CAPSTONE PROJECT - CUSTOMER CHURN

Final Report - Mrudhulaa PV

19/03/2023 - PGP DSBA Online -March' 22

Content of Report

List of figures

List of tables

1. Introduction – Business Problem Definition

Problem Statement

The current market is presenting significant challenges for an E-commerce company, as competition is intense and it has become increasingly difficult to retain existing customers. To address this issue, the DTH company is seeking to develop a model that can accurately predict customer churn and provide segmented offers to potential churners. Account churn is a major concern for this company since one account can represent multiple customers, meaning that losing just one account could result in the loss of several customers.

To tackle this challenge, we have been tasked with developing a unique churn prediction model for the company and providing insightful business recommendations for the campaign. It is important that the model and campaign are effective and efficient, as the offers suggested must be sharp and appealing to customers in order to retain them. Moreover, the suggested offers must benefit both the company and the customers to avoid any adverse effects on revenue.

Need of the project/study

The cost of acquiring a new customer is the biggest expense for DTH providers. It takes years for the cost of acquisition to be recovered, and for the account to become profitable. Therefore, customer churn has a direct impact on the profitability of DTH operators. These providers face constant pressure to increase their customer base, as their profitability is tied to the number of customers they have. To protect their current customer base, it is crucial to not only increase it but also prevent customer churn.

Acquiring a new customer can cost up to five times more than retaining an existing one. By increasing customer retention by just 5%, profits can rise by anywhere from 25-95%. As customer churn directly impacts both the top-line and bottom-line revenue of the business, protecting the existing customer base is imperative. However, providing offers to all customers to retain them could significantly impact profitability. Therefore, it is essential to focus only on a select set of customers who are at a higher risk of churning, and offer them incentives to keep them from leaving.

Objective

The aim is to analyse the factors contributing to customer churn and to come up with an optimum model which will efficiently predict the churning of customers and provide segmented offers as a apart of retention strategy.

Scope

- The most effective model for predicting churn
- Significant insights and recommendations from the model and EDA

Constraints

- Prior retention initiatives may have included cashbacks or discounts, but this is unknown
- Unknown campaign budget

Data Report

Dataset of problem: Customer churn analysis

Data Dictionary:

- AccountID -- account unique identifier
- Churn -- account churn flag (Target Variable)
- Tenure -- Tenure of account
- City_Tier -- Tier of primary customer's city
- CC_Contacted_L12m -- How many times all the customers of the account has contacted customer care in last 12months
- Payment -- Preferred Payment mode of the customers in the account
- Gender -- Gender of the primary customer of the account
- Service_Score -- Satisfaction score given by customers of the account on service provided by company
- Account_user_count -- Number of customers tagged with this account
- account_segment -- Account segmentation on the basis of spend
- CC_Agent_Score -- Satisfaction score given by customers of the account on customer care service provided by company
- Marital_Status -- Marital status of the primary customer of the account
- rev_per_month -- Monthly average revenue generated by account in last 12 months
- Complain_l12m -- Any complaints has been raised by account in last 12 months
- rev_growth_yoy -- revenue growth percentage of the account (last 12 months vs last 24 to 13 month)
- coupon_used_l12m -- How many times customers have used coupons to do the payment in last 12 months
- Day_Since_CC_connect -- Number of days since no customers in the account has contacted the customer care
- cashback_l12m -- Monthly average cashback generated by account in last 12 months
- Login_device -- Preferred login device of the customers in the account

Data Ingestion:

Loaded the required packages, set the work directory and load the datafile.
Data set has 11,260 number of observations and 19 variables (18 independent and 1 dependent or target variable).

	0	1	2	3	4
AccountID	20000	20001	20002	20003	20004
Churn	1	1	1	1	1
Tenure	4	0	0	0	0
City_Tier	3.0	1.0	1.0	3.0	1.0
CC_Contacted_LY	6.0	8.0	30.0	15.0	12.0
Payment	Debit Card	UPI	Debit Card	Debit Card	Credit Card
Gender	Female	Male	Male	Male	Male
Service_Score	3.0	3.0	2.0	2.0	2.0
Account_user_count	3	4	4	4	3
account_segment	Super	Regular Plus	Regular Plus	Super	Regular Plus
CC_Agent_Score	2.0	3.0	3.0	5.0	5.0
Marital_Status	Single	Single	Single	Single	Single
rev_per_month	9	7	6	8	3
Complain_ly	1.0	1.0	1.0	0.0	0.0
rev_growth_yoy	11	15	14	23	11
coupon_used_for_payment	1	0	0	0	1
Day_Since_CC_connect	5	0	3	3	3
cashback	159.93	120.9	NaN	134.07	129.6
Login_device	Mobile	Mobile	Mobile	Mobile	Mobile

Table 1: Dataset sample

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

- For a random sample of 11,260 distinct account IDs, information about gender and marital status has been gathered.
- We can infer that the data has been collected over the past 12 months by looking at the variables "CC Contacted L12m," "rev per month," "Complain L12m," "rev growth yoy," "coupon Used L12m," "Day Since CC Connect," and "cashback L12m."
- 19 variables make up the data: 18 independent variables and the target variable, which indicates whether or not a customer churned.
- The information consists of the services that clients use, their preferred method of payment, and also their basic personal information.

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

- Data has 11,260 rows and 19 variables.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11260 entries, 0 to 11259
Data columns (total 19 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   account_id              11260 non-null  int64
 1   Churn                   11260 non-null  int64
 2   account_tenure          11158 non-null  object
 3   city_tier               11148 non-null  float64
 4   cust_care_contacts_12m  11158 non-null  float64
 5   payment_method          11151 non-null  object
 6   gender                  11152 non-null  object
 7   service_score           11162 non-null  float64
 8   customers_per_account   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  revenue_per_month       11158 non-null  object
 13  account_complaints_12m  10903 non-null  float64
 14  rev_growth_yoy          11260 non-null  object
 15  coupons_used            11260 non-null  object
 16  days_since_cc_contact   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 2: Dataset Information

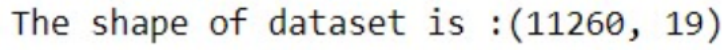


Fig 1: Shape of dataset

- Describing data: - This shows description of variation in various statistical measurements across variables which denotes that each variable is unique and different.

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
account_id	11260.0	NaN	NaN	NaN	25629.5	3250.62635	20000.0	22814.75	25629.5	28444.25	31259.0
Churn	11260.0	NaN	NaN	NaN	0.168384	0.374223	0.0	0.0	0.0	0.0	1.0
account_tenure	11158.0	38.0	1.0	1351.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
city_tier	11148.0	NaN	NaN	NaN	1.653929	0.915015	1.0	1.0	1.0	3.0	3.0
cust_care_contacts_12m	11158.0	NaN	NaN	NaN	17.867091	8.853269	4.0	11.0	16.0	23.0	132.0
payment_method	11151	5	Debit Card	4587	NaN	NaN	NaN	NaN	NaN	NaN	NaN
gender	11152	4	Male	6328	NaN	NaN	NaN	NaN	NaN	NaN	NaN
service_score	11162.0	NaN	NaN	NaN	2.902526	0.725584	0.0	2.0	3.0	3.0	5.0
customers_per_account	11148.0	7.0	4.0	4569.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
account_segment	11163	7	Super	4062	NaN	NaN	NaN	NaN	NaN	NaN	NaN
cc_agent_score	11144.0	NaN	NaN	NaN	3.066493	1.379772	1.0	2.0	3.0	4.0	5.0
marital_Status	11048	3	Married	5860	NaN	NaN	NaN	NaN	NaN	NaN	NaN
revenue_per_month	11158.0	59.0	3.0	1746.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
account_complaints_12m	10903.0	NaN	NaN	NaN	0.285334	0.451594	0.0	0.0	0.0	1.0	1.0
rev_growth_yoy	11260.0	20.0	14.0	1524.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
coupons_used	11260.0	20.0	1.0	4373.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
days_since_cc_contact	10903.0	24.0	3.0	1816.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
cashback	10789.0	5693.0	155.62	10.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
login_device	11039	3	Mobile	7482	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Table 3: Describing Dataset

- Except variables “AccountID”, “Churn”, “rev_growth_yoy” and “coupon_used_for_payment” all other variables have null values present.

```
account_id                0
Churn                     0
account_tenure          102
city_tier               112
cust_care_contacts_12m  102
payment_method          109
gender                  108
service_score            98
customers_per_account   112
account_segment          97
cc_agent_score          116
marital_Status          212
revenue_per_month       102
account_complaints_12m  357
rev_growth_yoy            0
coupons_used              0
days_since_cc_contact   357
cashback                471
login_device            221
dtype: int64
```

Table 4: Showing Null Values in Dataset

- Data has “NIL” duplicate observations.
- With above understanding of data renaming of any of the variable is not required.
- We can move towards the EDA part where in we will understand the data little better along with treating bad data, null values and outliers.

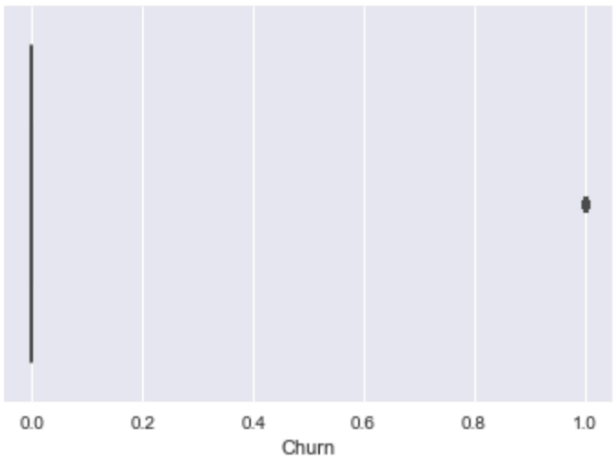
2. Exploratory Data Analysis

Univariate analysis:

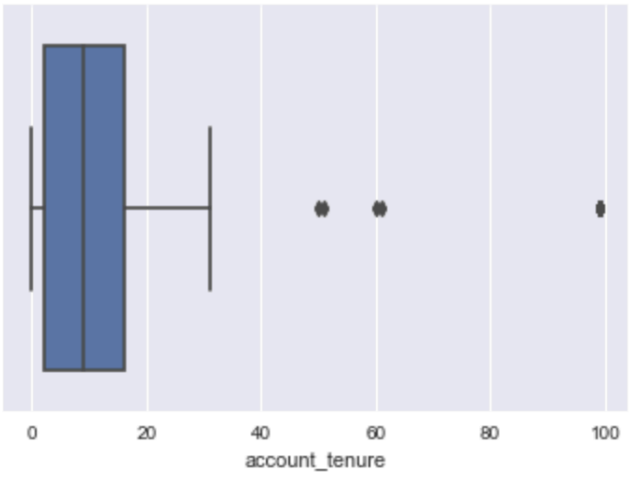
For the objective of monitoring the distribution and spread for every continuous property as well as the distribution of data in categories for categorical ones, univariate analysis is used. For continuous variables, it has been done by observing:

• Box plots and histograms
• For categorical variables, count plots

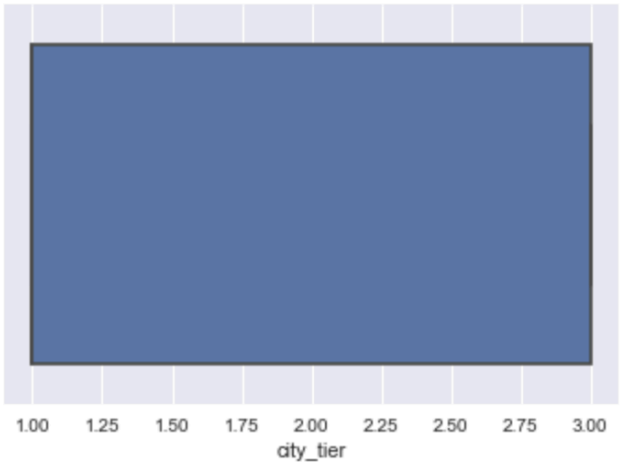
BoxPlot of Churn



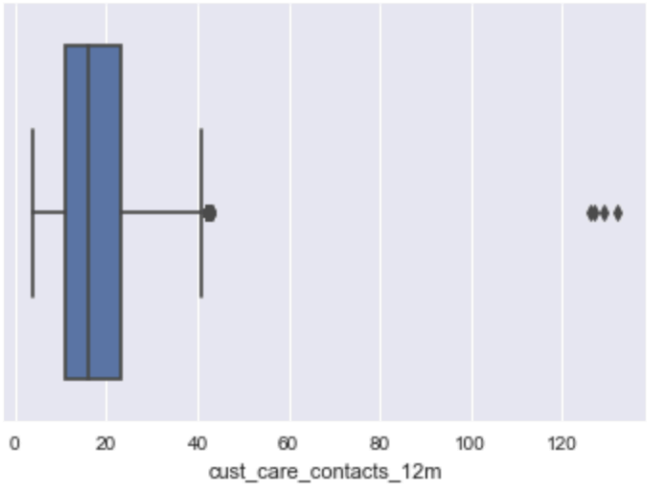
BoxPlot of account_tenure



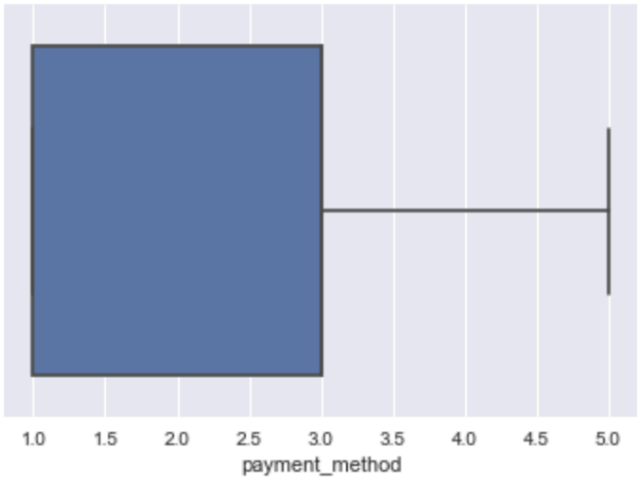
BoxPlot of city_tier



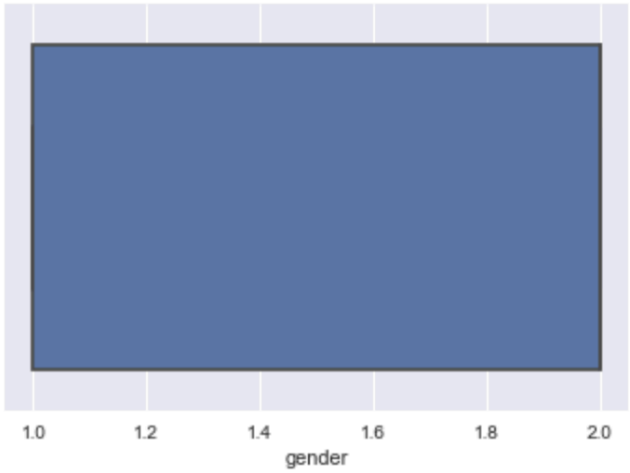
BoxPlot of cust_care_contacts_12m



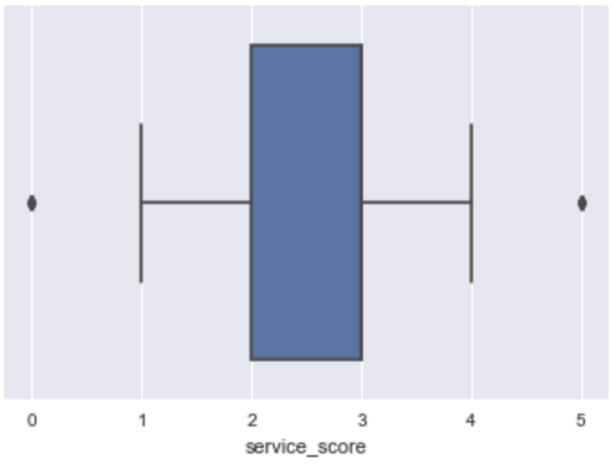
BoxPlot of payment_method



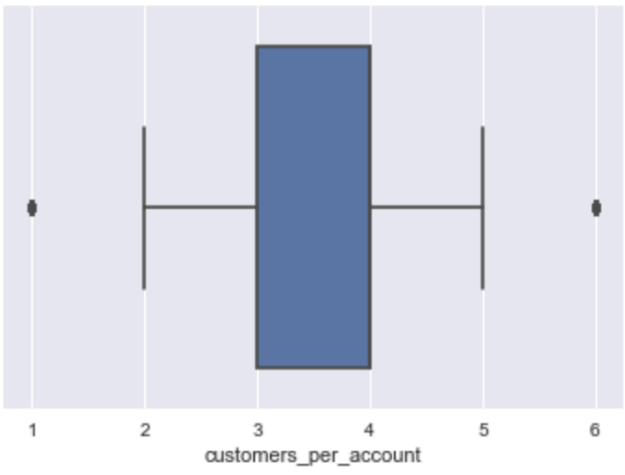
BoxPlot of gender



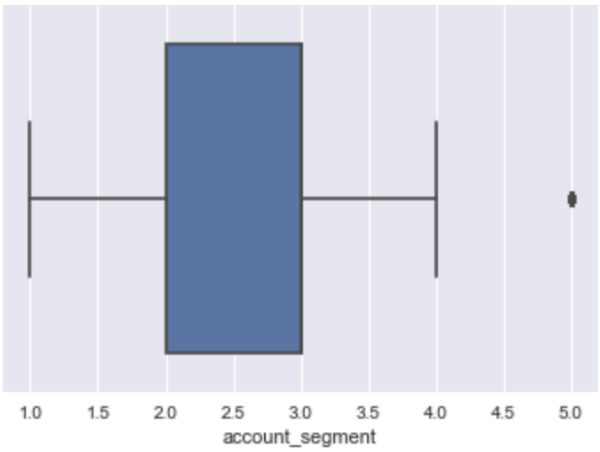
BoxPlot of service_score



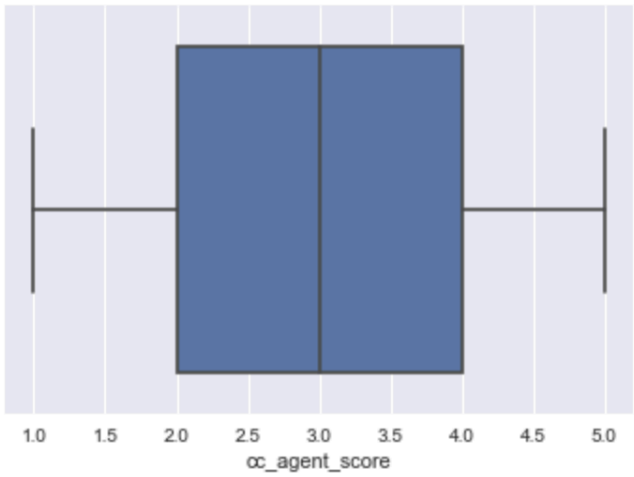
BoxPlot of customers_per_account



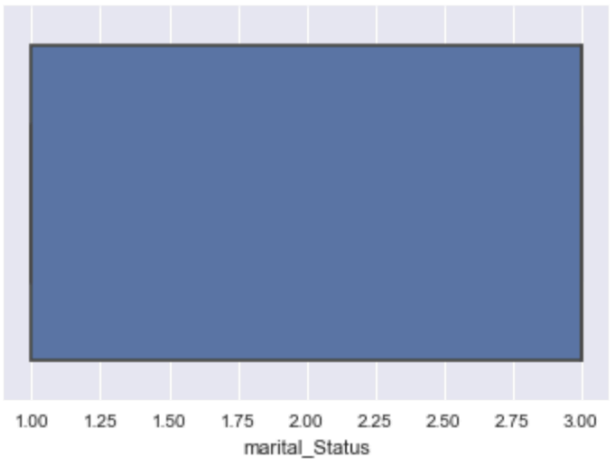
BoxPlot of account_segment



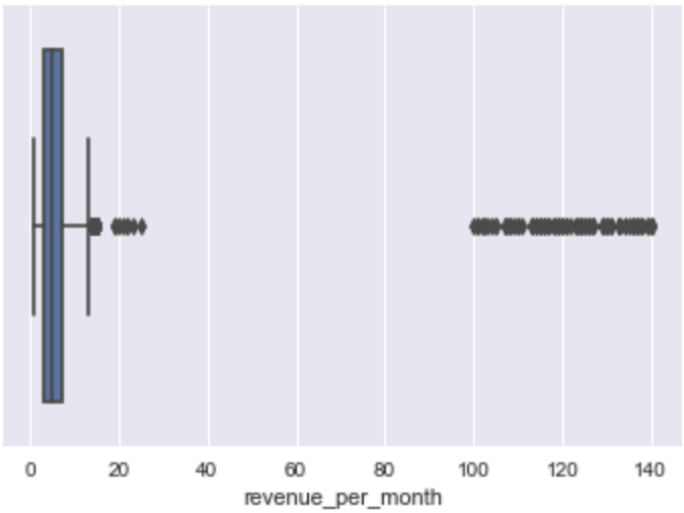
BoxPlot of cc_agent_score



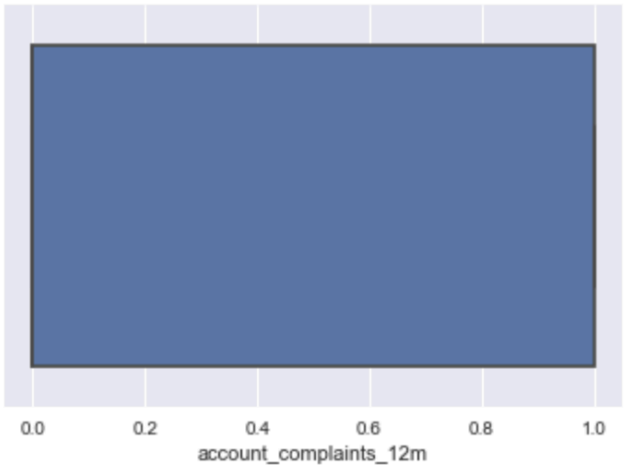
BoxPlot of marital_Status



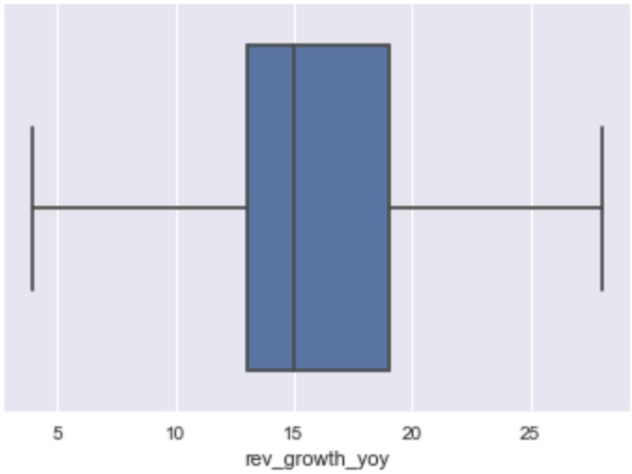
BoxPlot of revenue_per_month



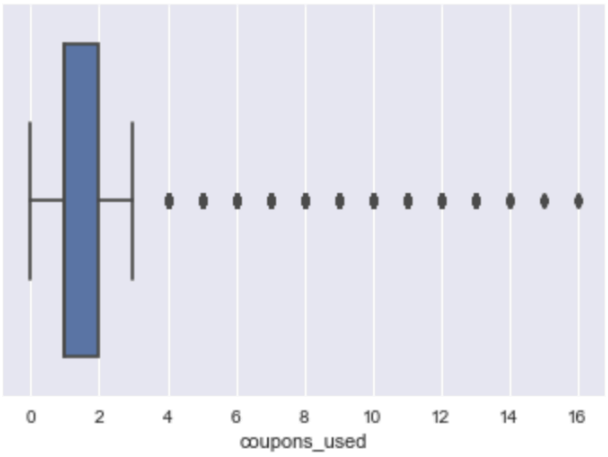
BoxPlot of account_complaints_12m



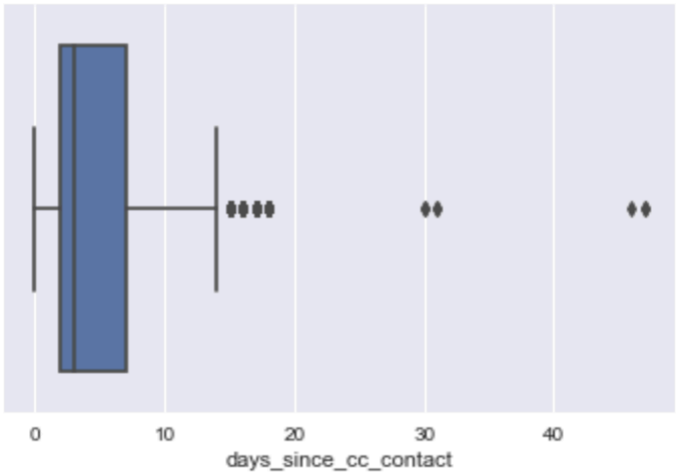
BoxPlot of rev_growth_yoy



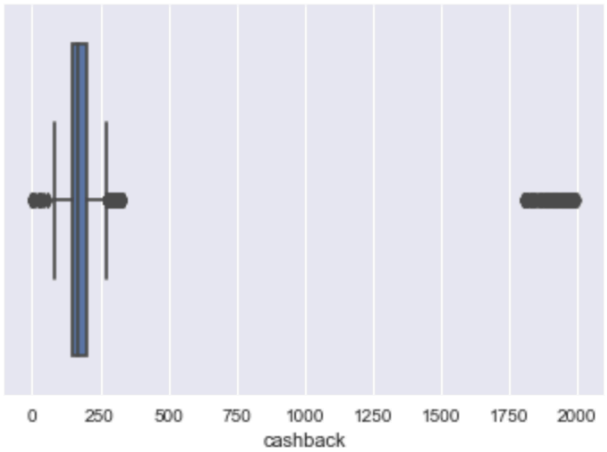
BoxPlot of coupons_used



BoxPlot of days_since_cc_contact



BoxPlot of cashback



BoxPlot of login_device

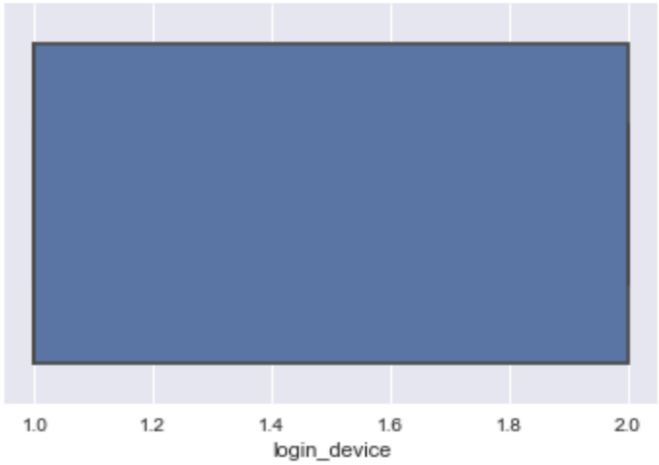
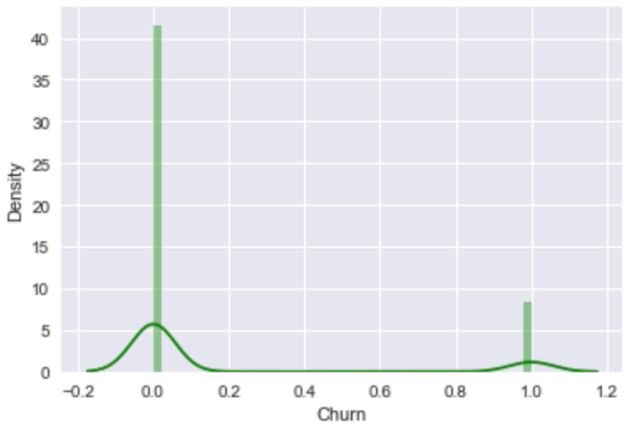
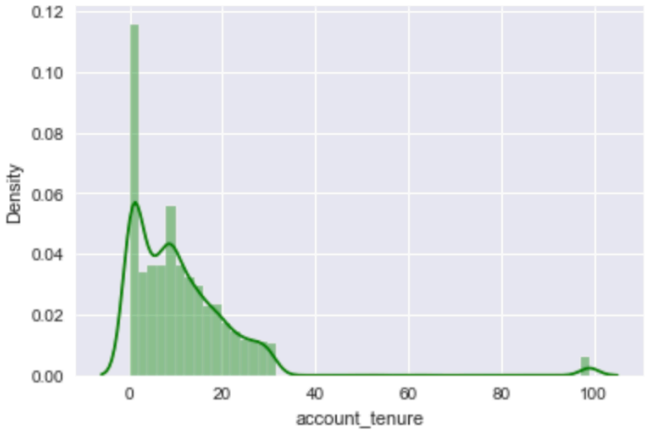


Fig 2: Box plots for numeric variables

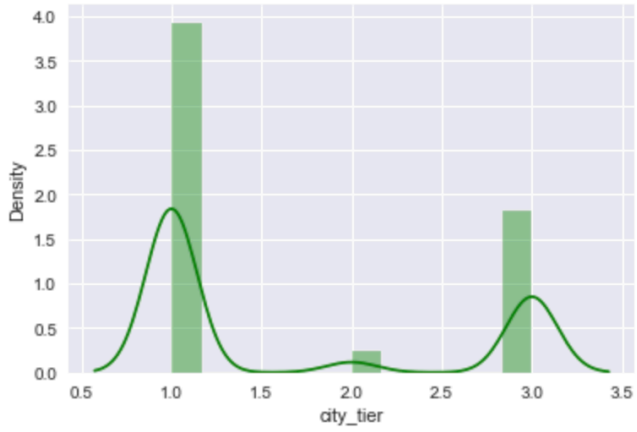
Distribution of Churn



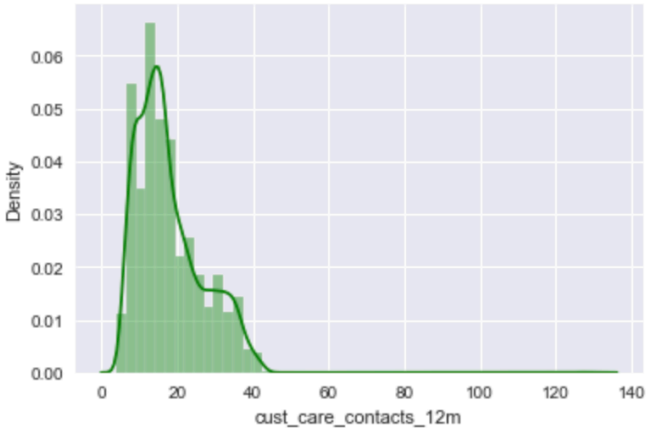
Distribution of account_tenure



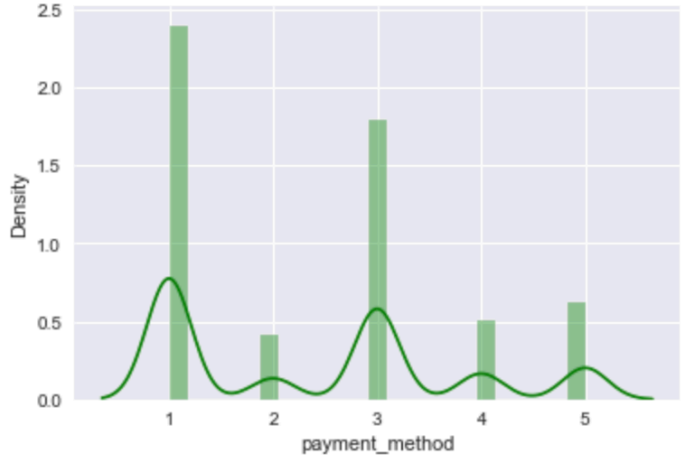
Distribution of city_tier



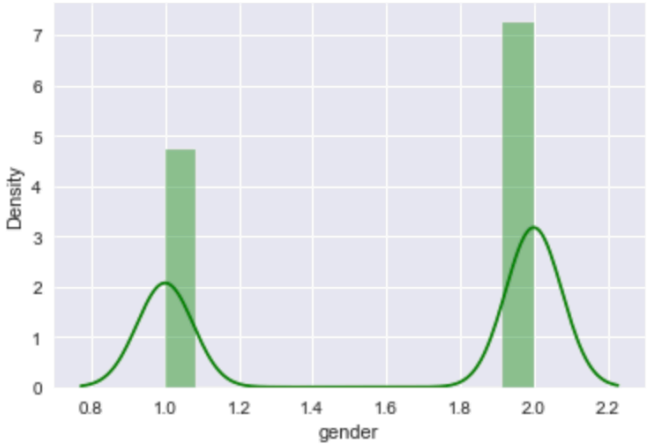
Distribution of cust_care_contacts_12m



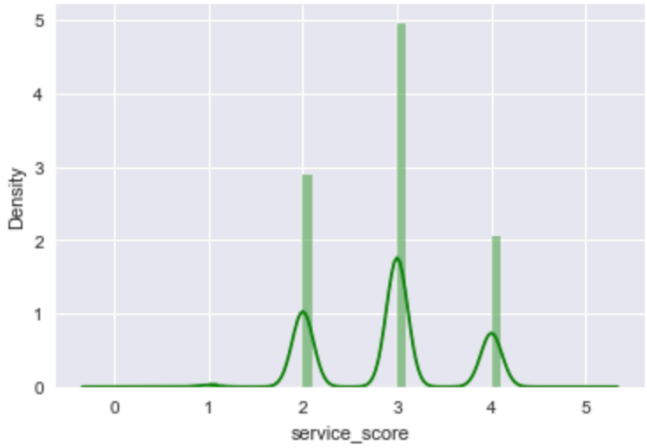
Distribution of payment_method



Distribution of gender

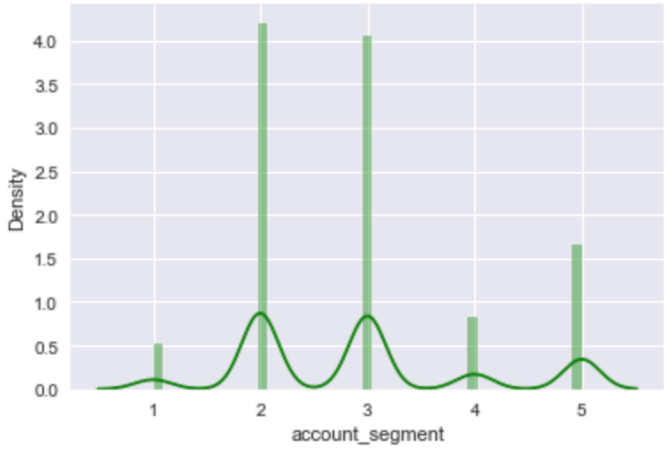


Distribution of service_score

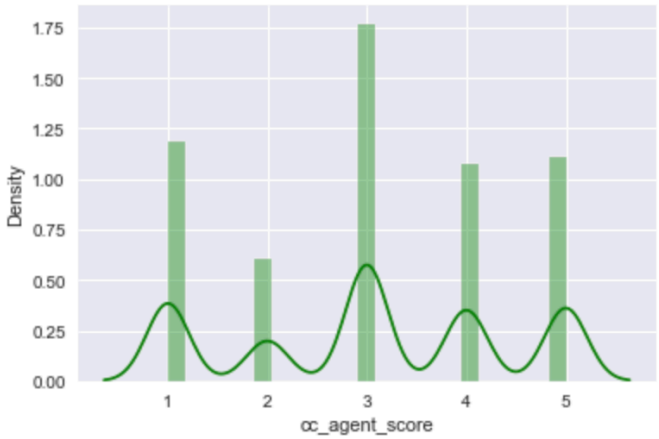


Distribution of customers_per_account

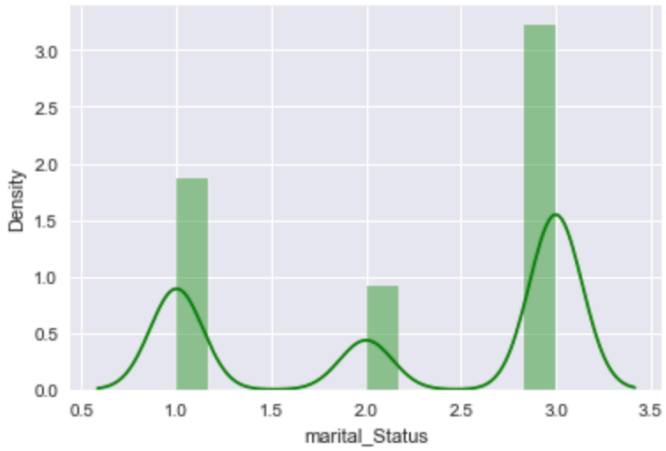
Distribution of account_segment



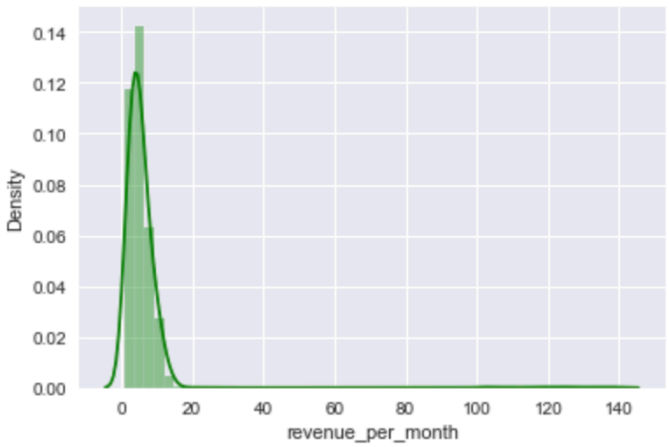
Distribution of cc_agent_score



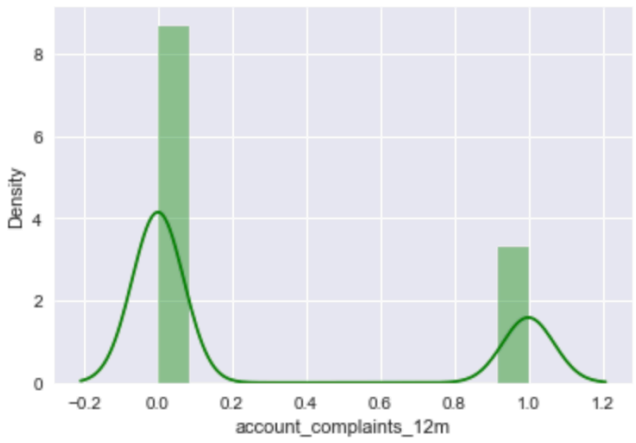
Distribution of marital_Status



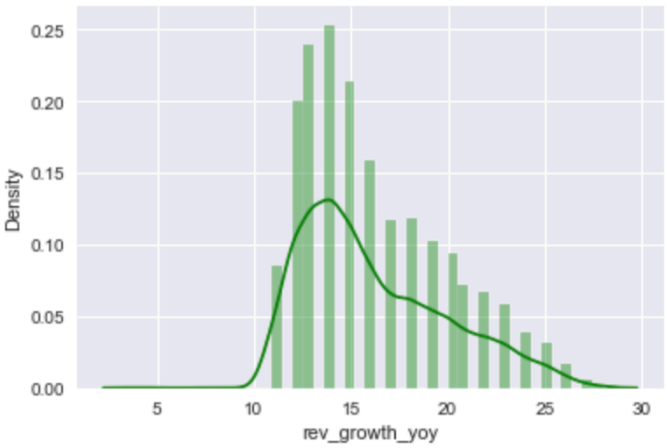
Distribution of revenue_per_month



Distribution of account_complaints_12m



Distribution of rev_growth_yoy



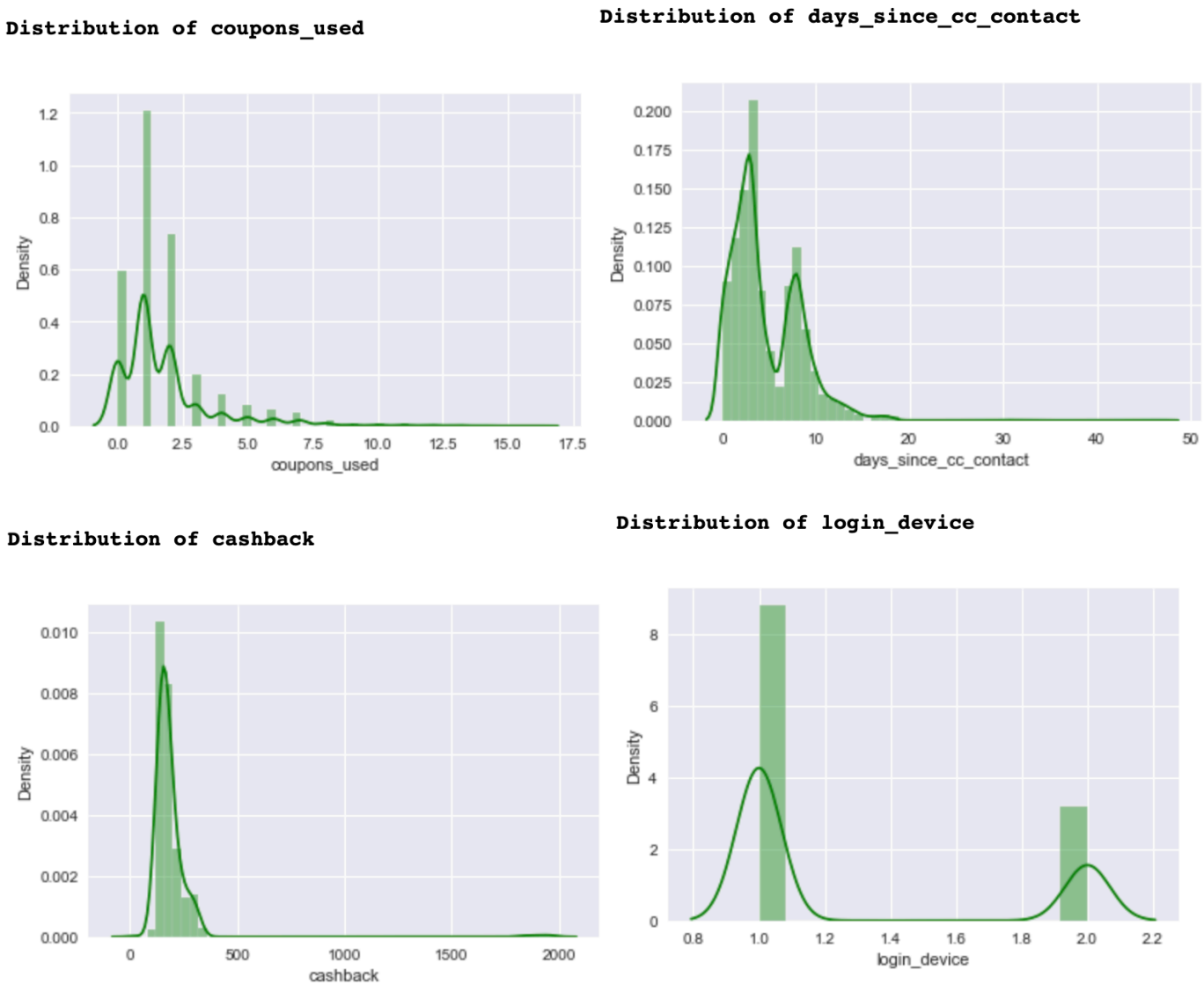


Fig 3: Histograms for numeric variables

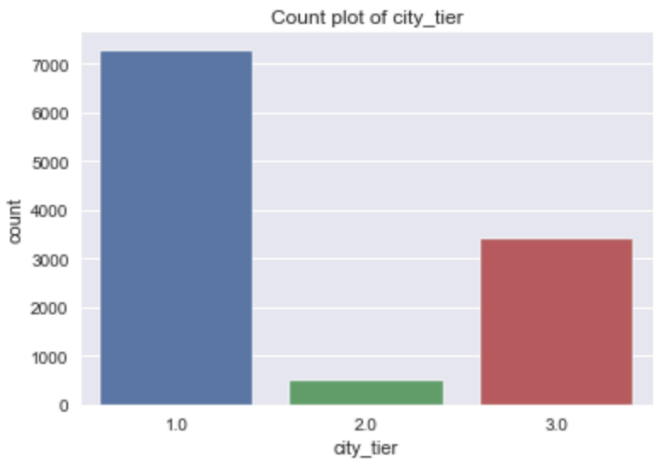
	Skewness
Churn	1.77
account_tenure	3.94
city_tier	0.75
cust_care_contacts_12m	1.44
payment_method	0.46
gender	-0.43
service_score	-0.00
customers_per_account	-0.44
account_segment	0.65
cc_agent_score	-0.13
marital_Status	-0.46
revenue_per_month	9.35
account_complaints_12m	1.00
rev_growth_yoy	0.75
coupons_used	2.55
days_since_cc_contact	1.33
cashback	8.88
login_device	1.06

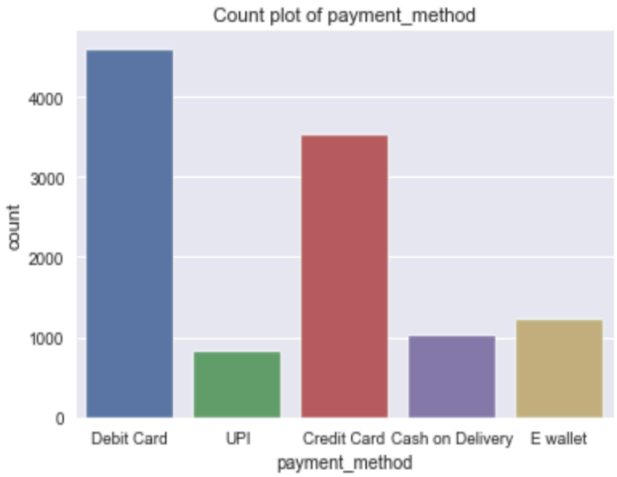
Table 5: Skewness of numeric variables

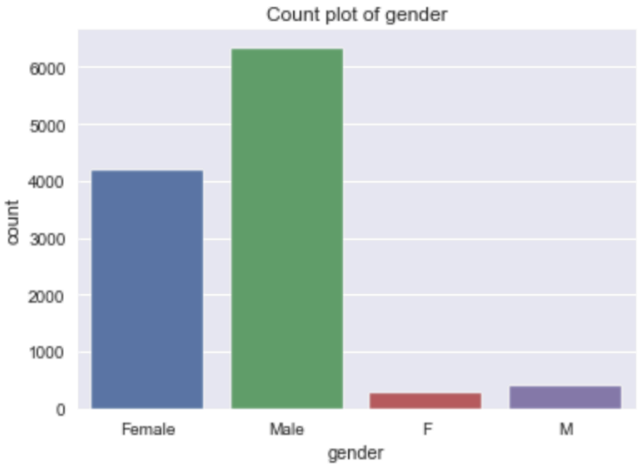
Observation:

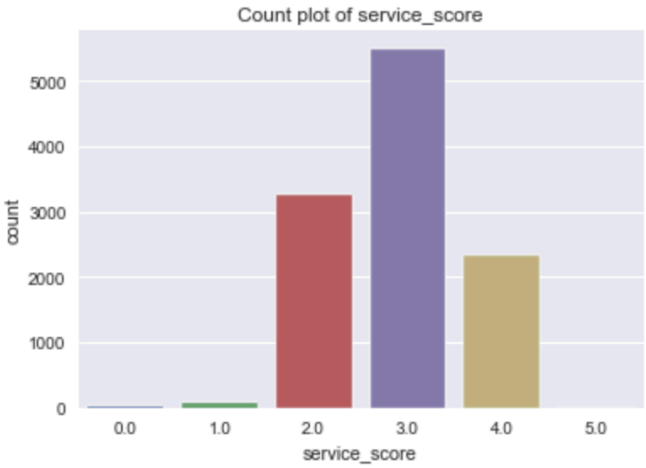
- The variable "churn" is highly skewed, with more values of 0 than 1.
- The variable "account_id" has a symmetrical distribution with values heavily concentrated around the mean, and no outliers.
- The variable "cust_care_contacts_12m" is moderately skewed, with a few high values that increase the mean, and some values far from the average.
- The variable "city_tier" is skewed to the right, with a few higher values that increase the mean, and some values deviating significantly from the average.
- The variable "service_score" has the majority of values centered around the median of 3, with relatively low deviation from the mean.
- The variable "cc_agent_score" also has the majority of values centered around the median of 3, but with a larger spread than the median, and some values deviating significantly from the average.
- The variable "account_complaints_12m" has the majority of values centered around 0, with relatively large deviation from the mean. The median is also equal to 0, indicating no skewness.

Categorical features – Count plot

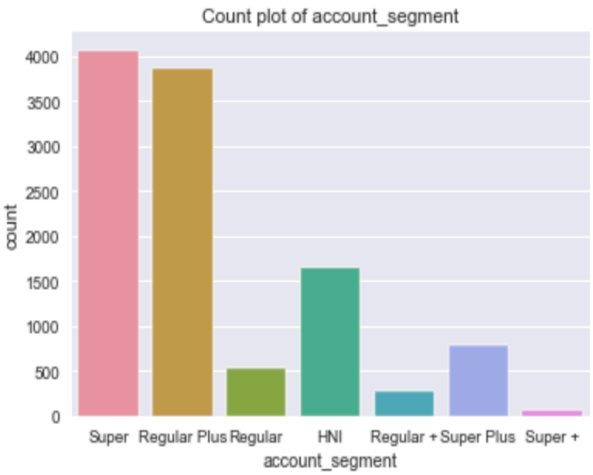

Count plot of city_tier
count
city_tier
1.0
2.0
3.0


Count plot of payment_method
count
payment_method
Debit Card
UPI
Credit Card
Cash on Delivery
E wallet


Count plot of gender
count
gender
Female
Male
F
M


Count plot of service_score
count
service_score
0.0
1.0
2.0
3.0
4.0
5.0


Count plot of customers_per_account
count
customers_per_account
3
4
5
2
@
1
6


Count plot of account_segment
count
account_segment
Super
Regular Plus
Regular
HNI
Regular +
Super Plus
Super +

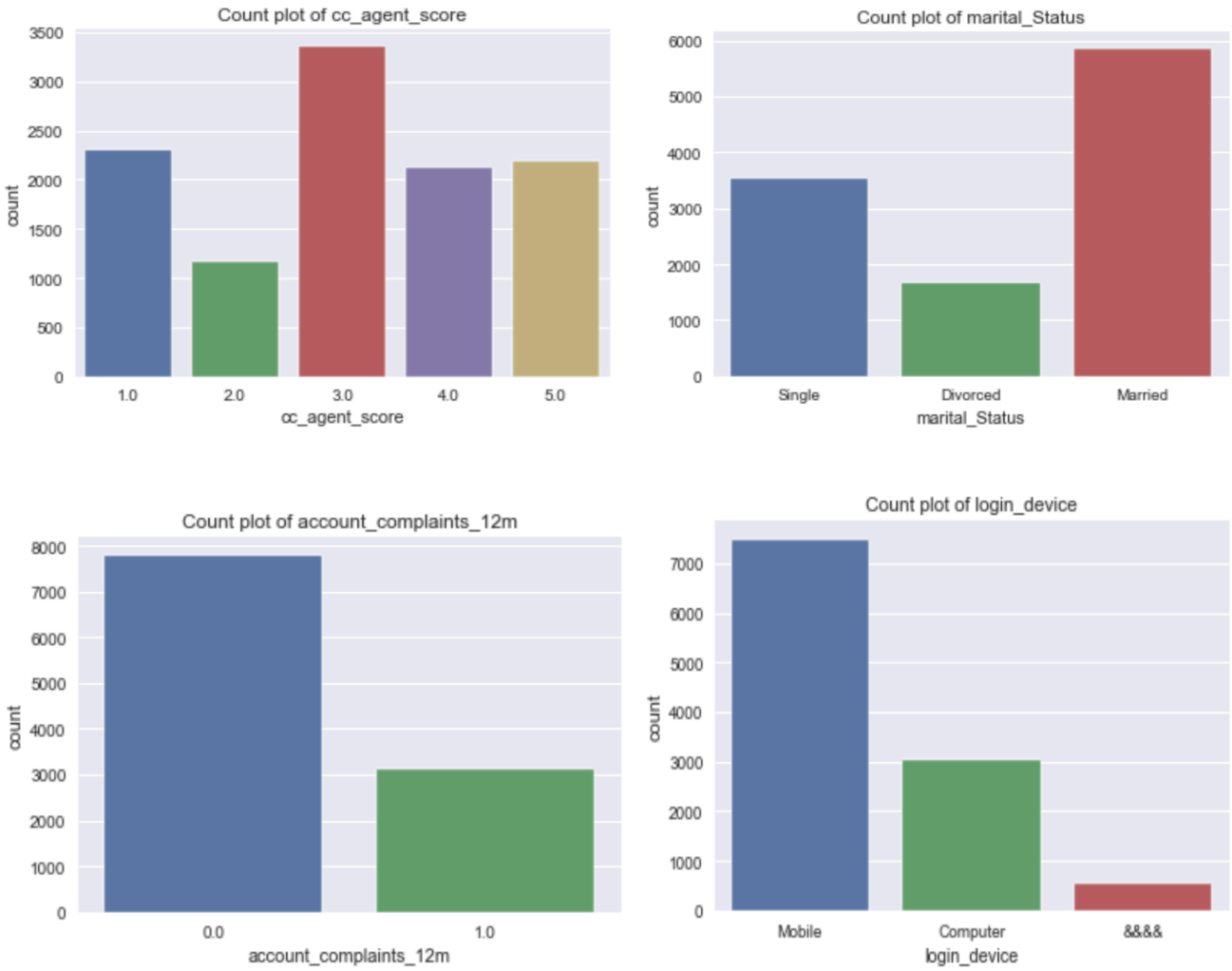


Fig 4: Count plots for categorical variables

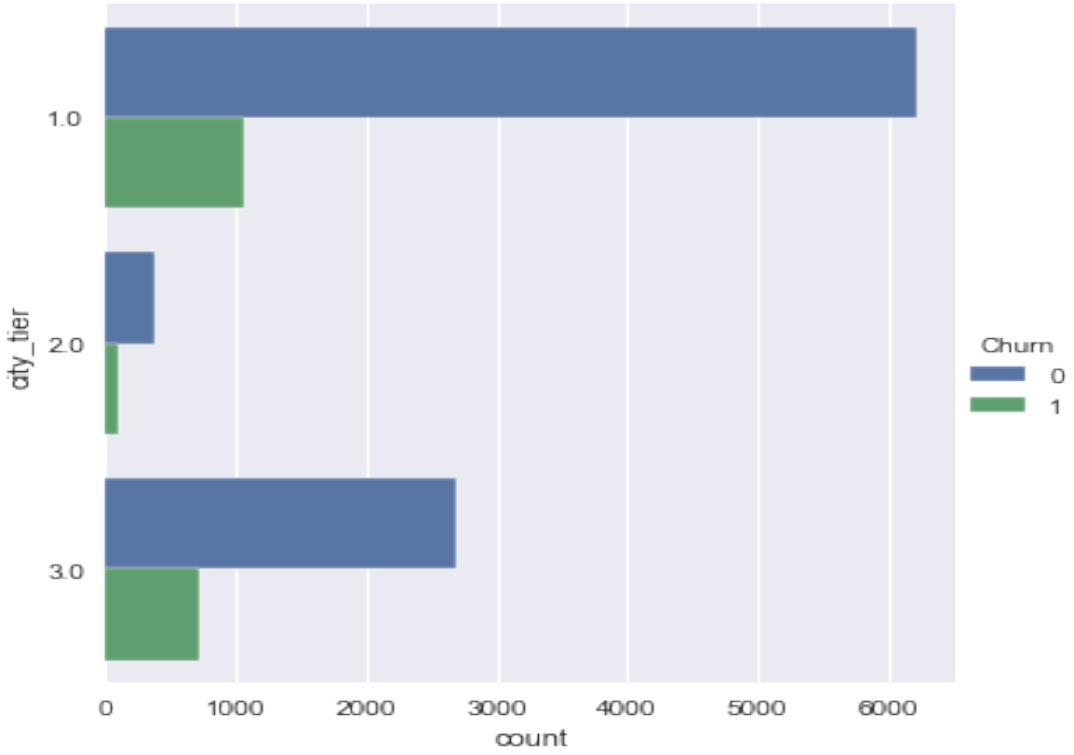
Observation:

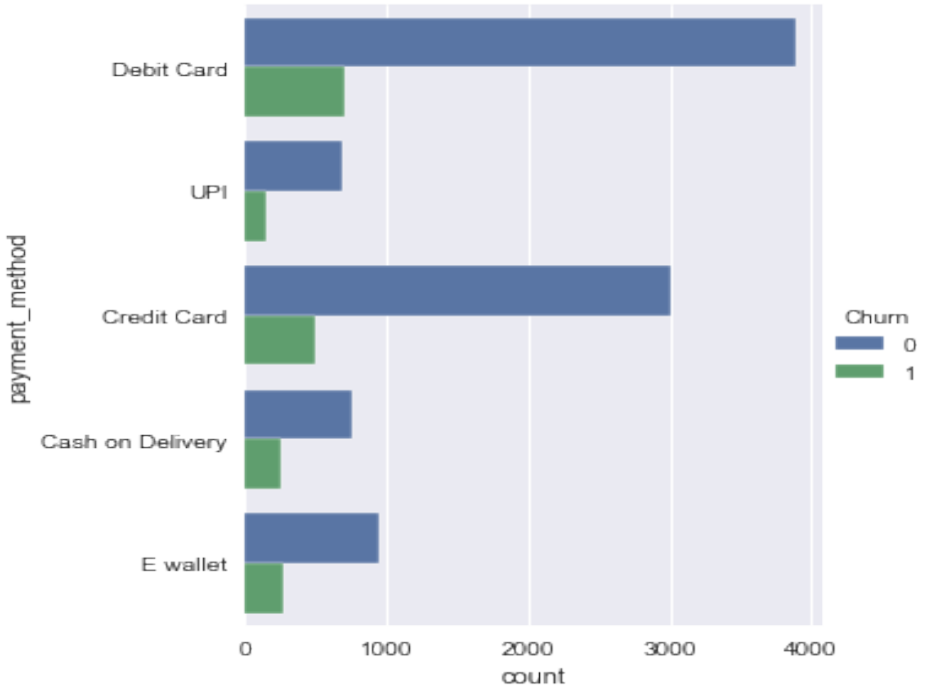
- The majority of clients come from city tyre type "1," which reflects the dense population in this city type.
- The majority of clients prefer using a debit or credit card to make payments.
- In comparison to female clients, there are more male customers.
- The typical service rating supplied by a consumer for the service received is "3," which indicates areas for improvement.
- Majority of the clients availing services are "Married".
- Most of the customers are into “Super” segment and least number of customers are into “Super+” segment.
- The majority of customers prefer "Mobile" as their method of service delivery.

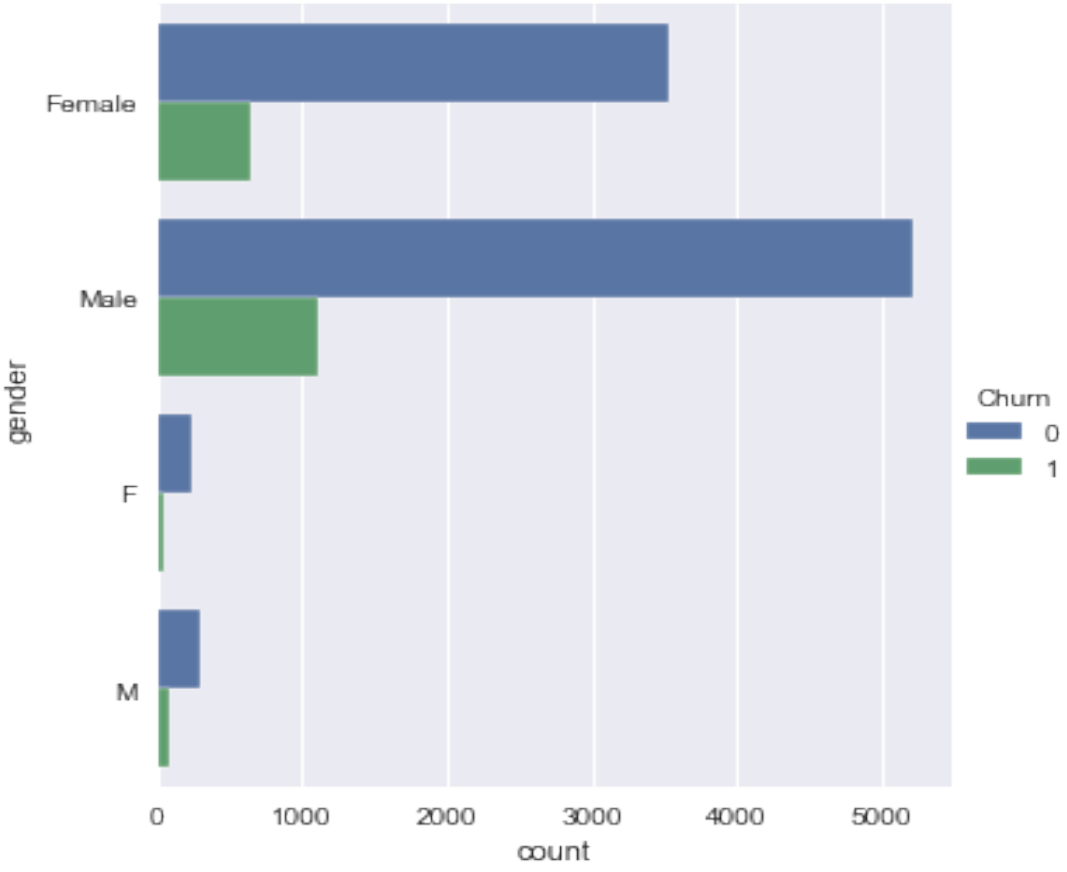
Bi-variate analysis:

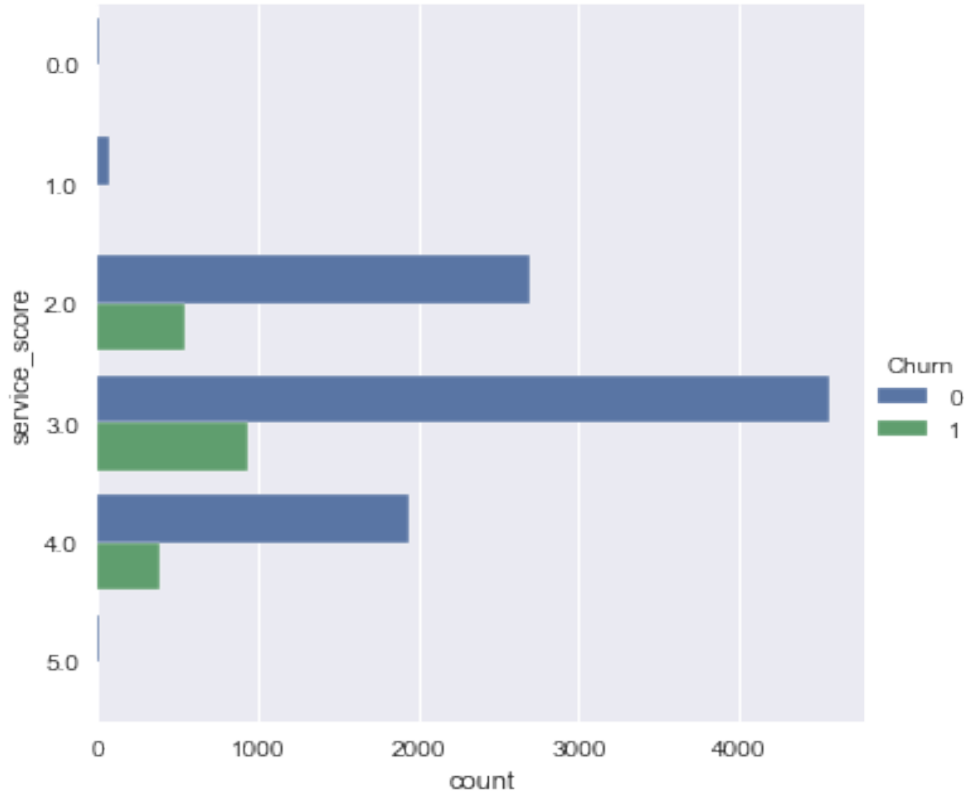
Fig 5: Pair plot across categorical variables

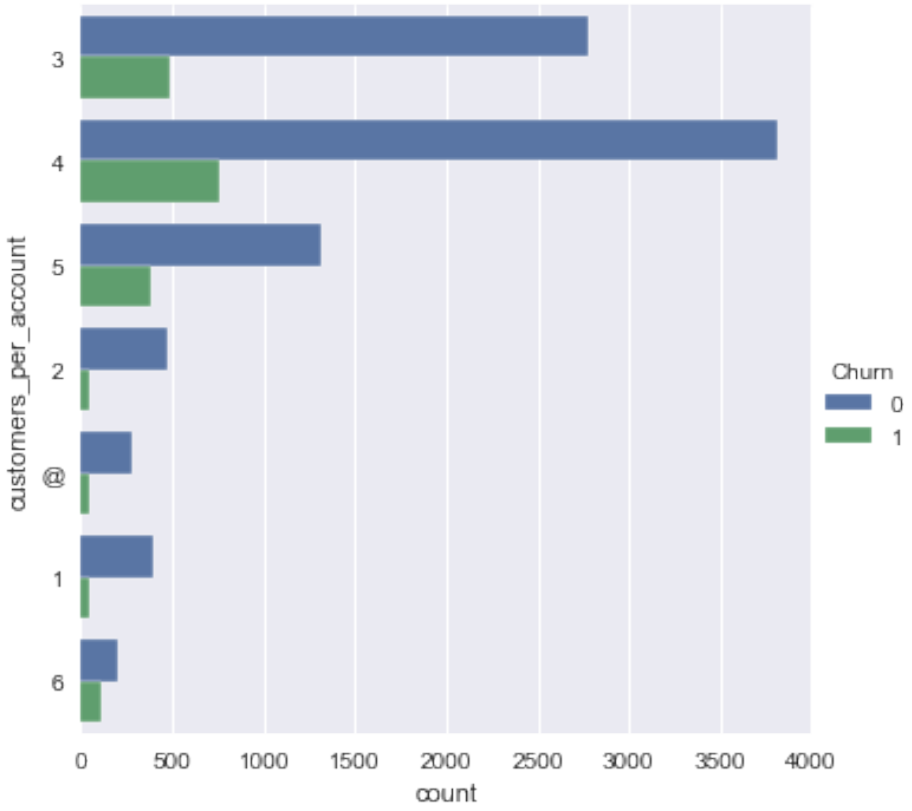
The pair-plot depicted above indicates that the independent variables are insufficient or poor predictors of the target variable since their densities overlap.

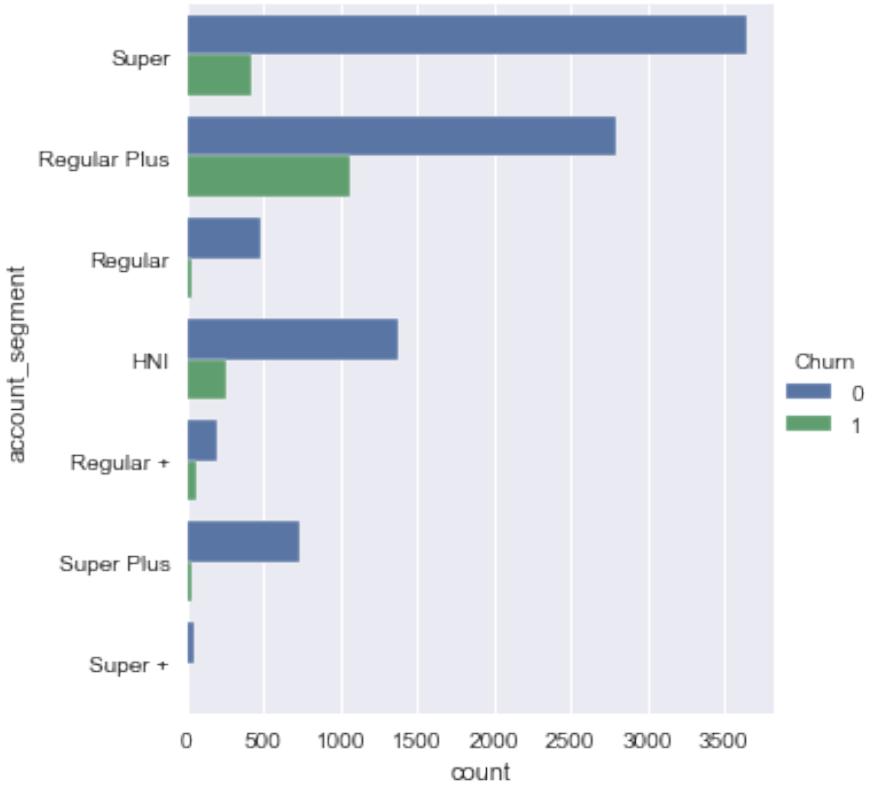

city_tier
1.0
2.0
3.0
0
1000
2000
3000
4000
5000
6000
count
Churn
0
1

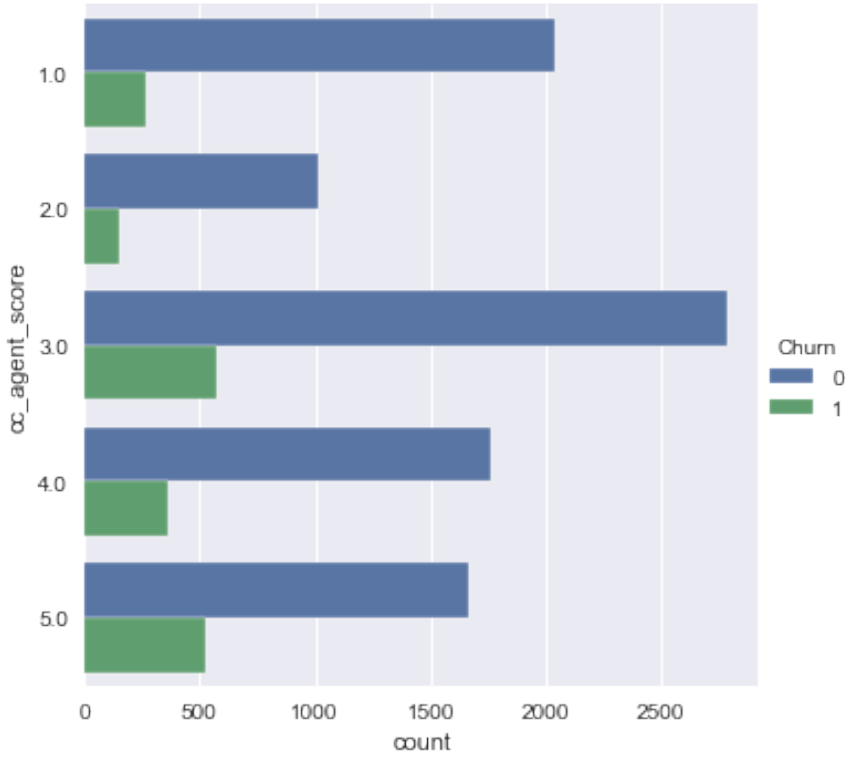

payment_method
Debit Card
UPI
Credit Card
Cash on Delivery
E wallet
0
1000
2000
3000
4000
count
Churn
0
1

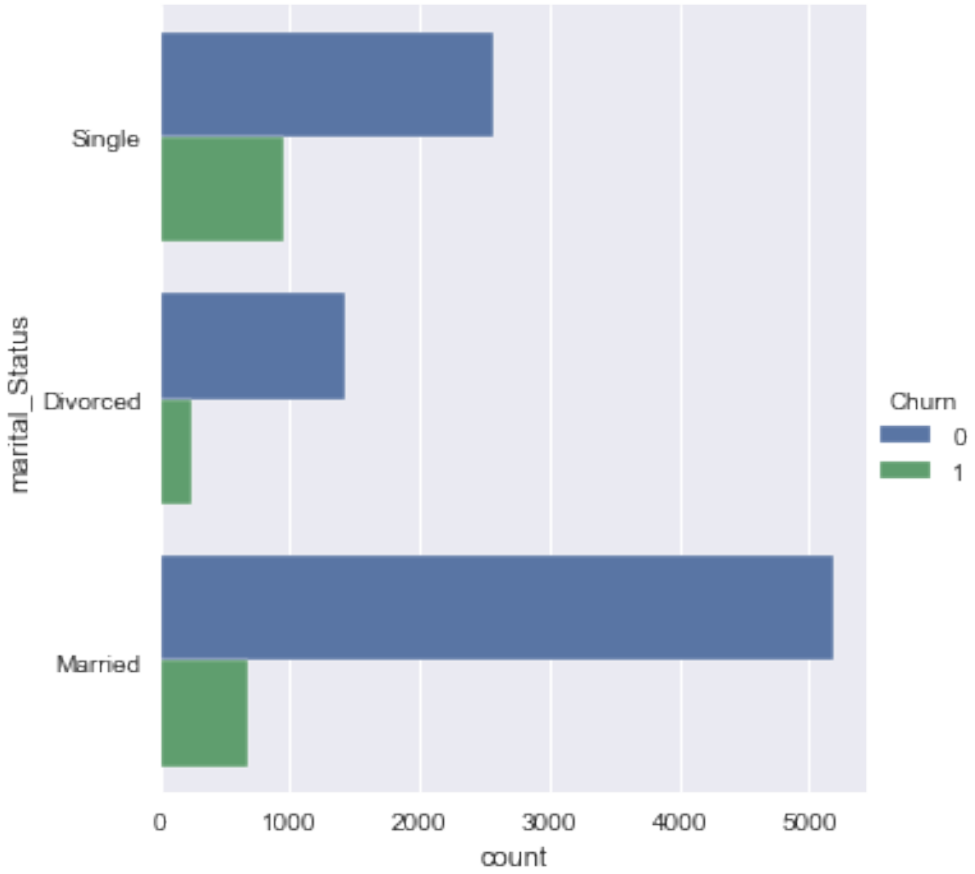

Female
Male
F
M
gender
0
1000
2000
3000
4000
5000
count
Churn
0
1


0.0
1.0
2.0
3.0
4.0
5.0
service_score
0
1000
2000
3000
4000
count
Churn
0
1


customers_per_account
3
4
5
2
@
1
6
0
500
1000
1500
2000
2500
3000
3500
4000
count
Churn
0
1


account_segment
Super
Regular Plus
Regular
HNI
Regular +
Super Plus
Super +
0
500
1000
1500
2000
2500
3000
3500
count
Churn
0
1


cc_agent_score
1.0
2.0
3.0
4.0
5.0
0
500
1000
1500
2000
2500
count
Churn
0
1


marital_Status
Single
Divorced
Married
0
1000
2000
3000
4000
5000
count
Churn
0
1

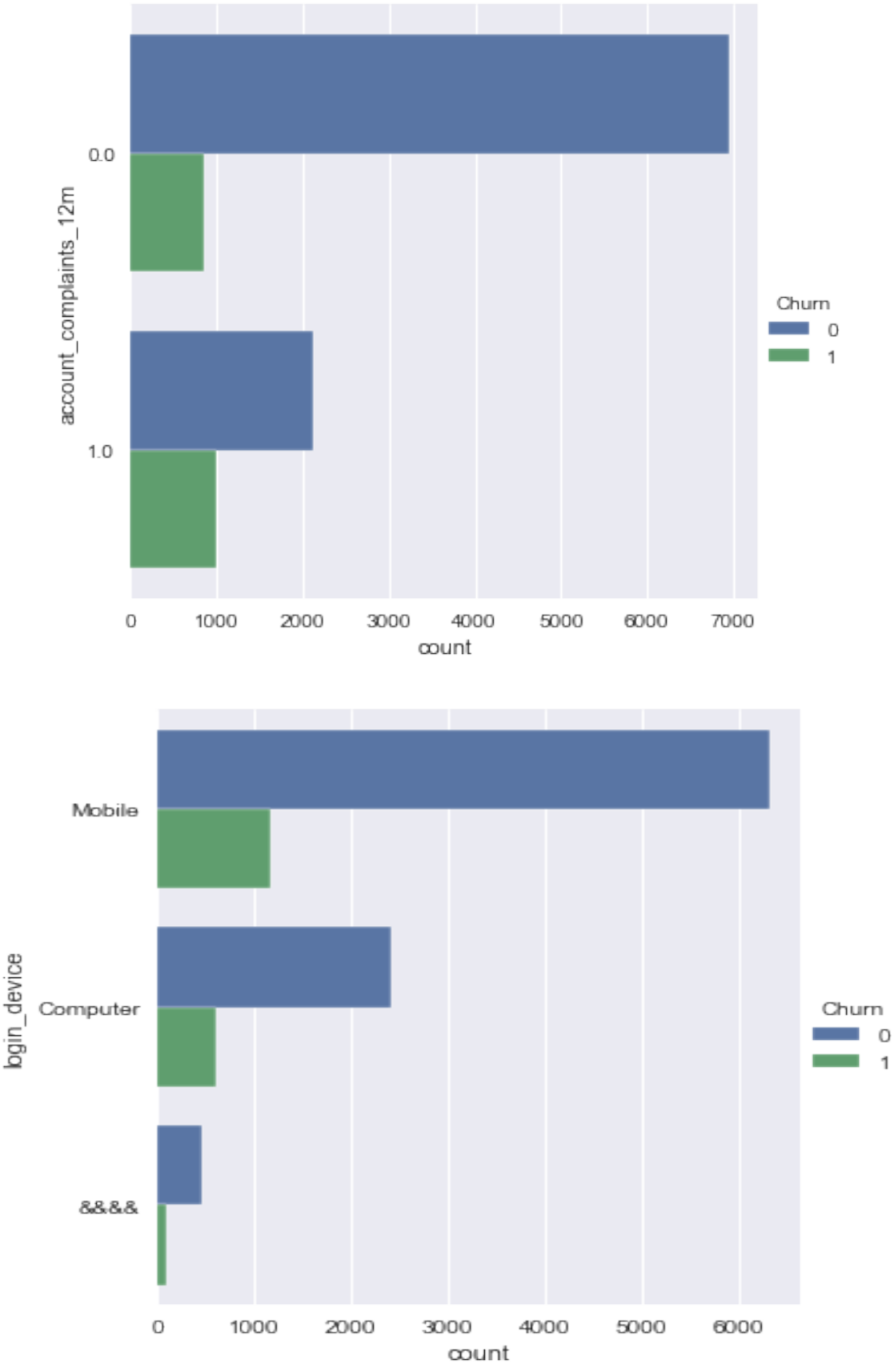


Fig 6: Contribution of categorical variables towards churn

Observation:

- Comparing "1" to "2" and "3," City tier "1" has displayed the most churning.
- Customers who prefer to pay using a "debit card" or "credit card" are more likely to leave the company.
- Consumers that identify as "Male" have a higher churn rate than do female customers.
- The "Regular Plus" segment's customers are churning out greater turnover.
- Customers who are single are more likely to leave than those who are divorced or married.
- Consumers that use the service through mobile see increased churn.

Correlation among variables:

Following the treatment of flawed data and missing values, correlation analysis was done between the variables. We also transformed into integer data types to test for correlation because categorical data won't display in the following images.

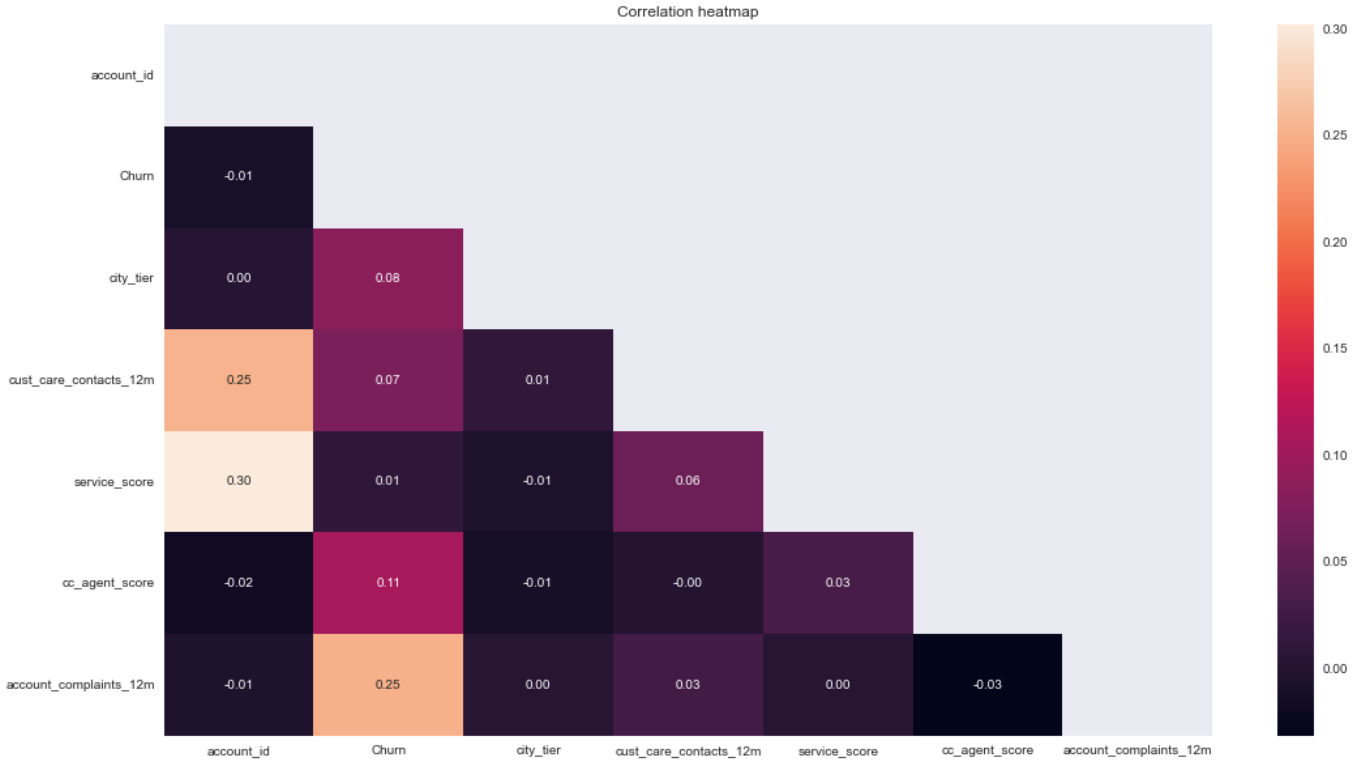


Fig 7: Correlation among variables

Observation:

- Variable “account_tenure” shows high co-relation with Churn.
- Variable “marital_Status” shows high co-relation with churn.
- Variable “account_complaints_2m” shows high- correlation with churn.

Business insights using clustering

- K-means cluster was used to create 3 clusters, and customers were divided into these 3 divisions.

- Based on the inertia value, 3 clusters were chosen.
- The second cluster has the highest customer counts, and the third cluster has the lowest.

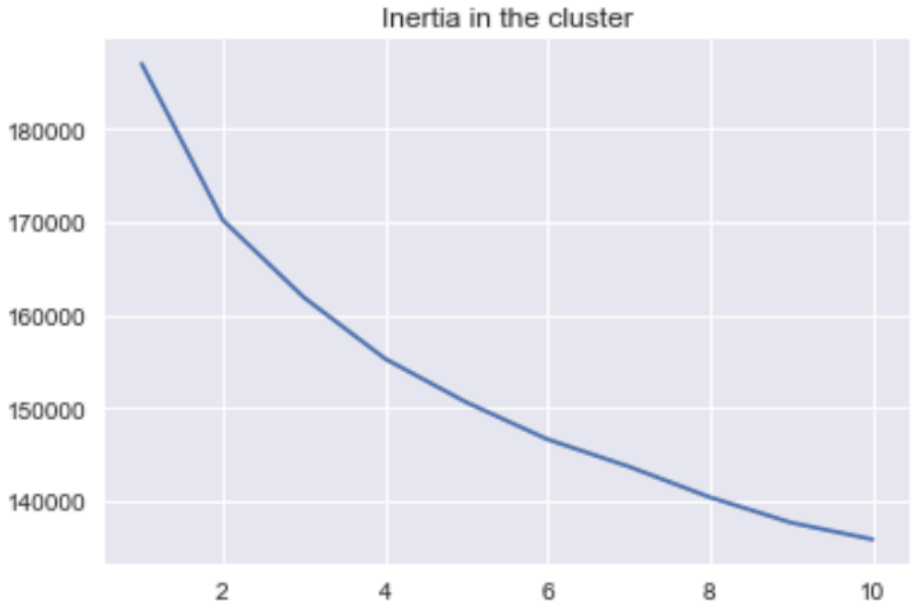


Fig. 8 Plotting clusters

Below table represents the clusters and the columns' count in each cluster

Clus_kmeans	0	1	2
account_tenure	2760	4976	3265
city_tier	2760	4976	3265
cust_care_contacts_12m	2760	4976	3265
payment_method	2760	4976	3265
gender	2760	4976	3265
service_score	2760	4976	3265
customers_per_account	2760	4976	3265
account_segment	2760	4976	3265
cc_agent_score	2760	4976	3265
marital_Status	2760	4976	3265
revenue_per_month	2760	4976	3265
account_complaints_12m	2760	4976	3265
rev_growth_yoy	2760	4976	3265
coupons_used	2760	4976	3265
days_since_cc_contact	2760	4976	3265
cashback	2760	4976	3265
login_device	2760	4976	3265

Fig. 9 K means clustering across all variables

3. Data Cleaning and Pre-processing

This dataset combines categorical and continuous variables. Because each category represents a certain customer type, applying outlier treatment to categorical variables is completely illogical. Therefore, we only apply outlier treatment to variables that are continuous in nature.

- Box plot was used to identify whether an outlier was present in a variable.
- The outlier in the variable is represented by the dots outside a quantile's upper bound.
- Eight continuous variables, including "account_tenure," " cust_care_contacts_12m," " customers_per_account," "cashback," " revenue_per_month," " days_since_cc_contact," "coupons_used" and "rev_growth_yoy," are included in the dataset.
- To eliminate outliers, we employed upper limit and lower limit. The visual depiction of variables both before and after outlier correction is shown below

Removal of unwanted variables:

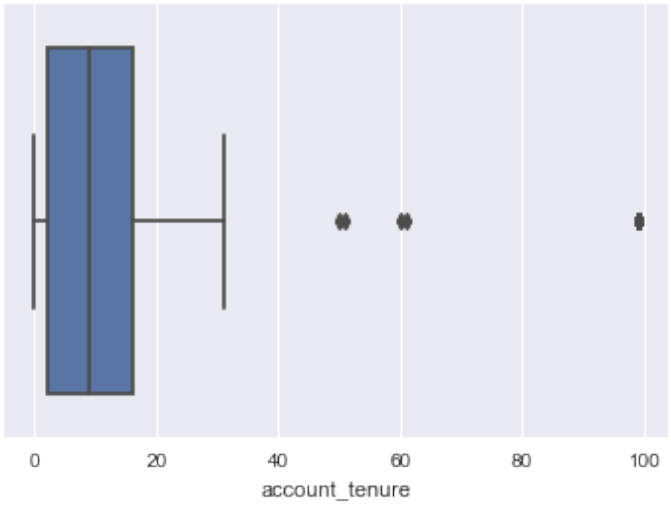
After carefully analysing the data, we come to the conclusion that at this point in the project, removing variables is not necessary. We can get rid of the variable "AccountID," which stands for a special ID given to special clients. But doing so will result in 8 duplicate rows. Looking at the univariate and bi-variate analyses, the rest of the factors appear to be significant.

Proportion of outliers present before outlier treatment are shown below:

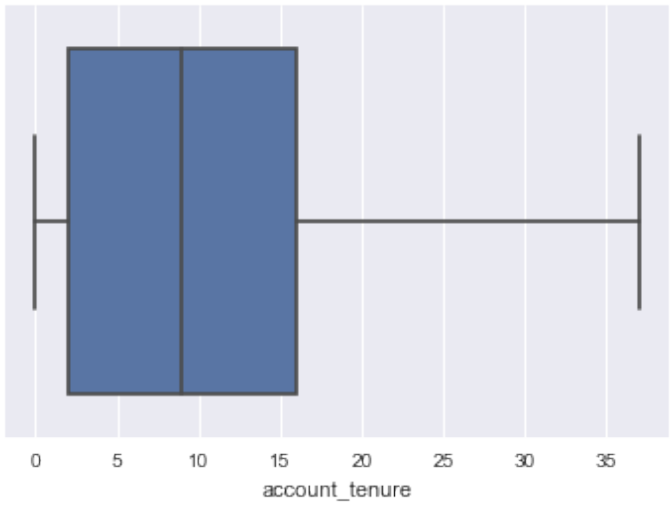
	outlier %
Churn	16.83
account_tenure	1.26
city_tier	0.00
cust_care_contacts_12m	0.38
payment_method	0.00
gender	0.00
service_score	0.12
customers_per_account	6.73
account_segment	14.68
cc_agent_score	0.00
marital_Status	0.00
revenue_per_month	1.68
account_complaints_12m	0.00
rev_growth_yoy	0.00
coupons_used	12.42
days_since_cc_contact	1.16
cashback	8.59
login_device	0.00

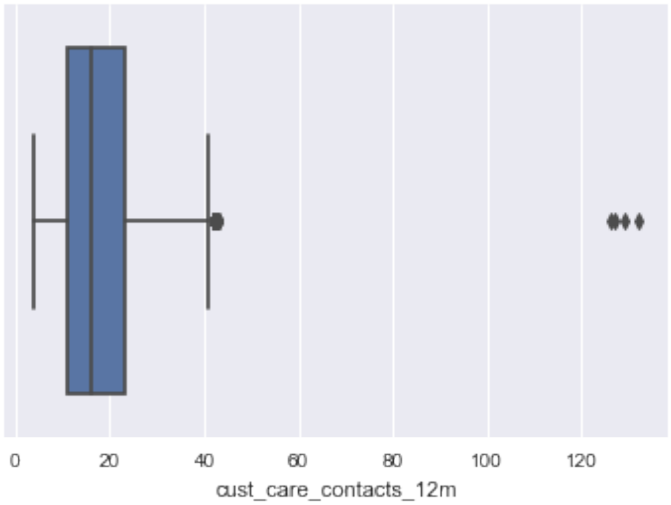
Table 6: Proportion of outliers present in the dataset

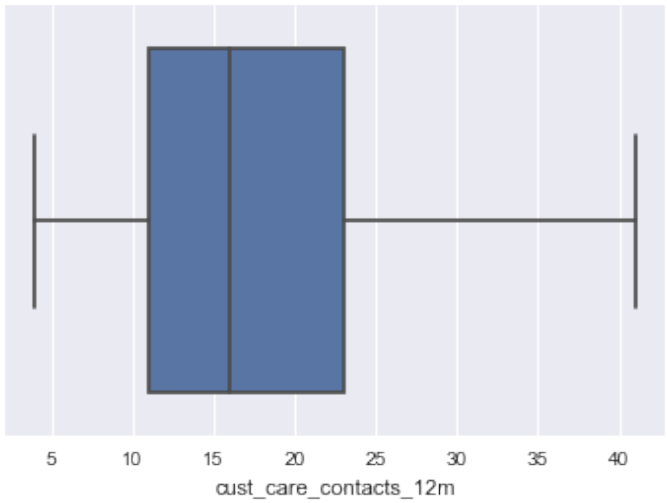
BEFORE

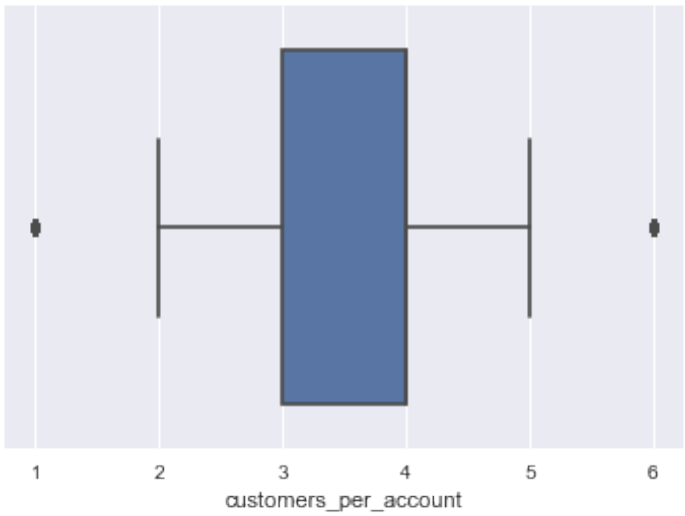


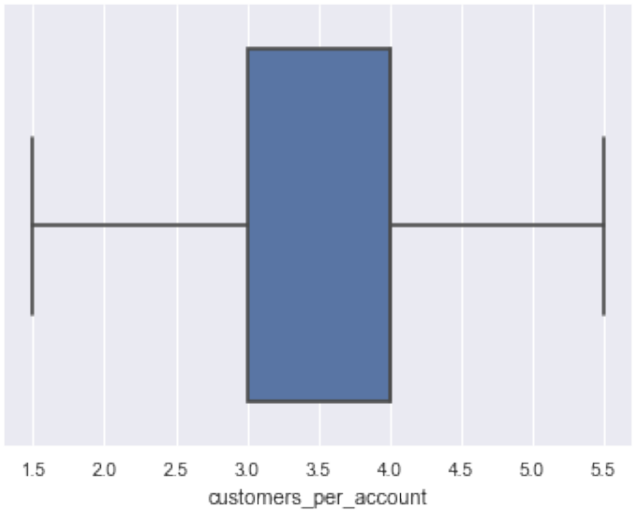
AFTER

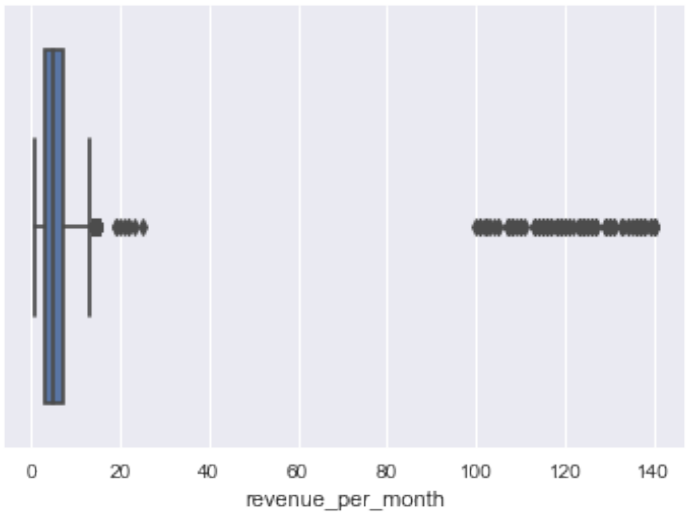


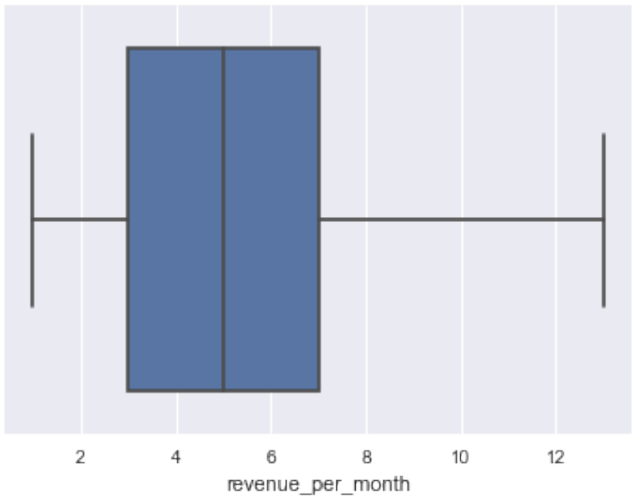


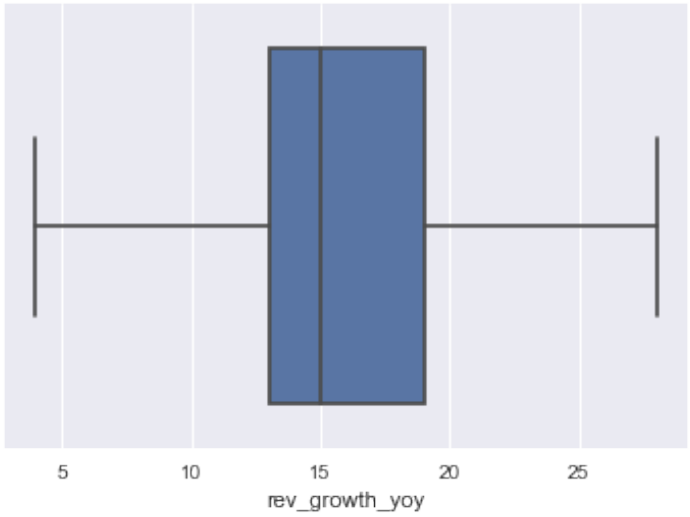


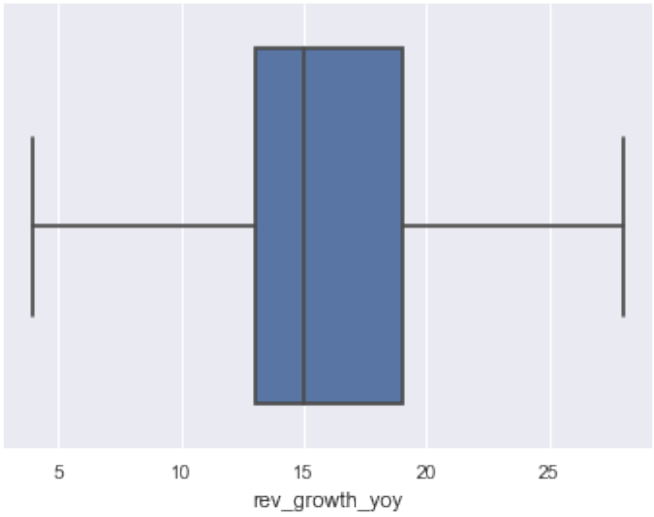


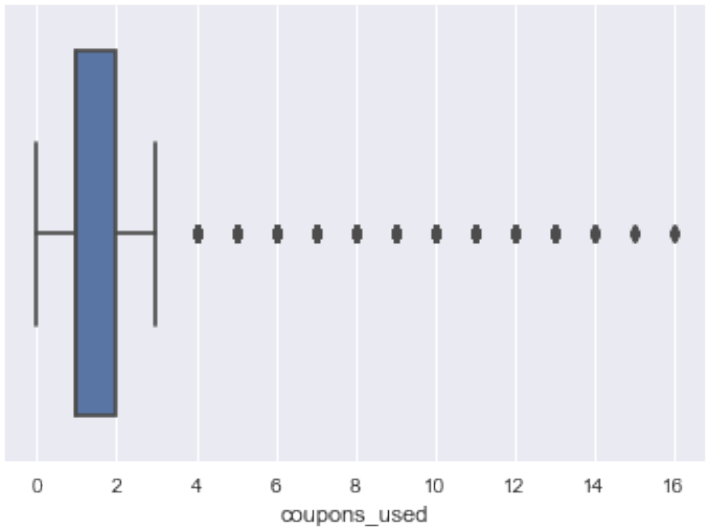


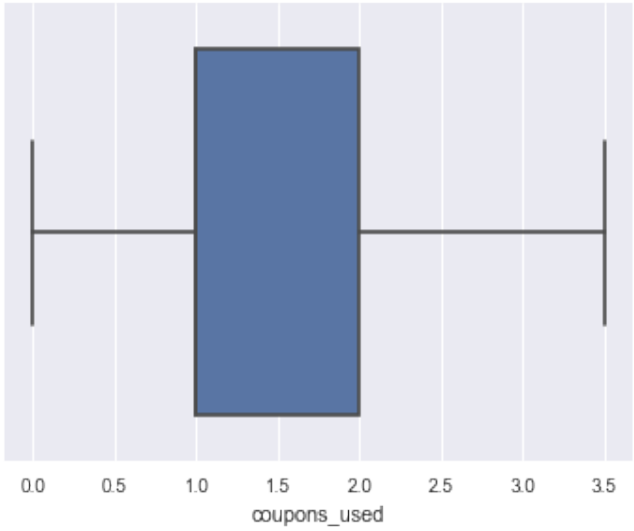

0
20
40
60
80
100
120
140
revenue_per_month


2
4
6
8
10
12
revenue_per_month


5
10
15
20
25
rev_growth_yoy


5
10
15
20
25
rev_growth_yoy


0
2
4
6
8
10
12
14
16
coupons_used


0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
coupons_used

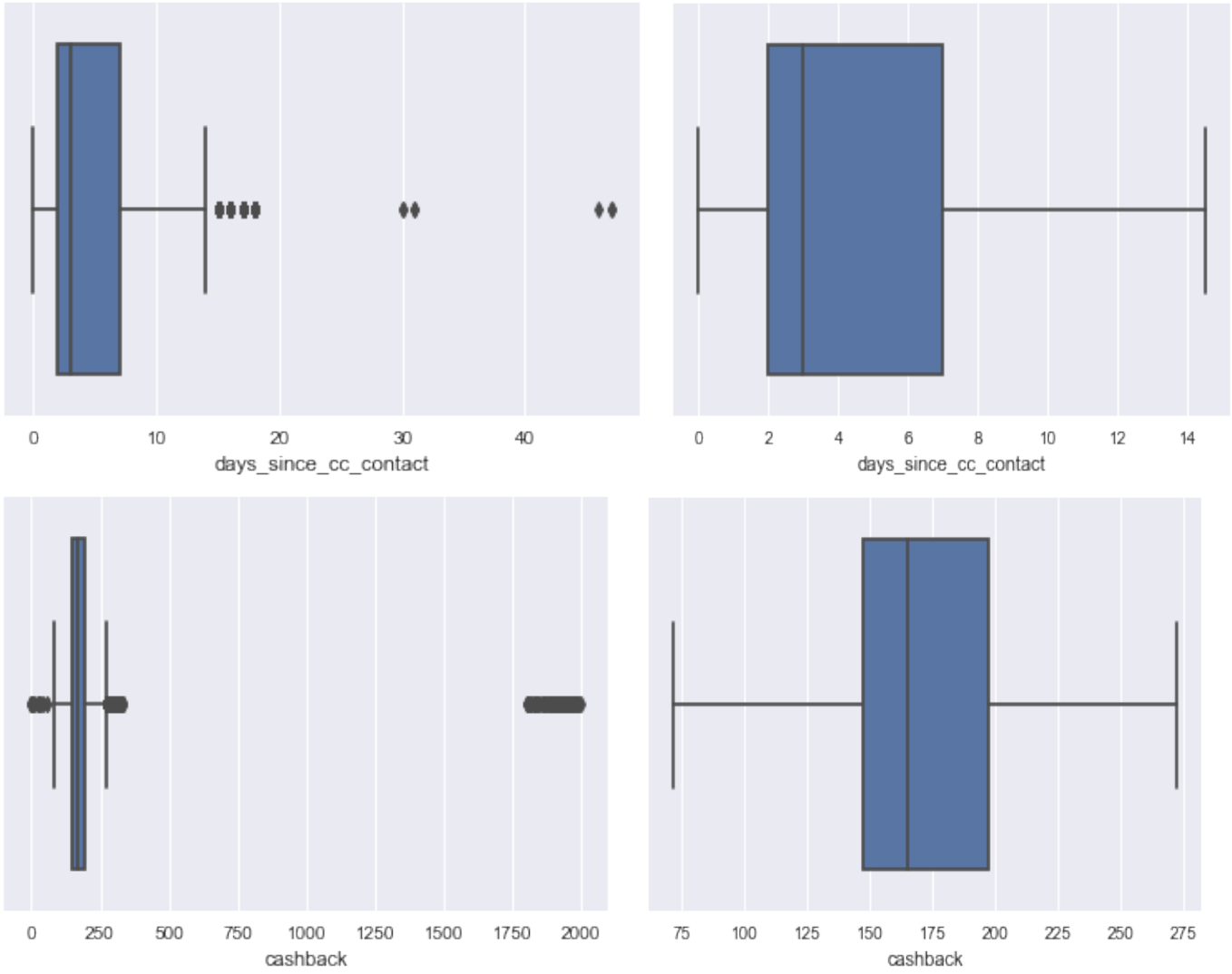


Fig 10: Before and after outlier treatment

Removing duplicates

There are 259 rows with duplicate values. Hence, we can drop them

Number of duplicate rows: 259

After dropping the duplicates,

Number of duplicate rows = 0

And the total rows of the dataset got decreased to 11001 from 11260

Missing value treatment

- We have data anomalies in 17 of the 19 variables, and null values in 15 of the variables.

- Using the median instead of the mean when the variable is continuous because the median is less likely to contain outliers than the mean.
- In situations where variables are categorical in nature, using "Mode: to impute null values.
- We have handled null values separately for each variable because each one is different in its own way.

Count of null values after null value treatment

```
Churn                       0
account_tenure              0
city_tier                   0
cust_care_contacts_12m      0
payment_method              0
gender                      0
service_score               0
customers_per_account       0
account_segment             0
cc_agent_score              0
marital_Status              0
revenue_per_month           0
account_complaints_12m      0
rev_growth_yoy              0
coupons_used                0
days_since_cc_contact       0
cashback                    0
login_device                0
dtype: int64
```

Fig 11: After null value treatment

We see NIL null values across variable which indicated that the data is now cleaned and we can move further for data transformation of required.

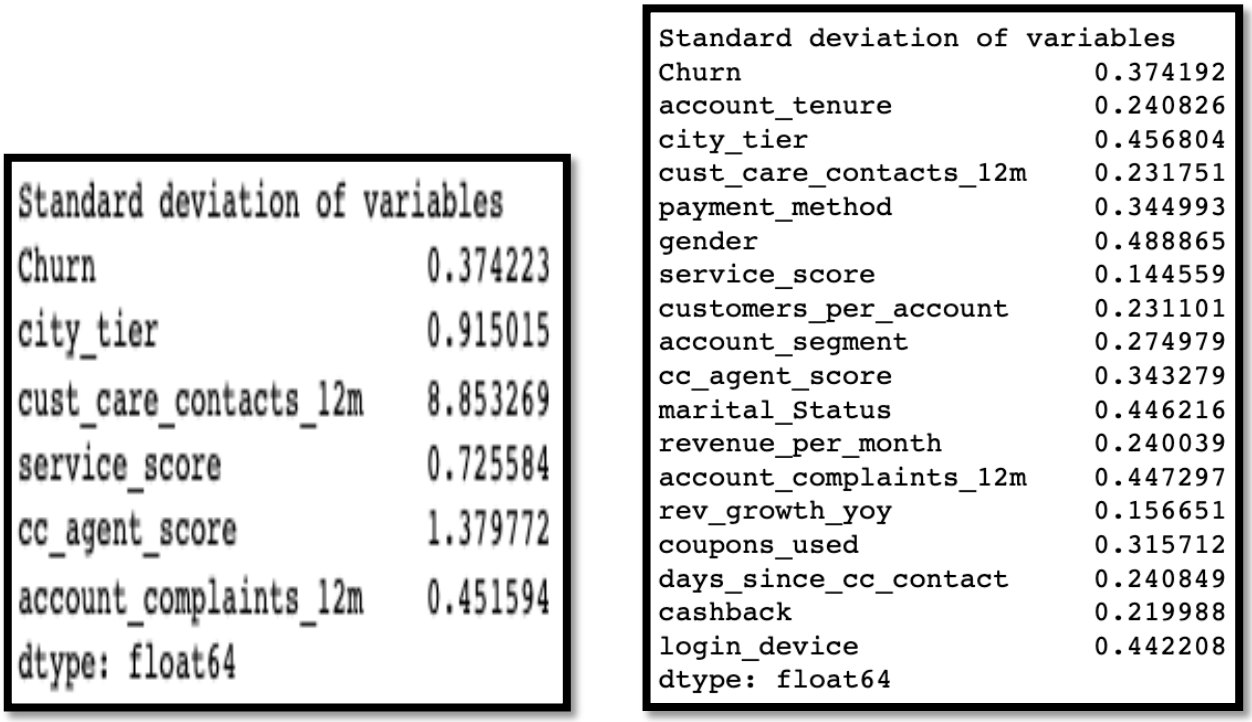
Variable transformation:

- We see that the different variable has different dimensions. Like variable “Cashback” denotes currency where as “CC_Agent_Score” denotes rating provided by the customers. Due to which they differ in their statistical rating as well
- Scaling would be required for this data set which in turn will normalize the date and standard deviation will be close to “0”

Using MinMax scaling to perform normalization of data.

Standard Deviation Before and After Normalization:

Before **After**



```
Standard deviation of variables
Churn                      0.374223
city_tier                  0.915015
cust_care_contacts_12m     8.853269
service_score              0.725584
cc_agent_score             1.379772
account_complaints_12m     0.451594
dtype: float64
```

```
Standard deviation of variables
Churn                      0.374192
account_tenure             0.240826
city_tier                  0.456804
cust_care_contacts_12m     0.231751
payment_method             0.344993
gender                     0.488865
service_score              0.144559
customers_per_account      0.231101
account_segment            0.274979
cc_agent_score             0.343279
marital_Status             0.446216
revenue_per_month          0.240039
account_complaints_12m     0.447297
rev_growth_yoy             0.156651
coupons_used               0.315712
days_since_cc_contact      0.240849
cashback                   0.219988
login_device               0.442208
dtype: float64
```

Fig 12: Before and after normalization

We can see that the standard deviation of the variables is now very near to zero. We also transformed the variables to the int data type, which will aid in the process of creating the model in the future.

Addition of new variables

At this time, we don't see the need to add any additional variables in the traditional sense. may be necessary at a later stage of model construction and can be made in that manner.

4. Model Building

We may infer from the aforementioned visual and non-visual analyses that the target variable has to be categorised as "Yes" or "No" in a classification model.
As data analysts, we may use the following algorithms to create the required mechanism to determine if a certain consumer will leave us or not:

Logistic Regression: The "Supervised machine learning" algorithm of logistic regression may be used to estimate the likelihood of a certain class or occurrence. It is applied when the outcome is binary or dichotomous and the data may be linearly separated. It implies problems involving binary classification are typically solved using logistic regression.

Linear Discriminant Analysis: Linear Discriminant Analysis, or LDA for short, is a predictive modelling algorithm for multi-class classification. It can also be used as a dimensionality

reduction technique, providing a projection of a training dataset that best separates the examples by their assigned class.

KNN: KNN calculates the distances between a query and each example in the data, chooses the K instances closest to the query, and then votes for the label with the highest frequency (in the case of classification) or averages the labels (in the case of regression).

Bagging (Random Forest) – Bagging, sometimes referred to as bootstrap aggregation, is a typical ensemble learning technique for lowering variance in noisy datasets. In bagging, a training set's data is randomly sampled and replaced, allowing for multiple selections of the same data points.

Ada Boosting: The method of boosting includes strengthening the power of a machine learning software by introducing more complicated or competent algorithms. Machine learning outcomes can be improved by using this approach to lessen bias and volatility.

Gradient Boosting: One kind of machine learning boosting is gradient boosting. It is predicated on the hunch that when prior models are coupled with the best feasible upcoming model, the overall prediction error is minimised. The next target outcome of the case is zero if a modest modification in the forecast for a case results in no change in error.

Decision tree: One of the supervised machine learning algorithms which can be used for regression and classification problems, yet, is mostly used for classification problems. A decision tree follows a set of if-else conditions to visualize the data and classify it according to the conditions.

ANN: ANNs are nonlinear statistical models which display a complex relationship between the inputs and outputs to discover a new pattern. A variety of tasks such as image recognition, speech recognition, machine translation as well as medical diagnosis makes use of these artificial neural networks.

Splitting Data into Train and Test Dataset

Following the accepted market practice, we have divided data into Train and Test dataset into 70:30 ratio and building various models on training dataset and testing for accuracy over testing dataset.

Below is the shape of Train and Test dataset:

```
X_train (7700, 17)
X_test (3301, 17)
y_train (7700,)
y_test (3301,)
```

Fig 13: Shape of training and test dataset

Is the data unbalanced? If so, what can be done?

The presented data set is unbalanced. Our target variable, "Churn," has a considerable degree of variation in

its categorical count. We have 9364 for "0" and 1896 for "1" in our count.

```
0    9149
1    1852
```

Fig. 14 Value counts of variable Churn

Using the SMOTE technique, it is possible to correct this dataset imbalance by generating more datapoints.

We must only use SMOTE on the train dataset; not the test dataset. As a standard business procedure, data were separated into train and test datasets in a 70:30 ratio (can be changed later as instructed).

Before SMOTE:

X_train (7700, 17)
X_test (3301, 17)
y_train (7700,)
y_test (3301,)

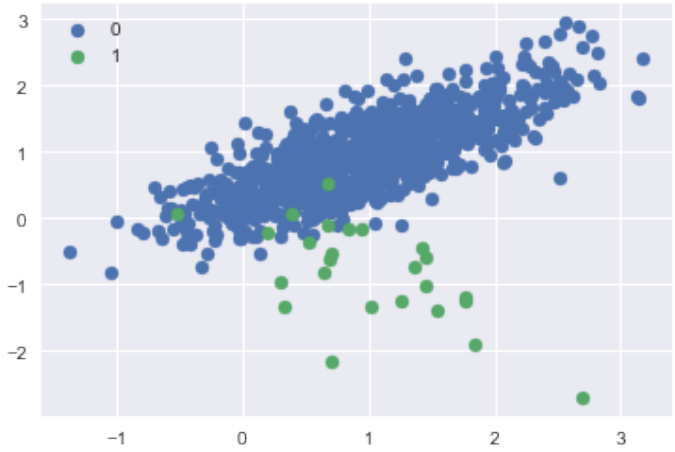


Fig. 15 Scatter plot before SMOTE

After SMOTE:

X_train_res (12812, 17)
y_train_res (12812,)

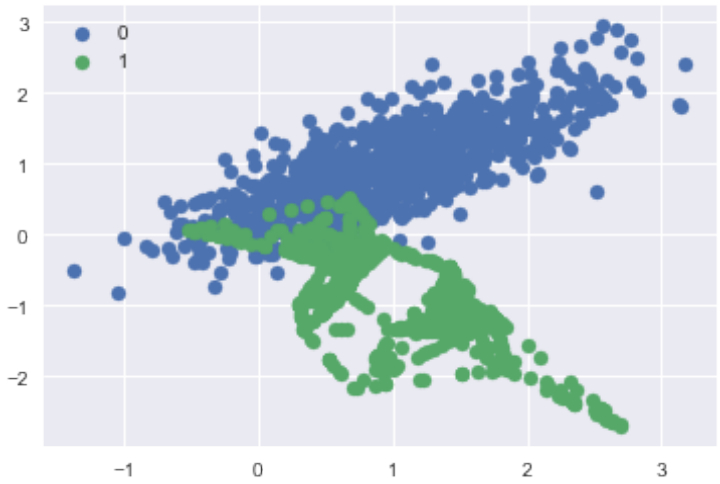


Fig. 16 Scatter plot after SMOTE

The increase in density of the green dots indicates the increase in data points.

Building tuned Gradient Boosting model

The majority of the accessible hyperparameters are same to random forest classifier.

Initial predictions are calculated using an estimator object called init. The initial raw predictions are set to zero if the value is "zero". A DummyEstimator that forecasts the class priors is used by default.

Gradient boosting lacks a class weights argument.

```
▼                    GradientBoostingClassifier
GradientBoostingClassifier(criterion='mse', learning_rate=0.5, lo
ss='deviance',
                           max_depth=9, max_features=11, min_samp
les_leaf=6,
                           min_samples_split=15, n_estimators=20
1,
                           random_state=0)
```

Fig.17 Hyper parameters used for tuned Gradient Boost Classifier

Metrics score

```
Accuracy on training set :  1.0
Accuracy on test set :  0.98242956679794
Recall on training set :  1.0
Recall on test set :  0.9247311827956989
Precision on training set :  1.0
Precision on test set :  0.9699248120300752
F1 on training set :  1.0
F1 on test set :  0.9467889908256881
```

Fig.18 Metrics score of tuned Gradient Boost Classifier

Classification report of training set

```
Classification Report of Training Data
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00      6406
         1.0       1.00      1.00      1.00      1294

    accuracy                           1.00      7700
   macro avg       1.00      1.00      1.00      7700
weighted avg       1.00      1.00      1.00      7700
```

Fig.19 Classification report of training data

Confusion matrix of training set:

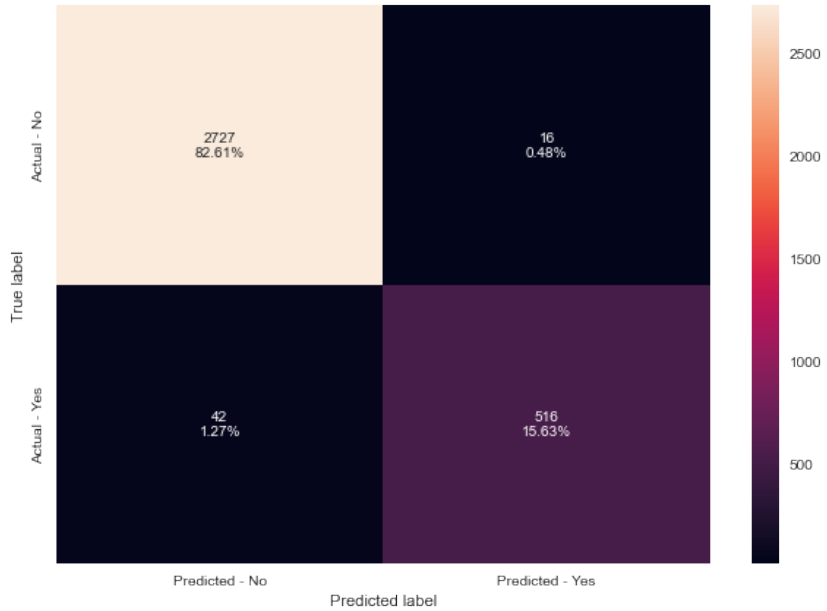


Fig.20 Confusion matrix of training data

Classification report of testing data:

Classification Report of Testing Data

	precision	recall	f1-score	support
0.0	0.98	0.99	0.99	2743
1.0	0.97	0.92	0.95	558
accuracy			0.98	3301
macro avg	0.98	0.96	0.97	3301
weighted avg	0.98	0.98	0.98	3301

Fig.21 Classification report of testing data

Confusion matrix of testing data:

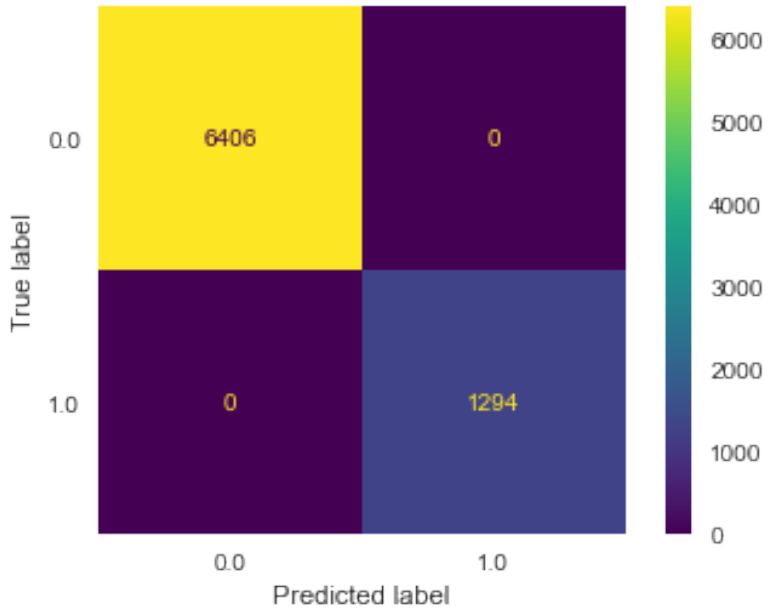


Fig. 22 Confusion matrix of testing data

AUC and ROC curve of training data:

```
AUC: 1.000
```

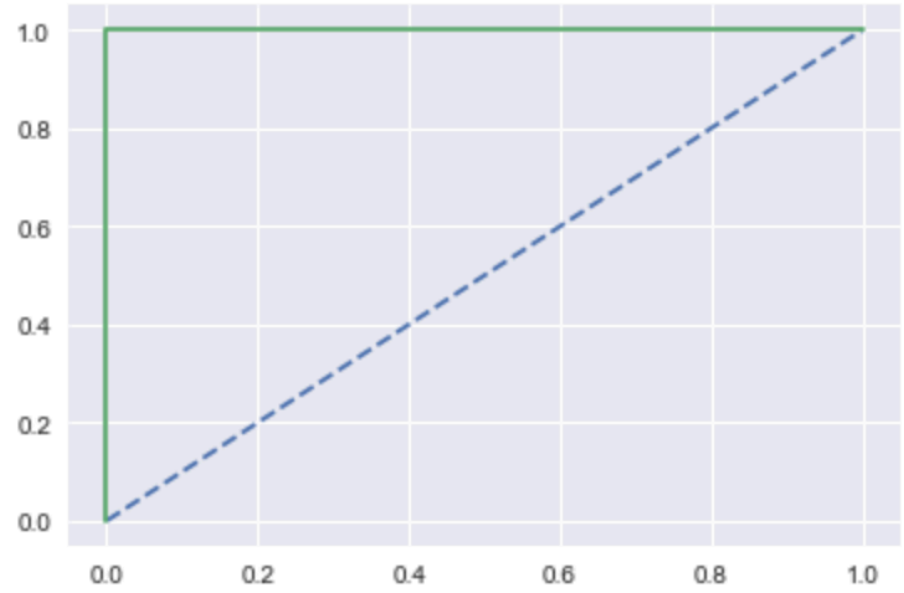


Fig.23 AUC and ROC curve of training data

AUC and ROC curve of testing data:

```
AUC: 0.996
```

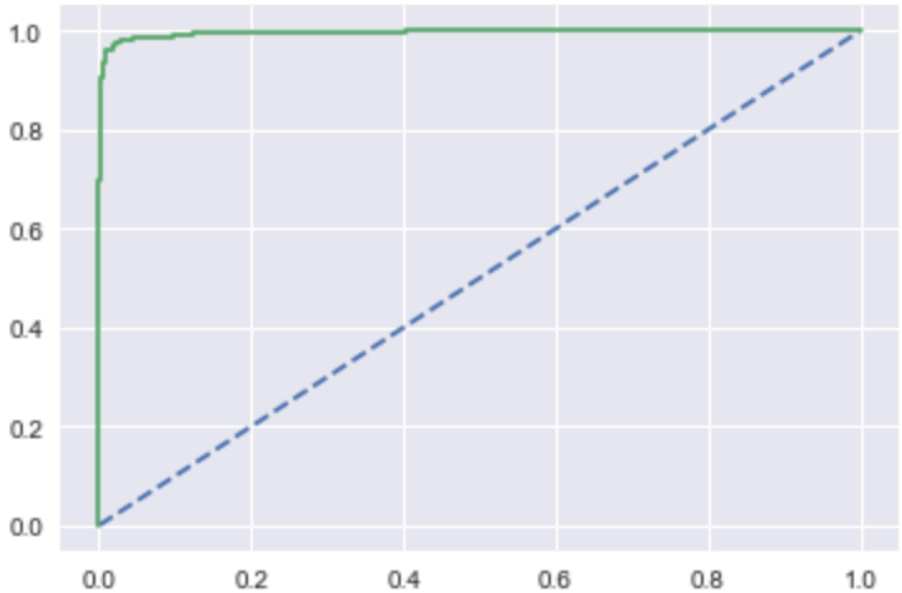


Fig. 24 AUC and ROC curve of testing data

Feature Importance

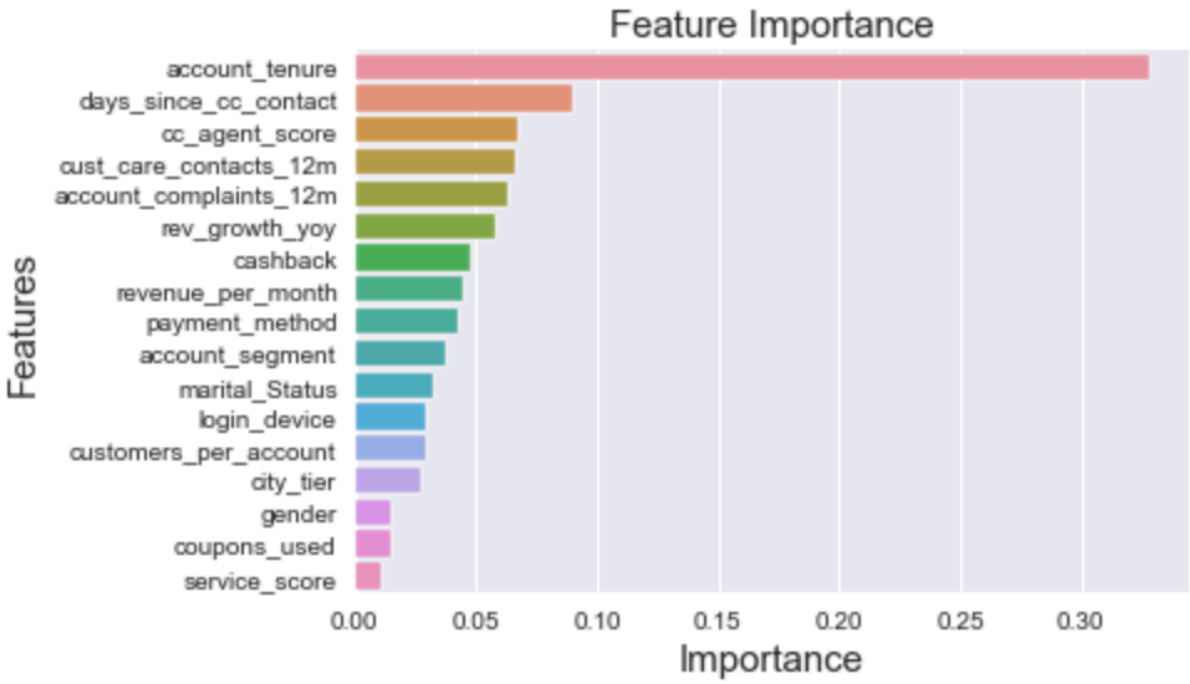


Fig. 25 Feature importance of tuned Gradient boost model

OBSERVATIONS FROM FEATURE IMPORTANCE:

- This model has picked account_tenure to be the most important feature followed by days_since_cc_contact an cc_agent_score.
- According to EDA, the turnover is higher for shorter tenures, particularly in the first year. In order to maintain a customer's satisfaction after acquisition, the first year is crucial.
- Compared to current customers, churned consumers had more recent interactions with customer service. According to EDA, active clients have a longer median number of days since their previous connection.
- Prior to churning, recently contacted consumers have called customer service. According to the tuned gradient boosting model, the customer satisfaction rating is also significant. This is in line with the trends and revelations that came from EDA.

INFERENCE:

- This model has achieved near-perfect accuracy on the training dataset (1.00), which suggests that it has likely overfit the training data.
- However, it still has very high accuracy on the testing dataset (0.98), which suggests that it is able to generalize well to unseen data.
- The recall score for both training and testing datasets is high, indicating that the model is able to identify most of the positive cases.
- The precision score for both training and testing datasets is also high, suggesting that the model does not produce many false positives.
- The F1 score for the testing dataset is slightly lower than the training dataset, but still relatively high, indicating a good balance between precision and recall.

- Overall, this model seems to perform very well, with good generalization to new data and a good balance between precision and recall.

Comparison of models across parameters

Model	Training Dataset				Testing Dataset			
	Accuracy	**Recall**	**Precision**	**F1-Score**	**Accuracy**	**Recall**	**Precision**	**F1-Score**
Gradient Boosting tuned	1.00	0.98	1.00	0.92	1.00	0.97	1.00	0.95
Bagging with Decision tree	1.00	0.97	1.00	0.87	1.00	0.95	1.00	0.91
Random Forest	1.00	0.97	1.00	0.86	1.00	0.95	1.00	0.91
XGBoost	1.00	0.97	1.00	0.89	1.00	0.92	1.00	0.90
XGBoost Tuned	1.00	0.97	1.00	0.91	1.00	0.92	1.00	0.91
Random Forest Tuned	0.96	0.93	0.78	0.67	0.98	0.91	0.87	0.77
ANN Tuned	0.98	0.88	0.96	0.87	0.99	0.90	0.98	0.
KNN	0.98	0.95	0.90	0.80	0.97	0.90	0.93	0.85
KNN Tuned	0.99	0.96	0.94	0.89	0.97	0.90	0.96	0.90
ANN	0.95	0.93	0.78	0.72	0.90	0.83	0.83	0.77

AdaBoost Tuned	1.00	0.94	1.00	0.86	1.00	0.81	1.00	0.83
Gradient boosting with default parameters	0.92	0.90	0.62	0.58	0.85	0.79	0.72	0.67
Logistic Regression	0.88	0.88	0.43	0.45	0.75	0.74	0.55	0.56
LDA	0.88	0.88	0.41	0.44	0.74	0.74	0.53	0.55
AdaBoost with default parameters	0.90	0.89	0.58	0.59	0.74	0.73	0.65	0.65
Logistic Regression Tuned	0.88	0.88	0.45	0.46	0.74	0.73	0.56	0.57
Logistic Regression with Smote	0.79	0.77	0.82	0.81	0.43	0.41	0.56	0.55

Table 7 Comparison of models with parameters

- Based on the above comparison table, the Gradient Boosting model with tuned parameters is the best suited for a classification problem like Customer Churn analysis.
- This is because it has the highest accuracy score on the testing dataset (0.98), which indicates that it can accurately predict whether a customer is likely to churn or not.
- Additionally, it also has high recall (0.92) and precision (0.97) scores on the testing dataset, which means that it has a low false negative rate and a low false positive rate, respectively. This is important in a churn analysis because the cost of retaining a customer who was predicted to churn is lower than losing a customer who was predicted to stay.
- Therefore, a model with high precision and recall is preferable for this problem.
- Finally, the high F1 score (0.95) on the testing dataset suggests that this model has a good balance between precision and recall, making it the best optimized model for the customer churn analysis problem.
- Compared to XGBoost-Tuned, Gradient Boosting Tuned has a slightly higher accuracy score on the testing dataset (0.98 vs. 0.97) and higher precision score on the testing dataset (0.97 vs. 0.92). However, XGBoost-Tuned has a slightly higher recall score on the testing dataset (0.91 vs. 0.92) and a comparable F1 score (0.91 vs. 0.95). Overall, the two models

are very similar, but Gradient Boosting Tuned has a slight edge in terms of precision and accuracy.

- Compared to Random Forest, Gradient Boosting Tuned has a higher accuracy score on the testing dataset (0.98 vs. 0.97), a higher recall score on the testing dataset (0.92 vs. 0.86), and a comparable F1 score (0.95 vs. 0.91). However, Random Forest has a higher precision score on the testing dataset (0.95 vs. 0.97). Overall, Gradient Boosting Tuned performs slightly better in terms of accuracy and recall, but Random Forest performs slightly better in terms of precision.
- Compared to Bagging with Decision Tree, Gradient Boosting Tuned has a higher accuracy score on the testing dataset (0.98 vs. 0.97), a higher recall score on the testing dataset (0.92 vs. 0.87), and a higher F1 score on the testing dataset (0.95 vs. 0.91). However, Bagging with Decision Tree has a higher precision score on the testing dataset (0.95 vs. 0.97). Overall, Gradient Boosting Tuned performs better in terms of accuracy, recall, and F1 score, while Bagging with Decision Tree performs slightly better in terms of precision.
- Compared to KNN-Tuned, Gradient Boosting Tuned has a higher accuracy score on the testing dataset (0.98 vs. 0.96), a higher recall score on the testing dataset (0.92 vs. 0.89), and a higher F1 score on the testing dataset (0.95 vs. 0.90). However, KNN-Tuned has a slightly higher precision score on the testing dataset (0.90 vs. 0.97). Overall, Gradient Boosting Tuned performs better in terms of accuracy, recall, and F1 score, while KNN-Tuned performs slightly better in terms of precision.
- Compared to the other models, the default Gradient Boosting model has lower performance scores, including lower accuracy, recall, precision, and F1 scores on both the training and testing datasets. This suggests that the default model is not well-optimized for the churn analysis problem and may benefit from hyperparameter tuning.
- Overall, the **Gradient Boosting Tuned model** performs well across multiple metrics, including accuracy, precision, recall, and F1 score, making it the best optimized model for the customer churn analysis problem.

5. Model Validation

Modelling approach used and why:

- When predicting customer churn, consider the cost of false positives and false negatives.
- The best model balances precision (predicted positives that are actually positive) and recall (actual positives correctly predicted)
- To assess the trade-off between these two criteria and determine which model is the best, the F1-score is a helpful metric
- While the goal is to reduce FN, recall is particularly important in this situation
- Recall values should be used to determine the ideal model

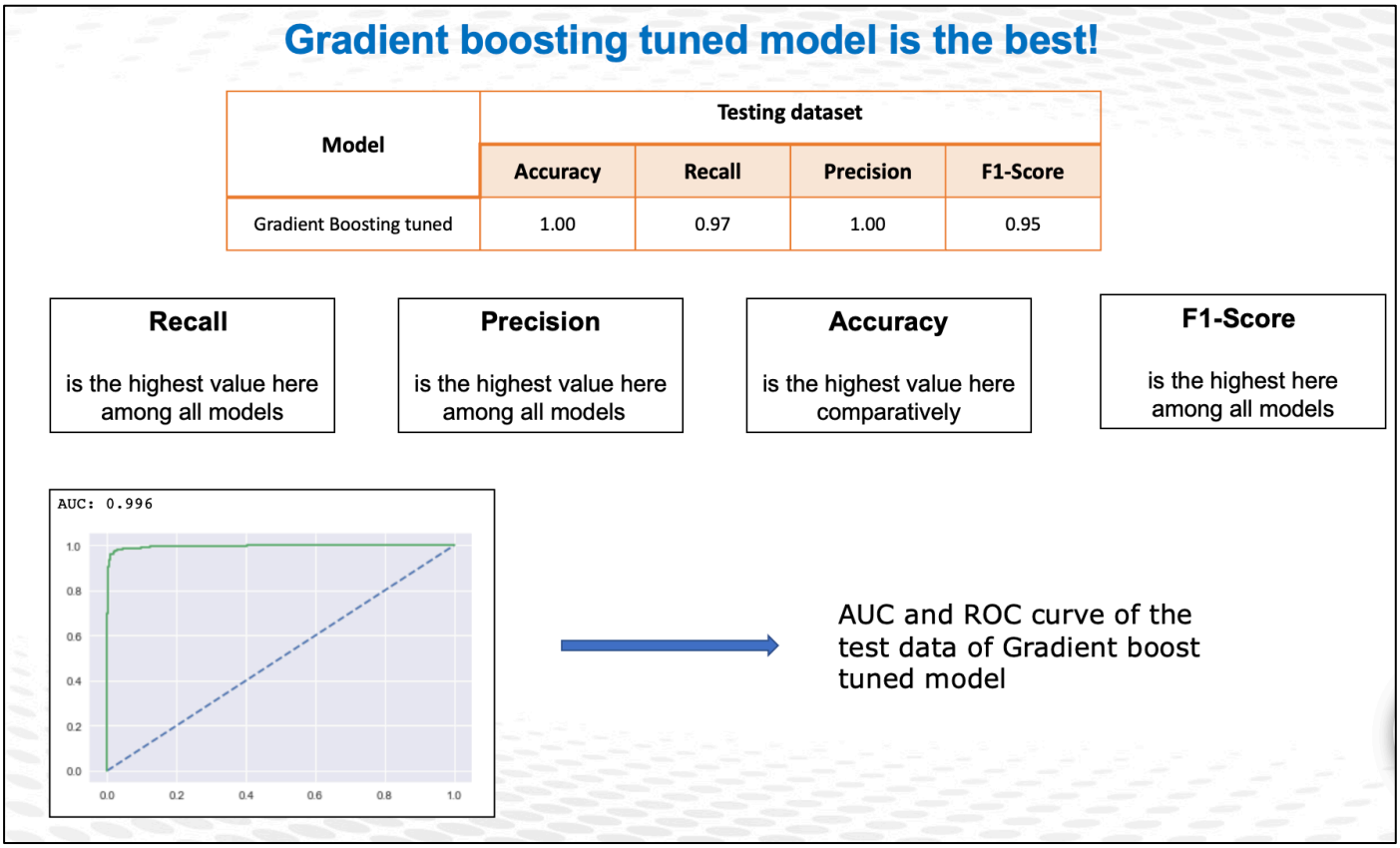


Fig. 26 Metrics summary of Tuned Gradient Boosting model

When it comes to model validation of a classification problem statement we cannot just get relied on accuracy, we need to look at various others parameter like F1 score, Recall, precision, ROC curve and AUC score, along with confusion matrix. The details of these parameters are described below:

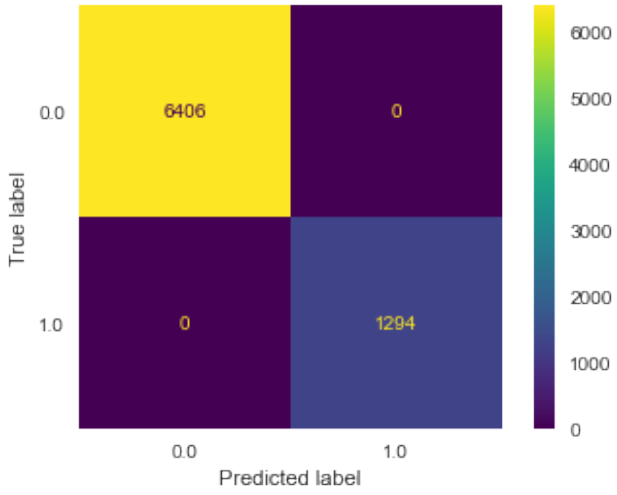


Fig. 27 Confusion matrix for test set in tuned gradient boosting model

Confusion Matrix:

Confusion Matrix usually causes a lot of confusion even in those who are using them regularly. Terms used in defining a confusion matrix are TP, TN, FP, and FN.

True Positive (TP): - The actual is positive in real and at the same time the prediction was classified correctly.
False Positive (FP): - The actual was actually negative but was falsely classified as positive.
True Negative: - The actuals were actually negative and was also classified as negative which is the right thing to do.
False Negative: - The actuals were actually positive but was falsely classified as negative.

Model interpretations

- The high accuracy score of 1.0 on the training set and 0.9824 on the test set indicate that the tuned gradient boosting model is performing very well in predicting customer churn, and it is not overfitting.
- The recall score of 1.0 on the training set and 0.9247 on the test set indicates that the model is able to identify most of the customers who are likely to churn, and only a small percentage of those who are not likely to churn will be identified as churners.
- The high precision score of 1.0 on the training set and 0.9699 on the test set indicates that the model is able to accurately predict the customers who are likely to churn, and only a small percentage of those who are not likely to churn will be identified as churners.
- The high f1-score of 1.0 on the training set and 0.9467 on the test set suggests that the model is able to balance the trade-off between precision and recall.
- The AUC score of 1.0 on the training set and 0.996 on the test set suggest that the model has a high ability to distinguish between positive and negative classes, which indicates a good discrimination power of the model.

Business implications

- The high performance metrics of the tuned gradient boosting model for customer churn analysis suggest that the model is very effective in predicting which customers are likely to churn.
- The high accuracy score and AUC score indicate that the model has a high ability to distinguish between positive and negative classes, and it is not overfitting.
- The high recall and precision scores suggest that the model is able to identify most of the customers who are likely to churn and accurately predict the customers who are likely to churn.
- The business implications of this analysis are that the company can use this model to identify the customers who are at risk of churning and take appropriate actions to retain them.
- For example, the company can offer special promotions or discounts to retain these customers or improve their customer service experience to reduce the likelihood of churning.
- By taking these actions, the company can reduce customer churn and improve customer retention, which can ultimately lead to increased revenue and profitability.
- Additionally, the company can use this model to optimize their marketing and sales efforts by targeting customers who are more likely to churn with specific marketing campaigns or sales incentives.

6. Final interpretation / recommendation

Insights from analysis:

- **City tier type 1** shows the **highest rate of churn** compared to city tier types 2 and 3
- Customers who prefer **debit cards or credit cards** as their mode of payment are more likely to churn
- **Male customers** have a higher churn rate compared to female customers
- Customers in the "**Regular Plus**" segment are showing a higher churn rate
- **Single customers** are more prone to churning compared to divorced or married customers
- Customers who use the service over a **mobile device** show a higher rate of churn
- Any **complaints raised in last 12 months** doesn't show any impact toward churn
- **Tenure** and **cashback** are directly proportional to each other

Insights from model building:

- We have 2 scenarios regarding metrics. In the 1st scenario, the model predicts incorrectly that some customers will not churn when they actually will.
- To address this, we focus on recall and aim to minimize false negatives as they can cause a significant loss. In the 2nd scenario, the model predicts incorrectly that some customers will churn when they will not.
- This can lead the company to spend more money than necessary and this may unintentionally reduce profits. In this case, precision is crucial and we strive to minimize false positive.
- Therefore, our priority is to minimize false negative, which means recall takes precedence over precision.
- So, compared to all the models, the only model with high recall parameter is **tuned Gradient Boosting model.**

Recommendations:

- Increase visibility in **tier 2 cities** to acquire new customers
- Promote hassle-free and secure payment methods like standing instructions in **bank accounts or UPI**
- **Conduct a survey** to better understand customer expectations and improve service scores
- **Train customer care executives** to deliver **better service** and provide them with necessary tools
- Offer **customized plans** based on customer spend and tenure to improve customer loyalty and retention
- Offer **family floater plans** for married customers to increase customer satisfaction and loyalty
- Tailor **services and marketing efforts** to meet the needs and preferences of target audience to improve **customer satisfaction and drive business growth**
- Analyse customer churning pattern based on **gender and address** the reasons why male customers are more likely to churn
- Focus on **improving customer experience** for the "Regular Plus" segment, as they have shown a higher churn rate
- Offer tailored packages with **added benefits or discounts** to reduce churn rates among single customers
- Conduct a detailed study of mobile user preferences and **address pain points** to reduce churn rates among mobile users
- Develop **loyalty programs or rewards** for repeat business to increase customer retention and loyalty
- **Offer financial incentives or flexible payment** options to reduce churn among customers who prefer debit or credit cards

- **Invest in technology** to streamline customer service processes and provide faster and more **efficient service to improve customer satisfaction** and reduce churn
- One potential method of categorizing customers involves **dividing them into four groups** based on their **spending habits and loyalty**.
  - The first group consists of **high-spending customers who have been loyal** to the company for a long time. These customers may receive exclusive rewards and benefits.
  - The second group consists of **high-spending customers who have not been loyal** to the company for a long time. They may need additional incentives to continue shopping with the company.
  - The third group consists of **low-spending customers who have been loyal** to the company for a long time. These customers may appreciate rewards and incentives to encourage them to spend more.
  - The fourth group consists of **low-spending customers who have not been loyal** to the company for a long time. These customers may require more attention and engagement to maintain their interest in the company.

APPENDIX:

```
import numpy as np
import pandas as pd
from pandas import datetime
from datetime import datetime

# data visualization
import matplotlib.pyplot as plt
import matplotlib.style
plt.style.use('seaborn')
from pylab import rcParams
from sklearn import metrics
from   sklearn.metrics import  mean_squared_error
from sklearn import metrics
from sklearn.metrics import roc_auc_score,roc_curve,classification_report,confusion_matrix,plot_confusion_matrix
import seaborn as sns # advanced vizs
%matplotlib inline

sns.set_style('darkgrid')
pd.set_option('display.max_rows', 500)
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 2000)
pd.set_option('display.expand_frame_repr', False)
import warnings
warnings.filterwarnings('ignore')
```

Hyperparameters used on Gradient boosting model:

```
gbc_init = GradientBoostingClassifier(init=AdaBoostClassifier(random_state=1),random_state=1)
gbc_init.fit(X_train,y_train)
```

```
#Using above defined function to get accuracy, recall and precision on train and test set
gbc_init_score=get_metrics_score(gbc_init)
```

```
# Choose the type of classifier.
gbc_tuned = GradientBoostingClassifier(criterion='mse', learning_rate=0.5, max_depth=9,max_features=11,
                                       min_samples_leaf=6,min_samples_split=15, n_estimators=201,random_state=0)
# Grid of parameters to choose from
## add from article
param_grid = {
 'loss': ['deviance'],
 'learning_rate': [0.1, 0.5, 1],
 'n_estimators': [201],
 'criterion': ['mse'],
 'min_samples_split': [15],
 'min_samples_leaf': [6],
 'max_depth':[9],
 'max_features':[11]
}
#
# Type of scoring used to compare parameter combinations
acc_scorer = metrics.make_scorer(metrics.recall_score)

# Run the grid search
grid_obj = GridSearchCV(gbc_tuned,param_grid,scoring=acc_scorer,cv=5)
grid_obj = grid_obj.fit(X_train, y_train)

# Set the clf to the best combination of parameters
gbc_tuned = grid_obj.best_estimator_

# Fit the best algorithm to the data.
gbc_tuned.fit(X_train, y_train)
```

```
#Using above defined function to get accuracy, recall and precision on train and test set
gbc_tuned_score=get_metrics_score(gbc_tuned)
```

```
make_confusion_matrix(gbc_tuned,y_test)
```

```
## Confusion matrix on the training data
plot_confusion_matrix(gbc_tuned,X_train,y_train)
plt.grid(False)
```

```
# predict probabilities
probs = gbc_tuned.predict_proba(X_train)
# keep probabilities for the positive outcome only
probs = probs[:, 1]
# calculate AUC
lr_train_auc = roc_auc_score(y_train, probs)
print('AUC: %.3f' % lr_train_auc)
# calculate roc curve
train_fpr, train_tpr, train_thresholds = roc_curve(y_train, probs)
plt.plot([0, 1], [0, 1], linestyle='--')
# plot the roc curve for the model
plt.plot(train_fpr, train_tpr);
```

```
# predict probabilities
probs = gbc_tuned.predict_proba(X_test)
# keep probabilities for the positive outcome only
probs = probs[:, 1]
# calculate AUC
lr_test_auc = roc_auc_score(y_test, probs)
print('AUC: %.3f' % lr_test_auc)
# calculate roc curve
test_fpr, test_tpr, test_thresholds = roc_curve(y_test, probs)
plt.plot([0, 1], [0, 1], linestyle='--')
# plot the roc curve for the model
plt.plot(test_fpr, test_tpr);
```

```
important_features = pd.DataFrame({'Features': X_train.columns,
                                   'Importance': gbc_tuned.feature_importances_})

important_features = important_features.sort_values('Importance', ascending = False)

sns.barplot(x = 'Importance', y = 'Features', data = important_features)

plt.title('Feature Importance', fontsize = 15)
plt.xlabel('Importance', fontsize = 15)
plt.ylabel('Features', fontsize = 15)

plt.show()
```