Great Learning

Final Report Submission_Capstone

Customer Churn Prediction

Submitted by Sachin Kose Paul

Contents

1. Introduction of the business problem

An E-commerce company is facing a lot of challenges in the market. It has become a challenge for the company to retain the existing customers in the current market. The company wants to develop a model that can predict the account churns and provide segmented offers to the potential churners. Customer churn basically refers to the customer/subscriber refraining from further business activities with the company. Account churn poses a serious challenge as a single account can have multiple customers and losing an account will result in losing more than one customer. Thus, the challenge is to develop a churn prediction model that provides unique and clear cut business recommendations for the company campaign such that the loss incurred by the company is minimal.

The need of this project/study has to do with the minimising of churn by making use of machine learning and exploratory data analysis(EDA). Moreover, machine learning and EDA can also be used to provide business suggestions and recommendations for the company by making use of the churn model.

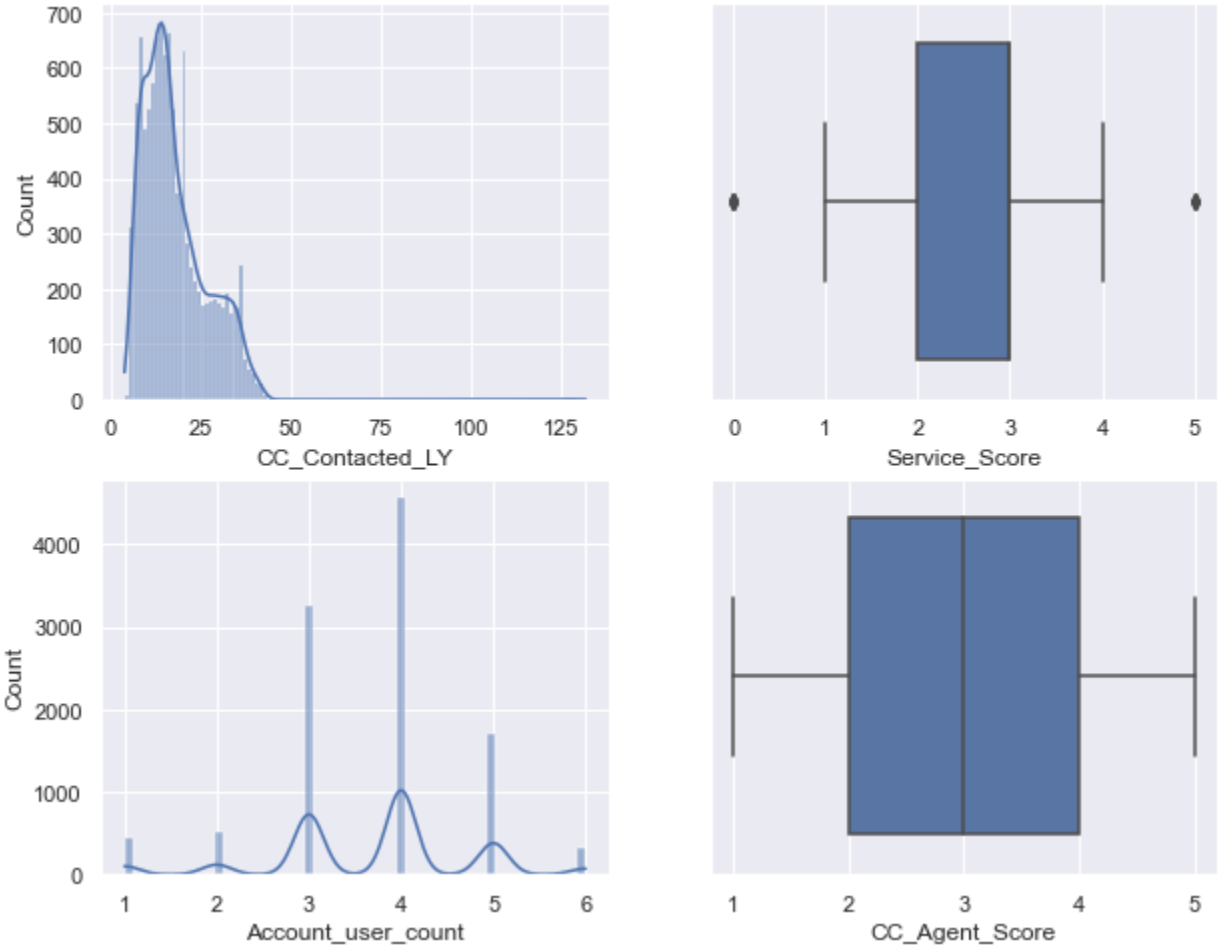
Reducing customer churn from a business perspective is nothing but the action of increasing the company profits. Customer churn is inversely proportional to customer retention. Decrease in customer churn means increase in customer retention and thus causing a resultant increase in the company profit. From a social point of view this means the opportunity to increase the number of customers/subscribers associated with the company or the product.

2. EDA and Business Implication

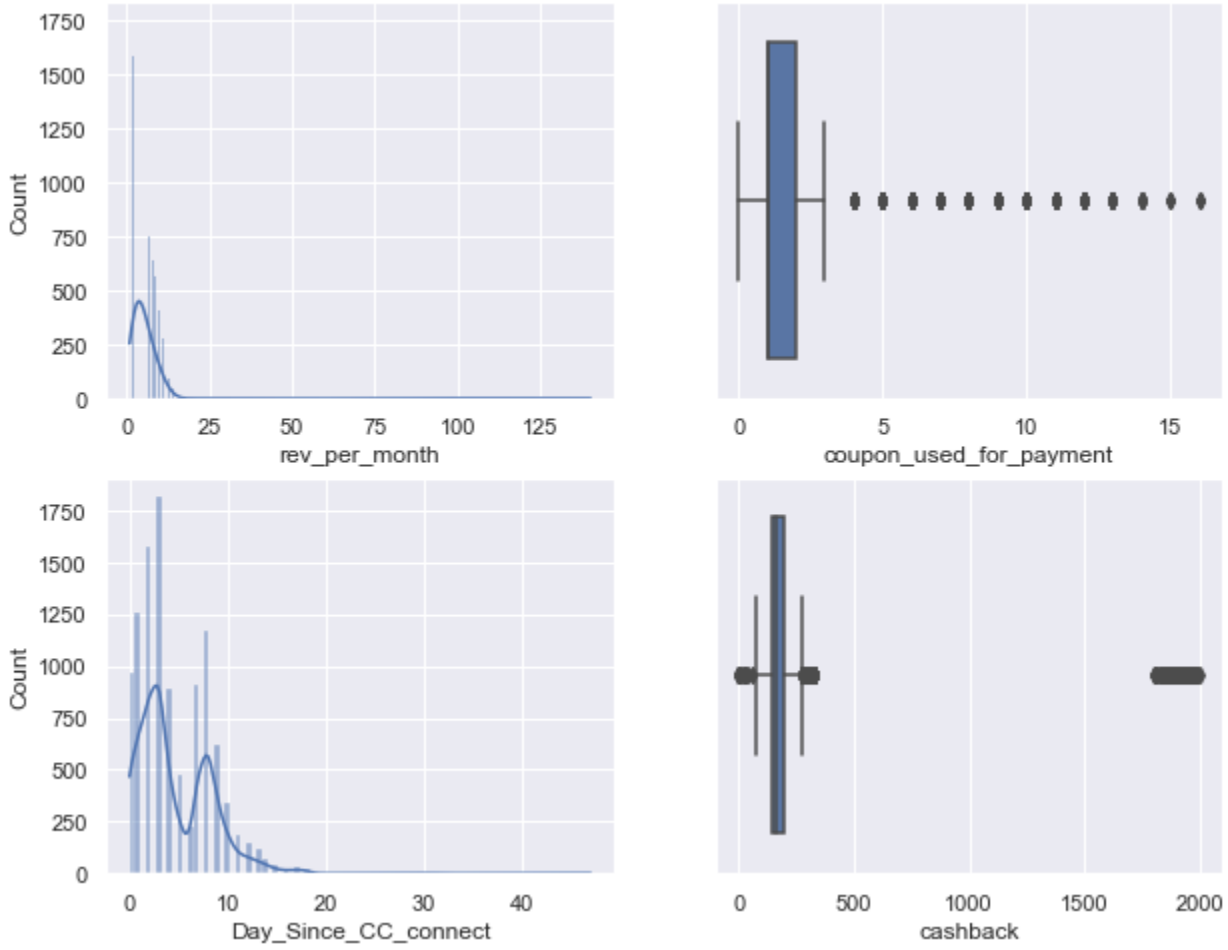
The describe function provides a brief summary of the entire dataset. A few pointers from the same are mentioned below.
The most widely used mode of payment is debit card.
The majority of the subscribers are male and they are married as well.

The most widely used account_segment is Super and most of the logins happen through mobile phones.
In the last 12 months, the customers had contacted the customer care service for an average of 18 times.
The mean Agent score is higher than the mean service score.
The remaining attributes are having erroneous values in the dataset due to which incomplete information has been displayed in the data summary.

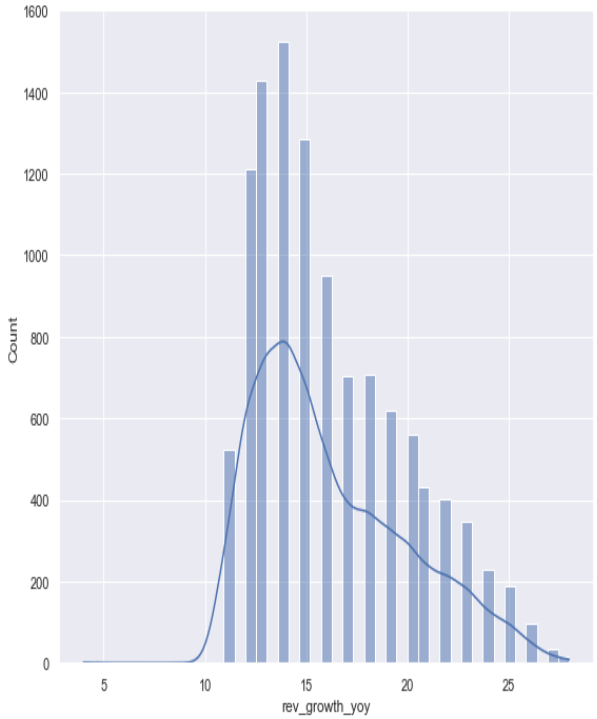
The dataset could be worked upon for Univariate, Bivariate and Multivariate Analysis.

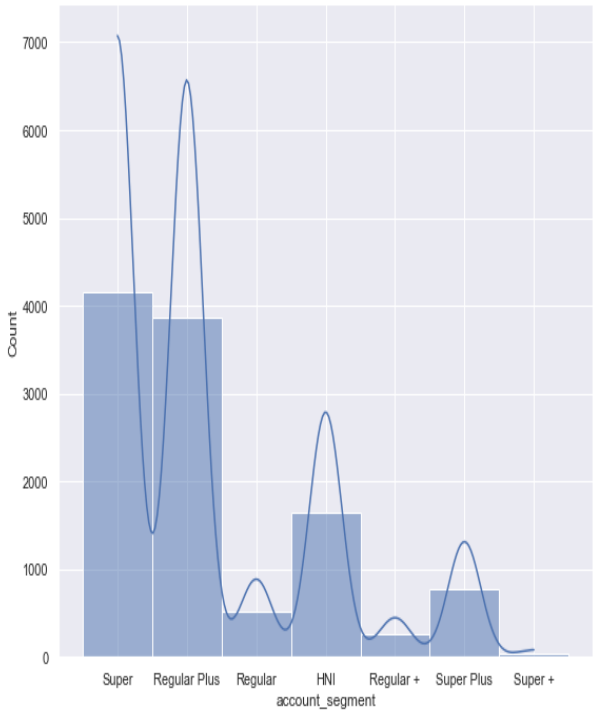


CC_Contacted_LY shows that almost 75 percentile of the customers have contacted the customer care for around 25 times in the last 12 months. This can be treated as a hint to understand that there is a trend of dissatisfaction amongst the majority of the customers. Account_user_count depicts a maximum of 4 customers per account. This gives even more of a reason to minimise the customer churn.



coupon_used_for_payment and cashback has outliers present in them. rev_per_month which is revenue generation per month has maximum customers below 5 which is not an ideal score.

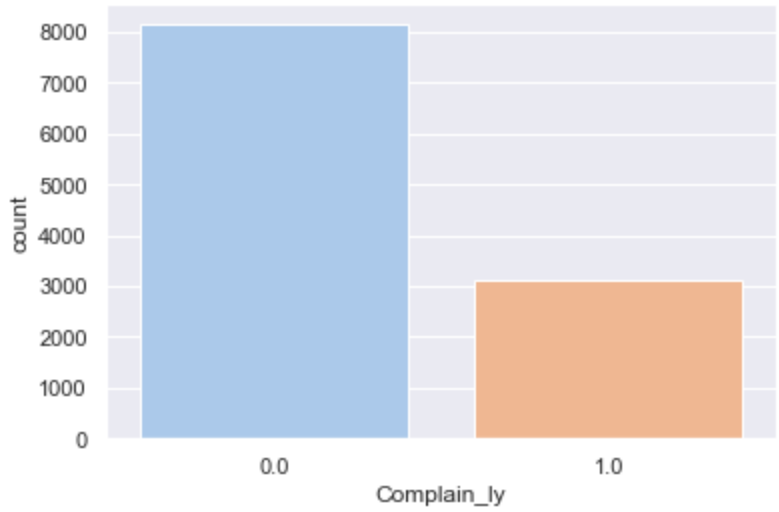




rev_growth_yoy(Percentage revenue growth) for the majority of the customers lie between 10 and 15. The account_segment shows that majority of the customers are satisfied with Super and Regular Plus programmes.

The past data shows that around 17 percent of the total customers were churned with the existing services provided by the company.

City_Tier 2.0 has a drastically smaller contribution to the customers as compared with 1.0 and 3.0

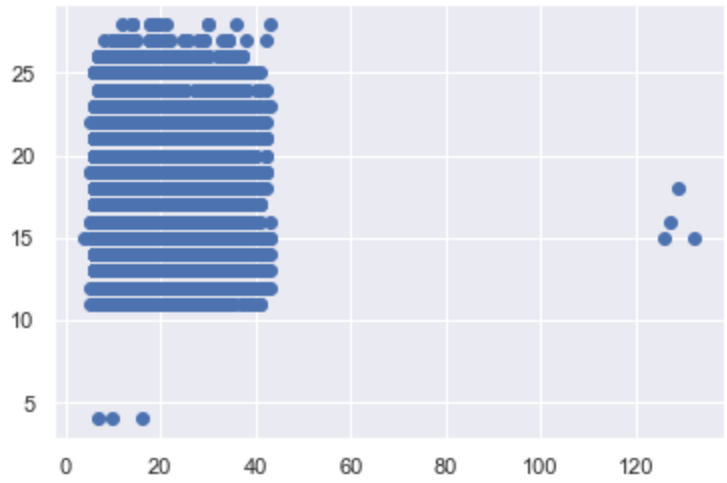


Around 26 percent of the customers have raised complaints with regards to the services provided in the last 12 months

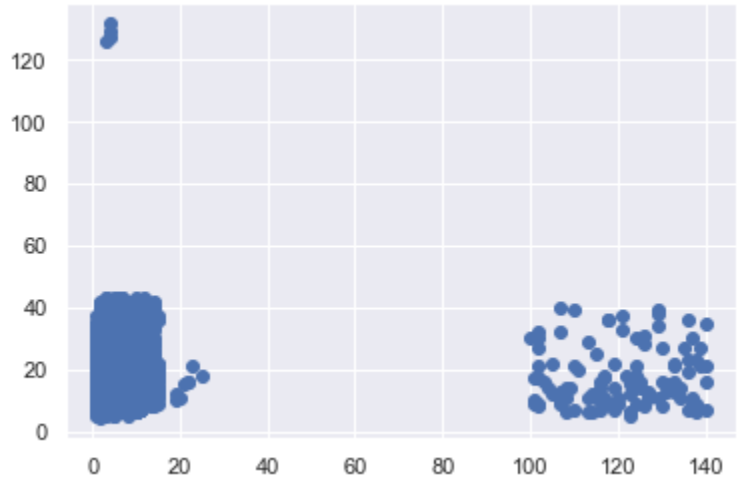
The other attribute “Payment” shows that debit cards make the major contribution. The company has male customers when compared with their female customers. Married couples make the majority of the crowd of customers. And the mostly used login device is mobile phones.

2.2. Bivariate Analysis

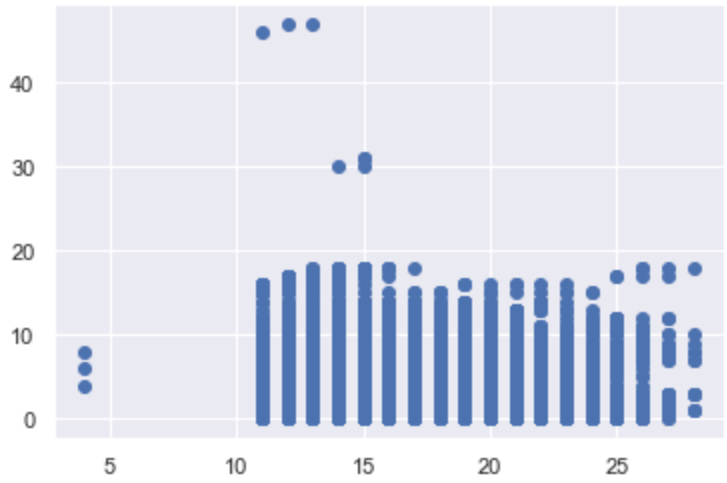
Numeric vs Numeric



rev_growth_yoy is the densely distributed towards the first 40 of CC_Contacted_LY. This means that subscribers who have contacted the customer care for more than 40 days in the last 12 months are having scanty revenue growth percentage accounts.

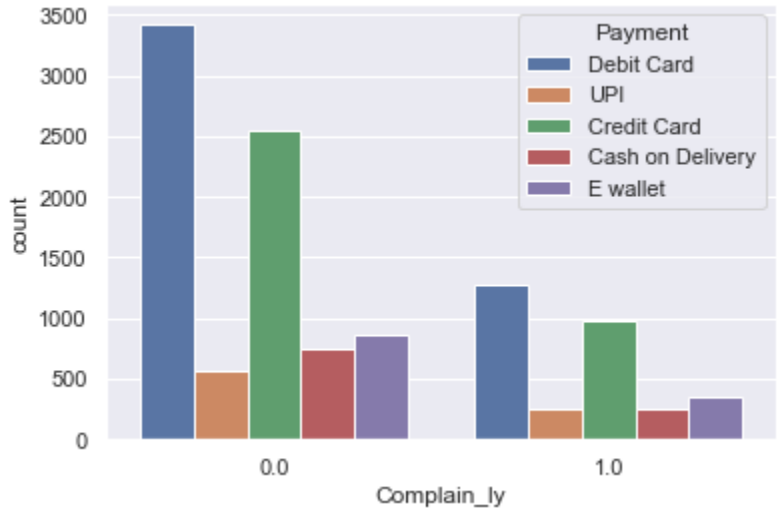


rev_per_month vs CC_Contacted_LY also proves the same fact as above.

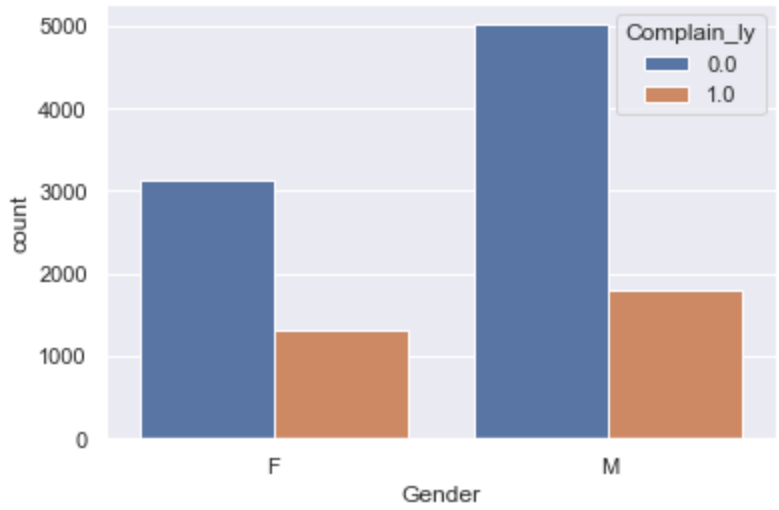


This is rev_growth_yoy vs Day_Since_CC_connect
The percentage revenue growth of the accounts of the subscribers whose last contact with the customer care was between 15 to 25 days are more compared with the rest. These customers are to be kept satisfied to keep them off from churn.

Categorical vs Categorical

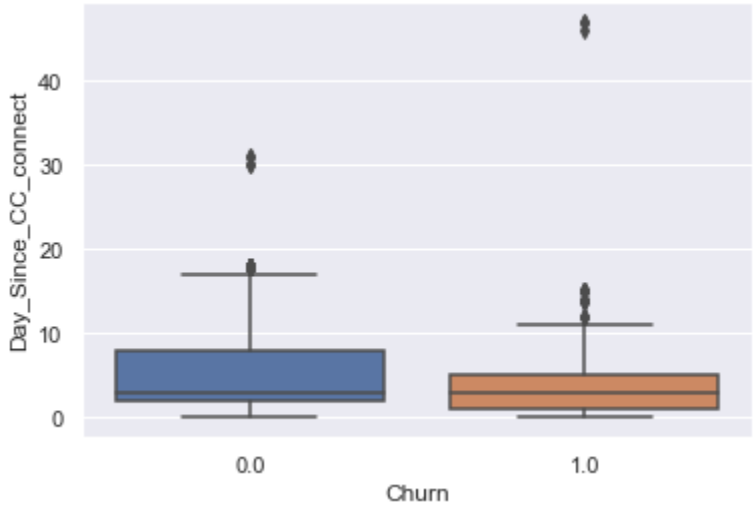


Subscribers using debit cards and credit cards complained the most.



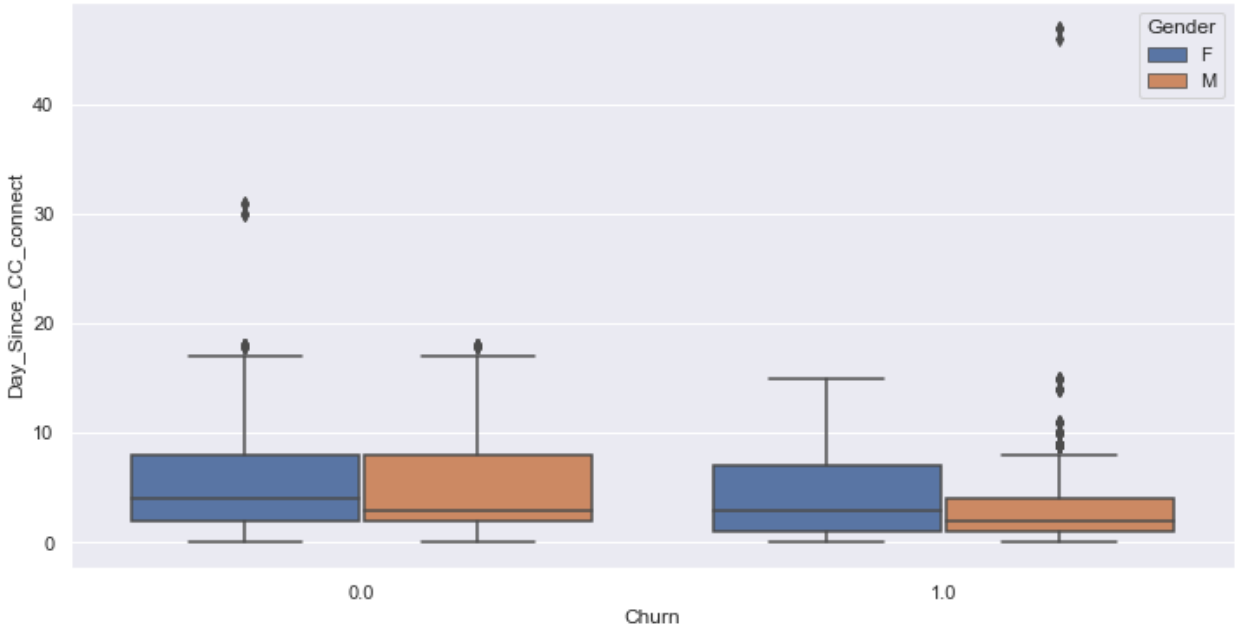
The proportion of the complaints from the female customers are more than the male customers.

Categorical vs Numeric



This graph shows that subscribers whose last connect date with the customer care was 2 weeks ago face the threat of being churned out.
A check has to be placed on all the subscribers
whose last connect date is beyond 10 days.

2.3. Multivariate Analysis



This graph shows that female subscribers who last contact with customer care was within 15 days are more likely to be churned compared to their male counterparts.

Outliers are present in coupon_used_for_payment and cashback. Since treating the outlier values here may lead to loss of information as a high number of observations in these variables are outliers we prefer not to do the outlier treatment.

3. Data Cleaning and Preprocessing

The attribute AccountID holds no significance in the churn prediction. Hence, it can be omitted from the dataset using the command “drop”. The remaining 18 attributes don't look removable and can be used for further evaluation.

3.1. Replacing Typos and other unwanted symbols

The data type of the attribute can be figured out from the function “info”.

This led to the discovery that even numeric attributes have “object” as their data type. This demanded a quality check of all the attributes for unwanted symbols or typo errors and an attempt on variable transformation has to be made.

Using the “unique” function on all the attributes, a conclusion was arrived upon as in which attributes had unwanted symbols or typo errors.
Unwanted symbols like “#”,”$”,”@”,”+” were present in the variables.
Making use of the replace function, all the unwanted symbols were replaced with null values.

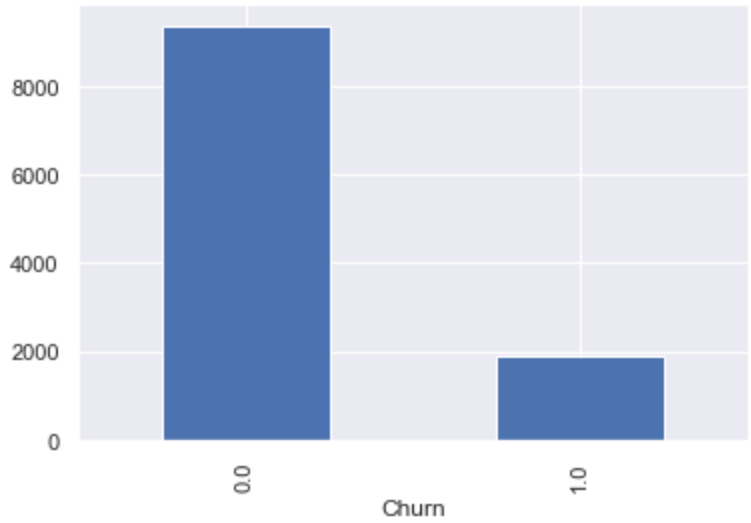
Moreover, the attribute “Gender” had values Female,F,Male,M. All the values in Gender were changed to F and M using the replace function.
The next task was to impute the missing values with mean/median/mode.

3.2. Imputation

The entire dataset was divided into a numeric dataset and a categorical dataset using the function select_dtypes.
Making use of the SimpleImputer imported from sklearn.impute all the missing values in the numeric dataset was imputed with median and all the missing values in the categorical dataset was imputed with mode.
The datasets are then converted back to dataframes and then concatenated using concat function.

3.3. SMOTE

There is class imbalance in the dataset. A simple example will the distribution of the contents of “Churn”



The predictions in the dataset with class imbalance cannot be relied upon. Hence, we have to implement SMOTE inorder to preprocess the dataset.

SMOTE will ensure that the proportion of false negatives will be lesser than false positives. From the business perspective, false positives can be handled at the later stage of the business pipeline compared to false negatives. Eg : Considering Corona, a false positive patient is less troublesome compared to false negative patient.

Also, since there was class imbalance in the dataset, during test-train split the function Stratify is used to ensure jusdistic distribution. This will ensure that the class imbalance present in the original dataset is replicated accordingly in the training and the testing set.

4. Model building

4.1. DECISION TREE

Decision tree is a supervised learning technique. Just like the trees they have nodes - Root node, Decision node and terminal node.

Initially, we have to ensure that all the variables have to be of the data type int or float, as decision tree variables cannot be of object data type. The

functions “Categorical” and “codes” are used for their conversion to int or float.

If the number of the independent variables are more than one, using Gini computation, the variable with the most importance is figured out. This variable will act as the splitting criterion for the root node.

Since the resultant decision tree is heavily branched, we pruned them.

Let us evaluate the confusion matrix for training data

[5998, 557],
[833, 5722]

No of True positives and True negatives = 5998+5722
False positives = 833
False negatives = 557

False positives > False negatives. This shows that the model built is efficient.

Let us evaluate the confusion matrix for testing data

[2542, 267],
[160, 409]

No of True positives and True negatives = 2542+409
False positives = 160
False negatives = 267

False positives < False negatives

4.2. Random Forest Classifier

Now we will build a Random Forest model. Random Forest model is an ensemble technique that relies on the wisdom of the crowd. Numerous base models will be built. These base models are not correlated to each other. They are weak classifiers by themselves. Each of these base classifiers use a different set of variables for prediction. Ultimately the predictions of the base classifiers are combined to generate one single prediction.

Bootstrapped datasets are built from the original dataset where each of these bootstrapped datasets are used to build a unique classifier. Bootstrapped datasets are generated by using "random sampling with replacement" so as to ensure that the rows in every dataset are unique. In this dataset there are 19 features. So, while building a bootstrapped datasets here, apart from implementing "random sampling with replacement" in the rows, columns are also randomly selected for each decision tree and using Gini computations the best variable among the randomly selected variables is determined for the splitting criterion

The code oob_score_ gives the value of out of bag scores.
oob_score _ here is 0.9837, ie. ~98%. This also means that the error rate is 2%.

Modeling techniques have been used here to improve the performance metrics.

Let us evaluate the confusion matrix for training data

[6029, 526],
[713, 5842]

No of True positives and True negatives = 6029+5842
False positives = 713
False negatives = 526

False positives > False negatives
Let us evaluate the confusion matrix for testing data

[2567, 242],
[118, 451]

No of True positives and True negatives = 2567+451
False positives = 118
False negatives = 242

False positives < False negatives

4.3. Logistic Regression

Let us evaluate the confusion matrix for training data

[4952, 1603],
[1308, 5247]

No of True positives and True negatives = 4592+5247
False positives = 1308
False negatives = 1603

False positives < False negatives
Let us evaluate the confusion matrix for testing data

[2134, 675],
[130, 439]

No of True positives and True negatives = 2134+439
False positives = 130
False negatives = 675

False positives < False negatives

5. Model Tuning

GridSearchCV

Ensemble modeling has been used for the Random Forest.
We have used GridSearch with Cross validation.

The performance metric before modeling was giving out poor results and thus we had to tune the model so as to obtain higher recall and precision.

We have specified 5 parameters for the Gridsearch. They are as follows :

max_depth
max_features
min_samples_leaf
min_samples_split
n_estimators

n_estimators ask us how many trees we need to build the random forest. max_features specifies the number of features that needs to be considered before Gini computation. min_samples_leaf is the minimum number of values that a terminal node should have. Min_samples_split is the minimum number of values that a parent node should have so as to split.

The best set of values that I have used for model building are :

max_depth: [10, 15],
max_features: [5, 6],
min_samples_leaf: [50, 100],
min_samples_split: [150, 300],
n_estimators: [201, 301]

With respect to the data provided, the Gridsearch will compute trees for 2*2*2*2*2 = 32 times.

Also, since we have included cross validation here, the value assigned is cv=3, the execution happens 32*3 = 96 times.

After the algorithm is executed 96 times, the best model of all the 96 is retained by the algorithm. The best parameters that the optimum model possess can be obtained using the function grid_search.best_params_

The best parameters thus obtained are as follows :

{'max_depth': 15,
'max_features': 5,
'min_samples_leaf': 50,
'min_samples_split': 150,
'n_estimators': 301}

Hence, with GridSearchCV we were able to model the Random Forest and obtain the optimum model.

Decision Tree Pruning

The Decision Tree Classifier had developed a heavily branched tree. This tree had to be trimmed via pruning before we could arrive at the results.

The parameters we chose for pruning the decision tree are as follows:

criterion,
max_depth
min_samples_leaf,
min_samples_split

Criterion refers to the computation algorithm that should be used so as to identify the most important feature. max_depth refers to the number of branches the tree should have. min_samples_leaf is the minimum number

of values that a terminal node should have. Min_samples_split is the minimum number of values that a parent node should have so as to split.

The values we have assigned to the parameters are as follows:

criterion = 'gini',
max_depth = 7,
min_samples_leaf=15,
min_samples_split=50

The decision tree thus we modeled ended up giving us excellent recall values that is the most essential metric in this business problem.

5) Model Validation

The performance metrics for Logistic Regression, Decision Tree and Random Forest before tuning and after tuning are show below for comparison

Before Tuning

	Training Dataset						Testing Dataset					
	DT		RF		Log R		DT		RF		Log R	
	0	1	0	1	0	1	0	1	0	1	0	1
precision	0.79	0.77	0.87	0.92	0.82	0.77	0.96	0.78	0.98	0.91	0.94	0.39
recall	0.76	0.8	0.92	0.87	0.75	0.83	0.95	0.81	0.98	0.89	0.75	0.78
f1-score	0.77	0.78	0.9	0.89	0.78	0.8	0.96	0.8	0.98	0.9	0.84	0.52

After Tuning

	Training Dataset						Testing Dataset					
	DT		RF		Log R		DT		RF		Log R	
	0	1	0	1	0	1	0	1	0	1	0	1
precision	0.87	0.92	0.9	0.92	0.81	0.77	0.95	0.63	0.95	0.65	0.94	0.4
recall	0.92	0.87	0.92	0.89	0.76	0.82	0.91	0.75	0.91	0.79	0.76	0.77
f1-score	0.9	0.89	0.91	0.9	0.78	0.8	0.93	0.68	0.93	0.71	0.84	0.52

Decision Tree Evaluation

Using the score function, accuracy is obtained, which is 0.77 for train and 0.76 for test variables before pruning.
After pruning, the accuracy of the training data rose upto 0.89 and the accuracy of the testing data rose up to 0.87 which means pruning has improved the efficiency of the system

Recall is inversely proportional to the type 2 error.

Higher recall would mean lesser false negatives. Reducing the number of false negatives is the performance metric of highest value to this data problem. As false negatives would mean customers who could be churned but are falsely identified as “cannot be churned”. This is something which we want to avoid in the business problem. **Hence, reducing false negatives and thus increasing recall is the primary objective while evaluating performance metrics.**

Precision is inversely proportional to Type 1 error. For the training data precision for 1.0 is higher than that of 0.0. But, it's the opposite for testing data.

In a business problem, if we need to reduce Type 1 and Type 2 errors, we have to improve the F1 score. F1 score is the harmonic mean of Recall and Precision. F1 score has improved for 0.0 in the testing data compared to the training data, but it’s value for 1.0 has dipped in the testing data.

If the values of Recall for 0.0 and 1.0 in the testing data are within the threshold of 10%, then the model is underfitting in training and overfitting in testing. We have to ensure in every performance metric that they are within the 10% threshold. Here, for 0.0 it is within 10% but for 1.0 it's a little over 10%.

Using the code model.feature_importances_ we can figure out which of the features of the datasets have been crucial in the creation of the Decision Tree model. Here, 2 of the most important features are "Tenure" and "Complain_ly".

Random Forest Evaluation

Accuracy for the training data after modeling techniques is
0.9054919908466820
Accuracy for testing data after modeling techniques is
0.8934280639431617
Here, the precision and the recall for 0.0 and 1.0 are within the 10% threshold which means the model is fitting perfectly.

Using the code model.feature_importances_ we can figure out which of the features of the datasets have been crucial in the creation of the Random Forest model. Here, 2 of the most important features are "Tenure" and "Complain_ly"

Logistic Regression Evaluation

Accuracy for the training data is 0.7779557589626239

Accuracy for testing data is 0.7616933096506808

Here,the recall for 0.0 and 1.0 are within the 10% threshold which means the model is fitting perfectly.

The most optimum model out of all the three models that I have built is the Random Forest Classifier. The reason why I arrived at such a conclusion is because, as far as this business problem is concerned, Recall is the metric with maximum business implications. **Recall is the most important metric** because higher the recall value, lower is the type 2 error. Lower the type 2 error the more efficient is this business model.

The recall value(testing data) of 0.0 is 0.91 and for 1.0 it is 0.79. This is the highest in all the three models for the testing data. On top of this, recall and precision values for Random Forest falls within the 10% training-testing threshold. This ensures that the Random Forest is correctly fitted.F1 score of the testing data of Random Forest is the highest among all the models. F1 score of 0.0 is 0.93 and for 1.0 is 0.71. Higher F1 score means higher Recall and Precision, which also means lesser Type1 and Type 2 errors.

Also, the accuracy and precision of Random Forest is the highest, when compared to all other models.

6. Final interpretation / recommendation

The most important features of the dataset have been identified and these features will play a pivotal role in the prediction of the model. The two most important features of the model are “Tenure” and “Complain_ly”. The customers with higher “Tenure” have a very low probability of being churned. Customers who have raised at least 1 complaint in the last 12 months have more probability of being churned.

The recommendation that can be provided after building this model are as follows:

1.Focus all the marketing activities on customers with low tenure values and the customers who have raised at least 1 complaint in the last 12 months. It is these customers that have more probability of being churned.

Hence, more effort has to be made on these customers so as to lower the customer churn rate of the company.

2. Retain the agents of the company. The mean Agent score is higher than the mean service score. The customers have more trust in agents than in the company. These agents form the backbone for customer retention

3. Almost 75 percentile of the customers have contacted the customer care for around 25 times in the last 12 months. This is a hint of dissatisfaction amongst the majority of the customers.

4. City_Tier 2.0 has a drastically smaller contribution to the customers as compared with 1.0 and 3.0. Marketing activities has to be focussed more on City_Tier 2.0

5. Around 75 percentile of the females for whom the number of days since the last customer care connect was within 8 days has higher probability of being churned. All such female customers have to be kept happy via offers or targeted marketing campaigns.

The End_ _ _