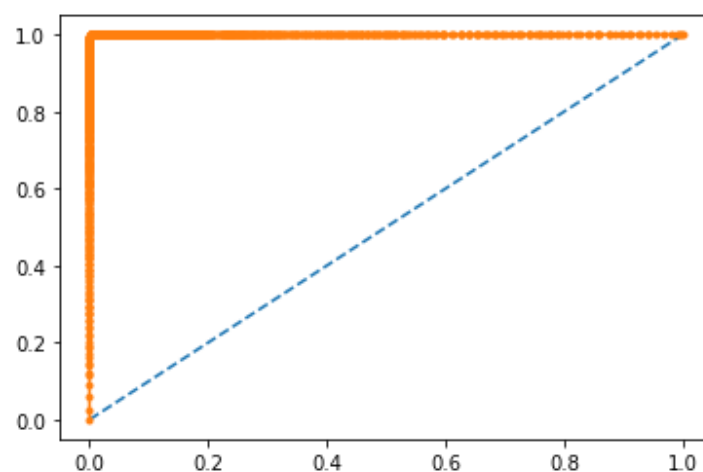


Fig. 79

Test dataset:

- Confusion matrix:

```
array([[2787, 22],
       [ 74, 495]], dtype=int64)
```

Fig. 80

- Classification report:

	precision	recall	f1-score	support
0	0.974135	0.992168	0.983069	2809.000000
1	0.957447	0.869947	0.911602	569.000000
accuracy	0.971581	0.971581	0.971581	0.971581
macro avg	0.965791	0.931058	0.947335	3378.000000
weighted avg	0.971324	0.971581	0.971031	3378.000000

Table. 35

- Accuracy score is 97.15%
- ROC_AUC score is 0.991
- ROC curve:

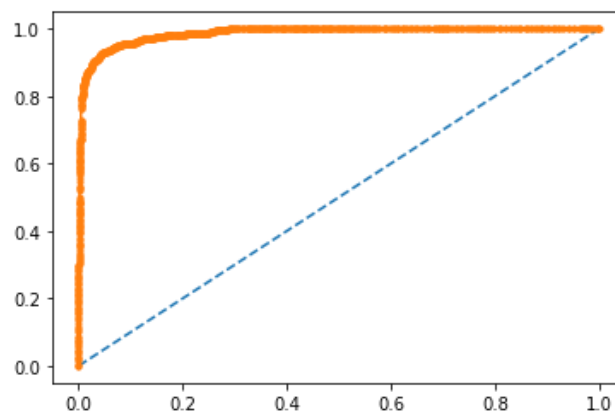


Fig. 81

Gradient Boosting Classifier:

- Let us fit the train dataset by using Bagging Classifier model.

```
GradientBoostingClassifier
GradientBoostingClassifier(random_state=1)
```

Fig. 82

- Hyperparameters used by the basic model are as shown below:

```
loss='log_loss',
learning_rate=0.1,
n_estimators=100,
subsample=1.0,
```

```
criterion='friedman_mse',
min_samples_split=2,
min_samples_leaf=1,
tol=0.0001
```

Accuracy scores:

- For train dataset, 93.61%
- For test dataset, 90.11%

Prediction and Model evaluation:

- Train and test datasets are predicted using Gradient Boosting Classifier model.
- Performance metrics and Model evaluation are shown below:

Train dataset:

- Confusion matrix:

```
array([[6183, 372],
       [ 465, 6090]], dtype=int64)
```

Fig. 83

- Classification report table:

	precision	recall	f1-score	support
0	0.930054	0.943249	0.936605	6555.000000
1	0.942433	0.929062	0.935699	6555.000000
accuracy	0.936156	0.936156	0.936156	0.936156
macro avg	0.936243	0.936156	0.936152	13110.000000
weighted avg	0.936243	0.936156	0.936152	13110.000000

Table. 36

- Accuracy score is 93.61%
- ROC_AUC score is 0.983%
- ROC curve:

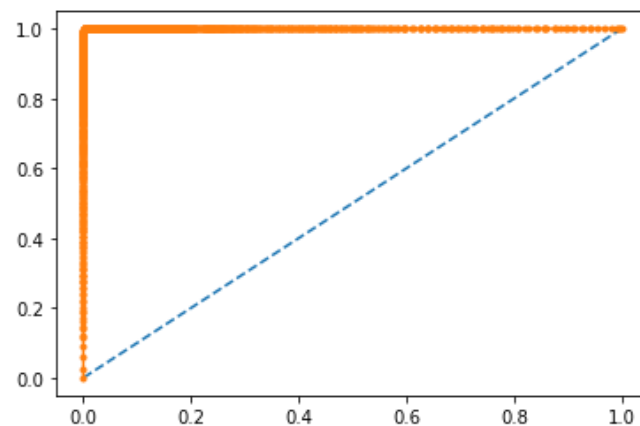


Fig. 84

Test dataset:

- Confusion matrix:

```
array([[2638, 171],  
       [ 163, 406]], dtype=int64)
```

Fig. 85

- Classification report:

	precision	recall	f1-score	support
0	0.941806	0.939124	0.940463	2809.000000
1	0.703640	0.713533	0.708551	569.000000
accuracy	0.901125	0.901125	0.901125	0.901125
macro avg	0.822723	0.826328	0.824507	3378.000000
weighted avg	0.901689	0.901125	0.901400	3378.000000

Table. 37

- Accuracy score is 90.11%
- ROC_AUC score is 0.920
- ROC curve:

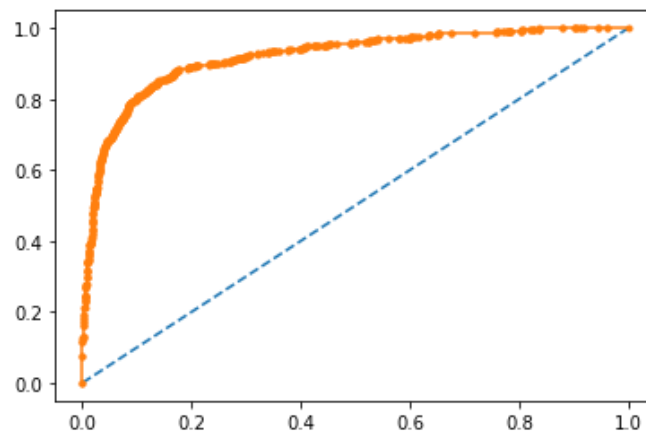


Fig. 86

5. Model validation

- **Summary and Interpretation of all the performance metrics of the classification models:**

Let us summarize the performance metrics of all the classification models performed:

	Accuracy	AUC	Precision	Recall	F1 Score
CART_Train	1.00	1.00	1.00	1.00	1.00
CART_Test	0.93	0.89	0.75	0.84	0.79
RF_Train	1.00	1.00	1.00	1.00	1.00
RF_Test	0.97	0.99	0.95	0.88	0.91
LOR_Train	0.85	0.92	0.86	0.86	0.86
LOR_Test	0.81	0.85	0.50	0.70	0.58
LDA_Train	0.86	0.93	0.86	0.86	0.86
LDA_Test	0.83	0.85	0.51	0.69	0.58
KNN_Train	1.00	1.00	1.00	1.00	1.00
KNN_Test	0.97	0.96	0.89	0.90	0.90

Table. 38

- RF, KNN are performing better in terms of all performance metrics. But there is more than more than 10% deviation between in some metrics for train and test sets. For e.g., Recall value has 12% deviation between train and test sets
- CART model performing very well on train set, but failing for the test set. We can try for tuning this CART model for better performance on the test set also.
- Logistic regression model is not performing well for this dataset. Logistic regression model assumes linear relationship between dependent and independent variables. But there is no such relationship in our dataset.
- LDA also not performing well for this dataset. LDA assumes all the variables to be normally distributed, but some of the variables are skewed in our dataset which is making LDA model poor performance.
- Since we have issue with linear relationship and distribution of variables. Cluster and distance-based models such Decision trees and KNN are giving good results here.
- Since we have plenty of categorical variables in our dataset, RF performing better over Logistic Regressions model and LDA.
- Clearly, our dataset is falling under non-parametric methods. That's why KNN, Decision trees are giving good results.

➤ Summary and Interpretation of the performance metrics of the ensemble models

Let us summarize the performance metrics of all the classification models performed:

	Accuracy	AUC	Precision	Recall	F1 Score
Bagging_Train	1.00	1.00	1.00	1.00	1.00
Bagging_Test	0.97	0.99	0.92	0.86	0.89
Adaboost_Train	1.00	1.00	1.00	1.00	1.00
Adaboost_Test	0.97	0.99	0.96	0.87	0.91
Gboost_Train	0.94	0.98	0.94	0.93	0.94
Gboost_Test	0.90	0.92	0.70	0.71	0.71

Table. 39

- Bagging and Adaboost ensemble models are giving good results. But, recall value is poor, it has more than 10% deviation between train and test sets.

- Since bagging and boosting models are also sort of non-parametric models, these are performing better for our dataset.

➤ **Summary of the tuned classification models in terms of misclassifications:**

Misclassifications/Model	FP	FN
CART	155	93
CARTg	176	139
RF	26	69
RFg	22	73
Logistic	395	171
LDA	383	177
KNN	63	55
KNNg	47	24
Bagging	40	78
Adaboost	22	74
Gradientboost	171	163

Table. 40

➤ **Summary and Interpretation of the tuned classification models:**

Let us summarize the performance metrics of all the classification models performed:

	Accuracy	AUC	Precision	Recall	F1 Score
CARTg_Train	1.00	1.00	1.00	1.00	1.00
CARTg_Test	0.93	0.89	0.71	0.76	0.73
RFg_Train	1.00	1.00	1.00	1.00	1.00
RFg_Test	0.97	0.99	0.96	0.87	0.91
KNNg_Train	1.00	1.00	1.00	1.00	1.00
KNNg_Test	0.98	0.98	0.92	0.96	0.94

Table. 41

Optimum model selection and Overall Interpretation:

- KNN model is optimum model for our dataset.
- After performing GridSearchCV on CART, RF, KNN models, KNN has given significant improvement in all the performance metrics for test dataset. All the metrics are within less than 10% deviation.

Interpretations from optimum model:

- Accuracy for the test set is 98%: Only 2% of the customers churn prediction is wrong.
- Recall vs Precision: Recall is more important than Precision since FNs prediction is more important and valuable than FPs for the company. If FNs are low, more the recall and lesser the risk for the business and the profits.
FN- Actually churn, but predicting as non-churn
FP- Actually non-churn, but predicting as churn

- More number of FN's are risk for the business, So KNN is the best model for this dataset.

6. Final interpretation / recommendation

Business Insights:

- From feature importance, we can see tenure, cashback and complain_ly are the top features. This is indicating customer service in different ways to customers, tenure of the customers, cashback for the purchases, complain cell playing key role in the e-Commerce industry.
- Mean revenue per account per month is less than cashback amount which is why revenue growth is moderate from year to year.
- Primary account holder and other customers are not completely satisfied towards the customer service.
- Customers under high revenue generation are less.
- Majority of the customers confined to Tier-01 cities.

Business recommendations:

- Customers should be attracted for having long term relationship with the company as average tenure is ~11 months only.
- Customer service team need to be trained well so that customer disappointment by the service will be lesser.
- Tenure based customers segmentation should be done.
- Advertisements and other publicity acts should be done to promote the business in Tier-02, Tier-03 cities.
- Attractive coupons number should be increased per account based on their purchase from the company.
- Products should be displayed on customers browsing history and frequently bought products recommendations also.
- Product recommendation engines can be used to perform email campaigns.
- New items should be highlighted to the customers in effective way.
- Product bundles need to created and should be shown to the customers.
- High rated items should be showcased.

THE END