BLOG/ARTICLE

TELECOM CUSTOMER CHURN ANALYSIS

Submitted by : Hari Krishnan J

Batch : DS0522

PROBLEM DEFINITION:

Customer churn rate is one of the important metrics for companies to evaluate their performance. Customer churn rate is a KPI to understand the leaving customers. Churn rate represents the percentage of customers that company lost over all the customers at the beginning of the interval.

Customer retention can be achieved with good customer service and products. But the most effective way for a company to prevent attrition of customers is to truly know them. The vast volumes of data collected about customers can be used to build churn prediction models. Knowing who is most likely to defect means that a company can prioritise focused marketing efforts on that subset of their customer base.

Preventing customer churn is critically important to the telecommunications sector, as the barriers to entry for switching services are so low.

Here, we have customer data from IBM Sample Data Sets with the aim of building and comparing several customer churn prediction models.

DATA ANALYSIS:

The dataset have 7043 records with 21 features.

Independent Features:

- Gender — Male / Female
- Age range — In terms of Partner, Dependent and Senior Citizen
- Phone service — If customer has Phone service, then services related to Phone like Multi-line Phone service
- Internet Service — If customer has Internet service, then services related to Internet like Online security, Online backup, Device protection, Tech support, Streaming TV, Streaming Movie
- Tenure — How long customer is with the company?

- Contract type — What kind of contract they have with a company? Like Monthly bases, On going bases — If on going bases, then One month contract or Two year contract
- Paperless billing — Customer is paperless billion option or not?
- Payment method — What kind of payment method customer has? Mailed check, Electronic check, Credit card (Automatic), Bank transfer (Automatic)
- Monthly charges
- Total charges

Dependent Label(Target):

- Churn — Whether customer left the company or still with the company?

EXPLORATORT DATA ANALYSIS:

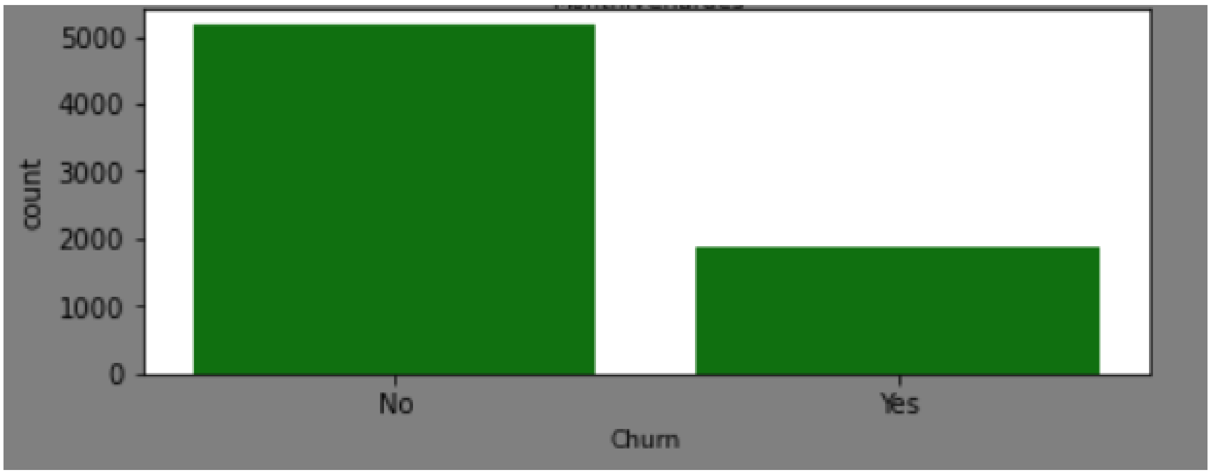
1. Data Types:

   The dataset contains Object, Int and float data types.

   This dataset had both categorical and Continuous features.

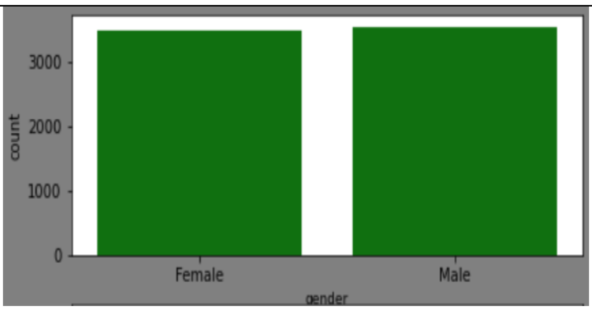
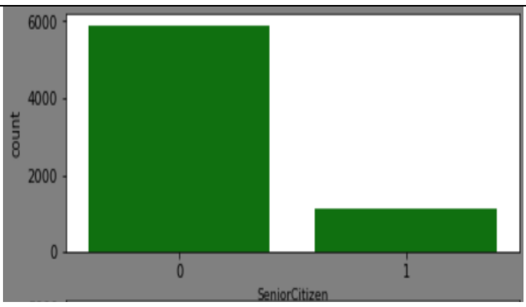
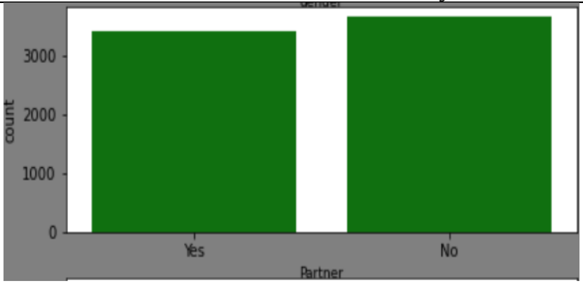
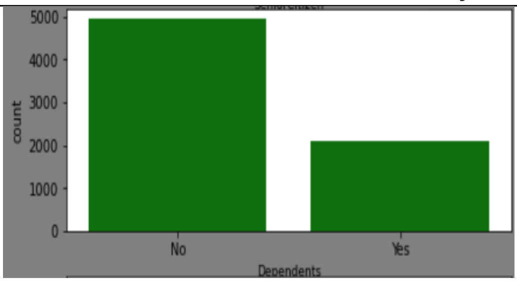
2. Null Values: No null values exist in this dataset. No need to treat.

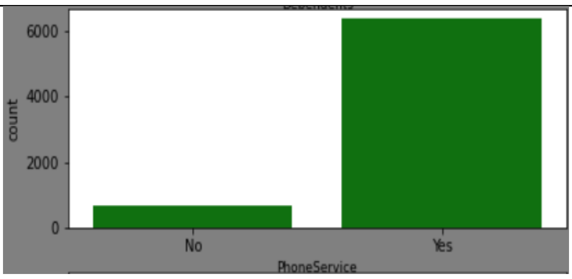
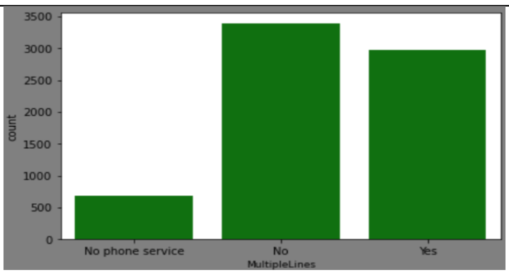
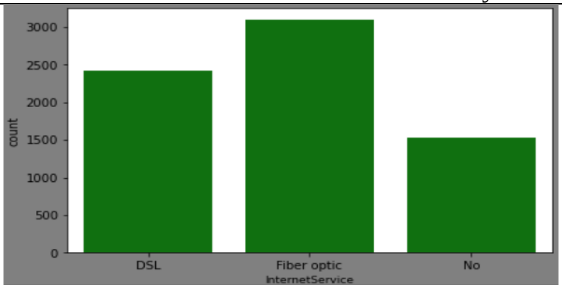
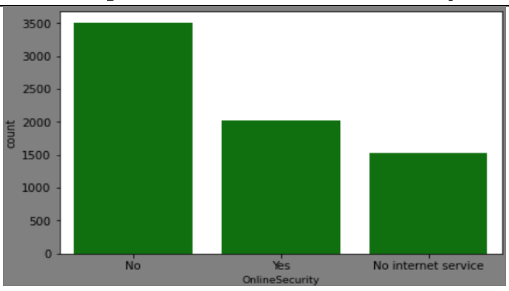
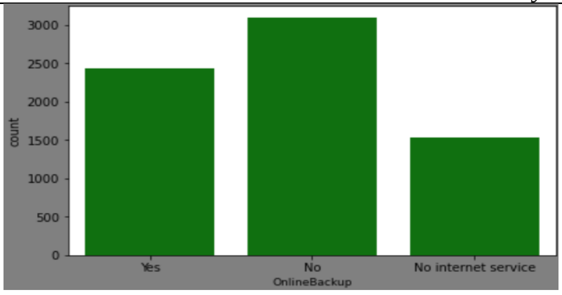
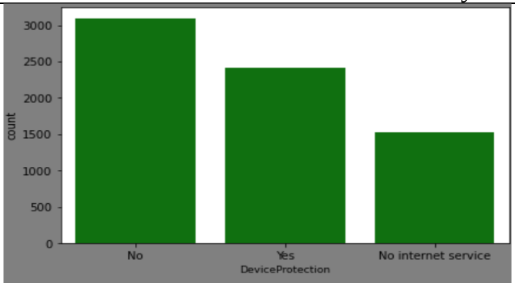
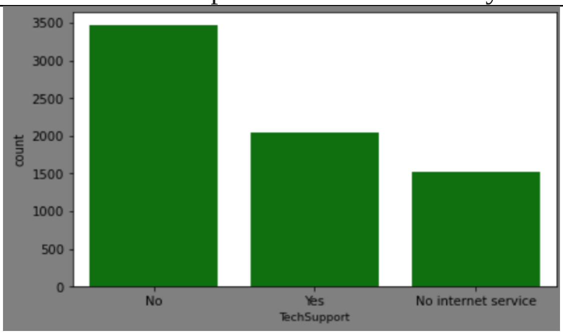
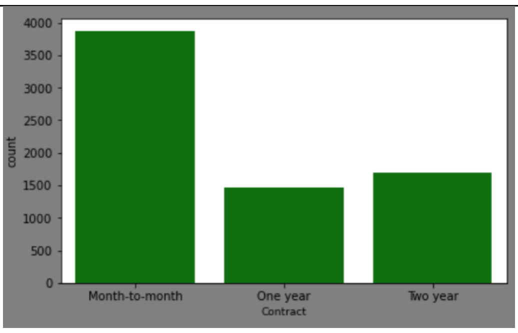
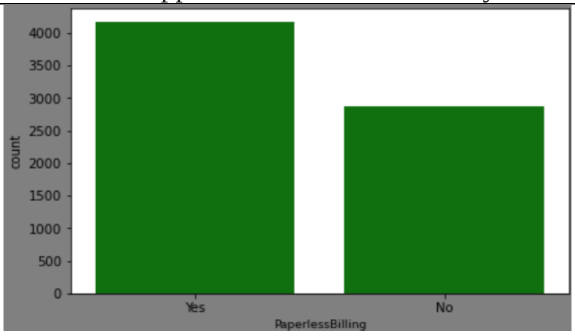
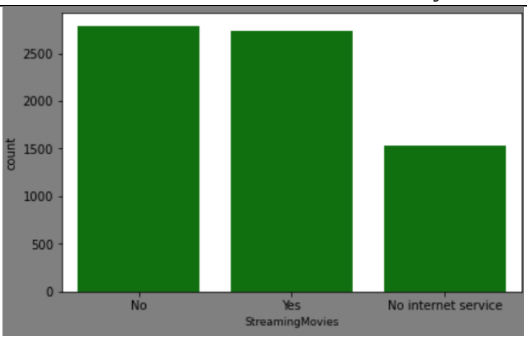
3. Customer Churn Distribution:



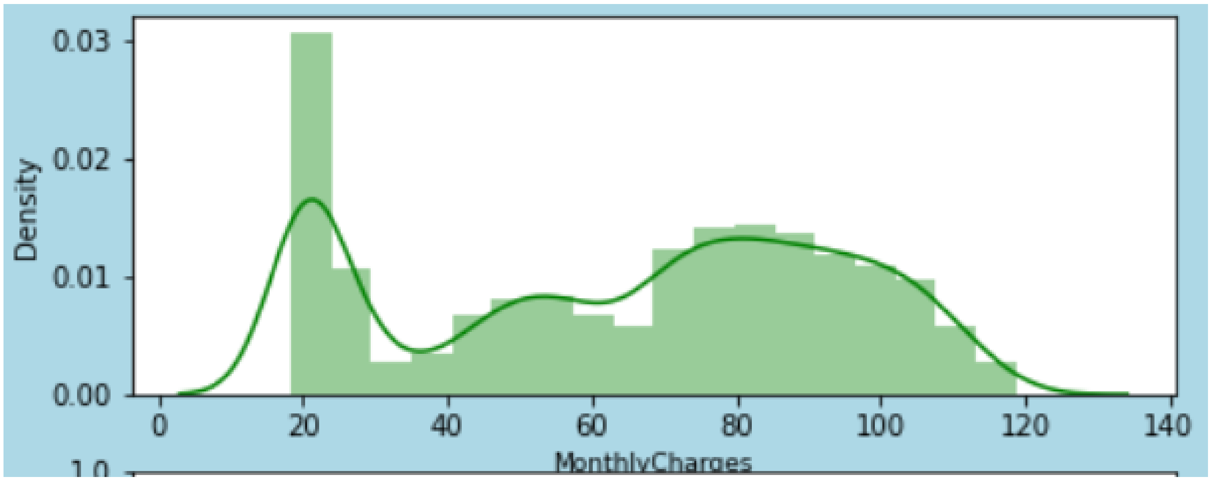
The churn data in dataset is not balanced. We need to Balance the dataset.

4. 'Customer ID' feature have all unique Values. So it does not have impact on target label. So we can drop this.
5. 1526 rows which does not have internet have same pattern. But there is mixed result in terms of churn. So we can treat these along with the other data sets.
6. Count Plot on Categorical Features:

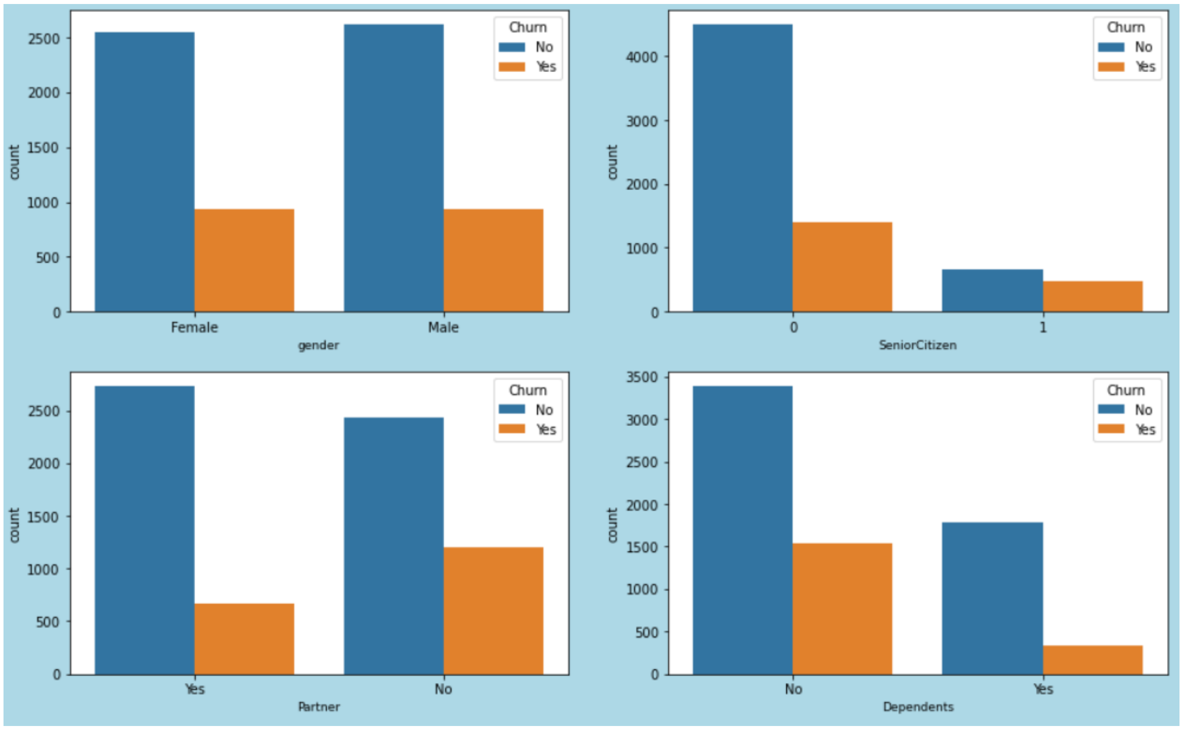
 Gender is distributed evenly.	 SeniorCitizen is distributed unevenly.
 Partner is distributed evenly.	 Dependents is distributed unevenly.

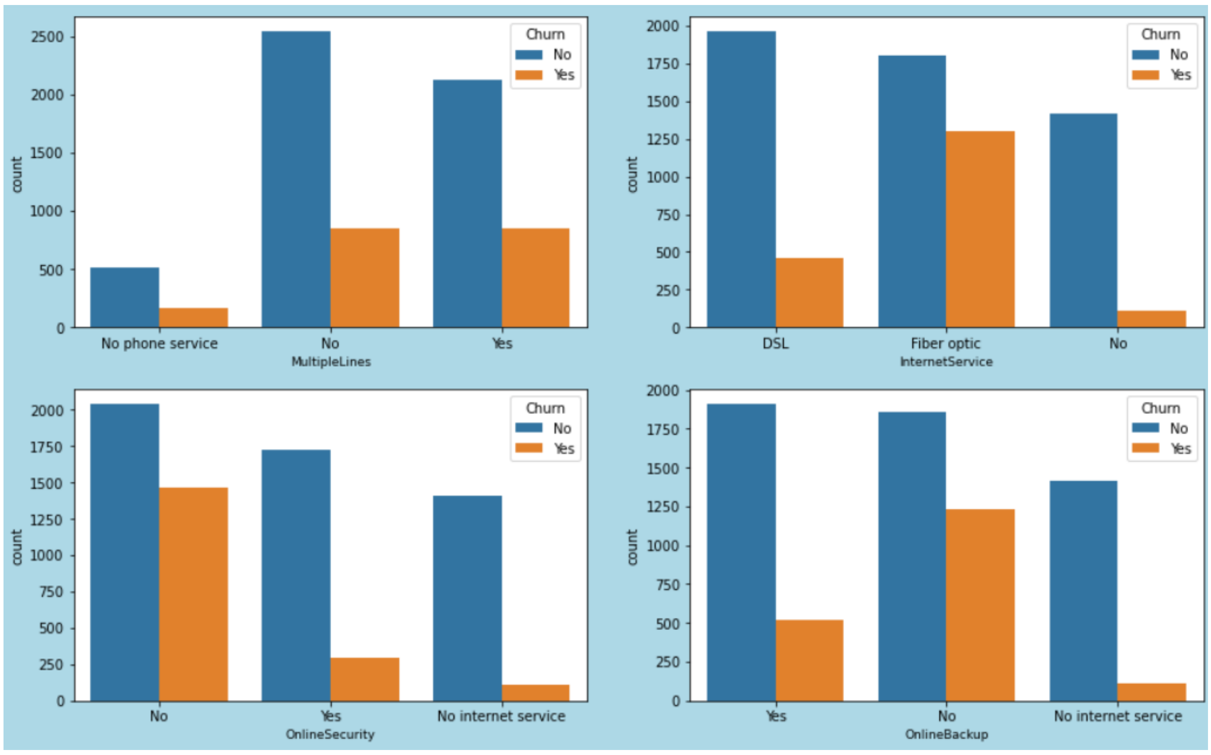
 PhoneService is distributed unevenly.	 MultiplotLines is distributed evenly.
 Fiber Internet serivice is distributed unevenly.	 OnlineService is distributed unevenly.
 OnlineBackup is distributed unevenly.	 DeviceProtection is distributed unevenly.
 TechSupport is distributed unevenly.	 Contract is distributed unevenly.
 PaperlessBilling is distributed unevenly.	 StreamingMovies is distributed evenly.

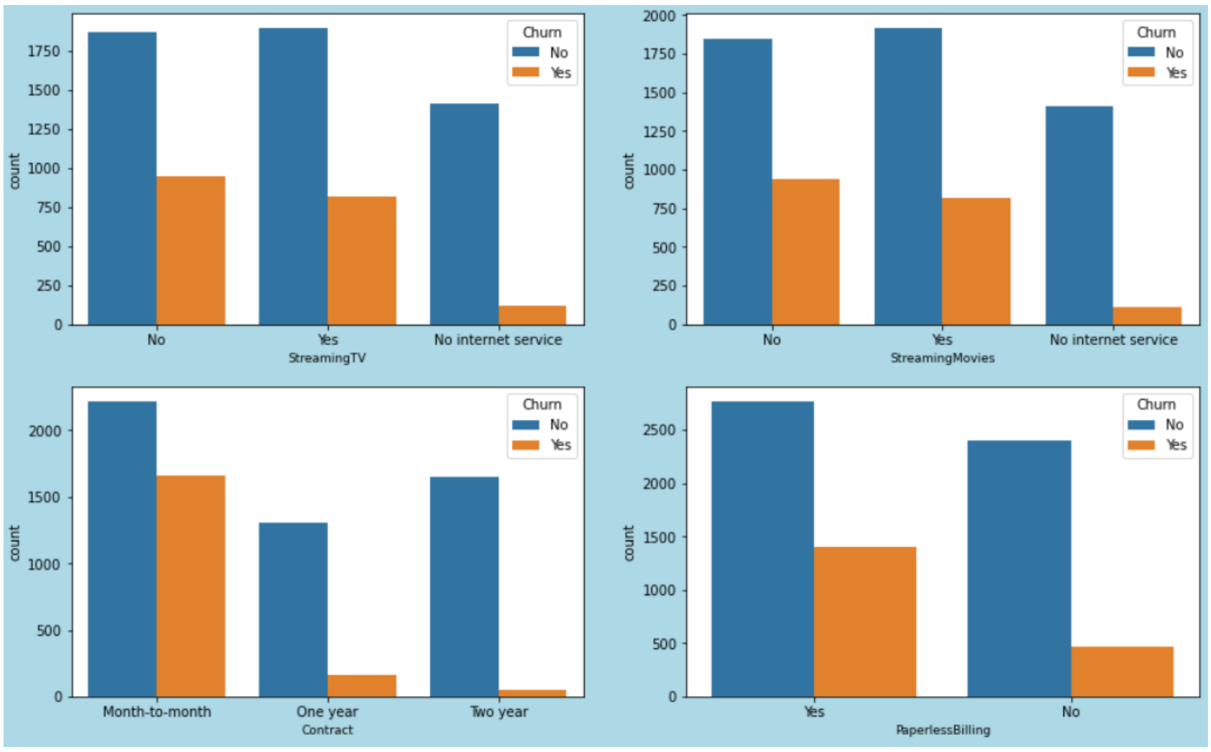
7. ‘MonthlyCharges’ is normally distributed.



8. Countplot compared with Churn numbers.




count
Churn
No
Yes
No phone service
No
Yes
MultipleLines
DSL
Fiber optic
No
InternetService
No
Yes
No internet service
OnlineSecurity
Yes
No
No internet service
OnlineBackup


count
Churn
No
Yes
No
Yes
No internet service
StreamingTV
No
Yes
No internet service
StreamingMovies
Month-to-month
One year
Two year
Contract
Yes
No
PaperlessBilling

DATA PREPROCESSING:

1. Encoding:

   Encoding done with Ordinal Encoder as below syntax.

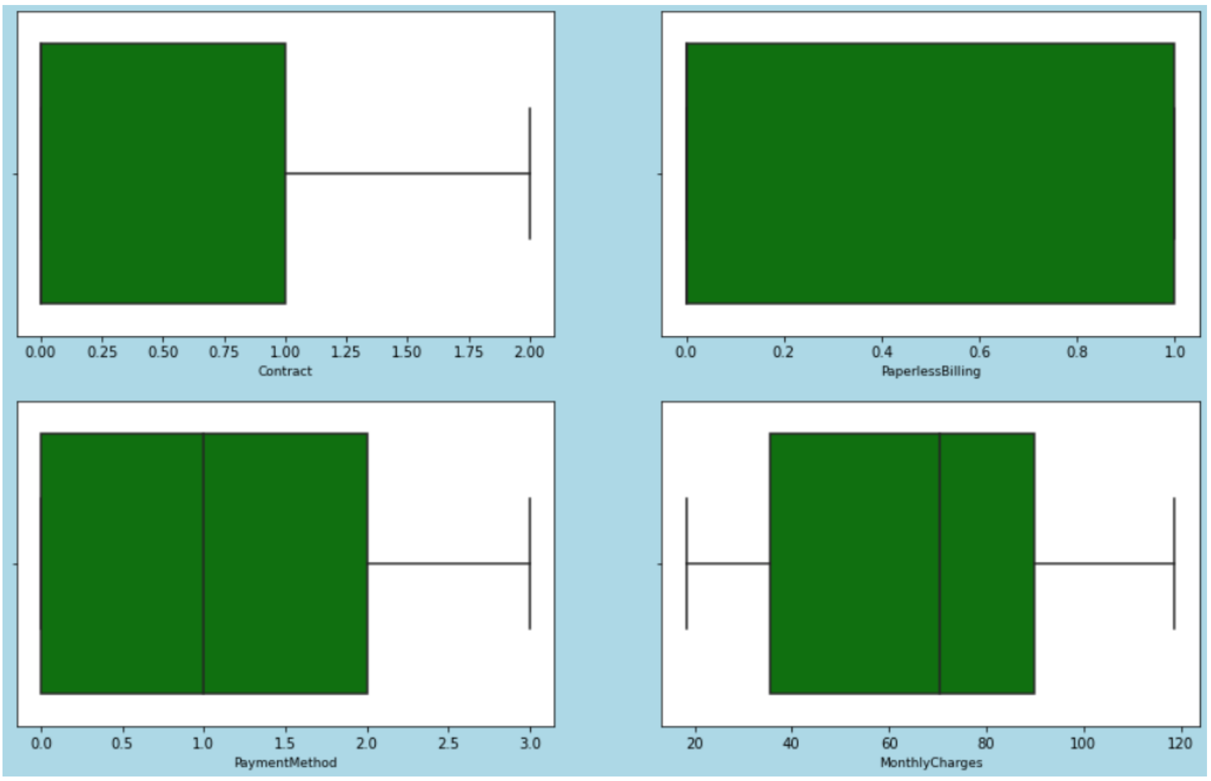
```
from sklearn.preprocessing import OrdinalEncoder
for i in data.columns:
    if data[i].dtypes == 'object':
        enco = OrdinalEncoder(categories=[data[i].unique()])
        data[i] = enco.fit_transform(data[[i]])
```

2. Outlier Treatment:

   Outlier can be found using the boxplot using below syntax.

```
plt.figure(figsize=(15,50),facecolor='lightblue')
plotnumber = 1
for column in data:
    if plotnumber<=20:
        ax=plt.subplot(10,2,plotnumber)
        sns.boxplot(data[column], color = 'green')
        plt.xlabel(column,fontsize=9)
    plotnumber+=1
plt.show()
```

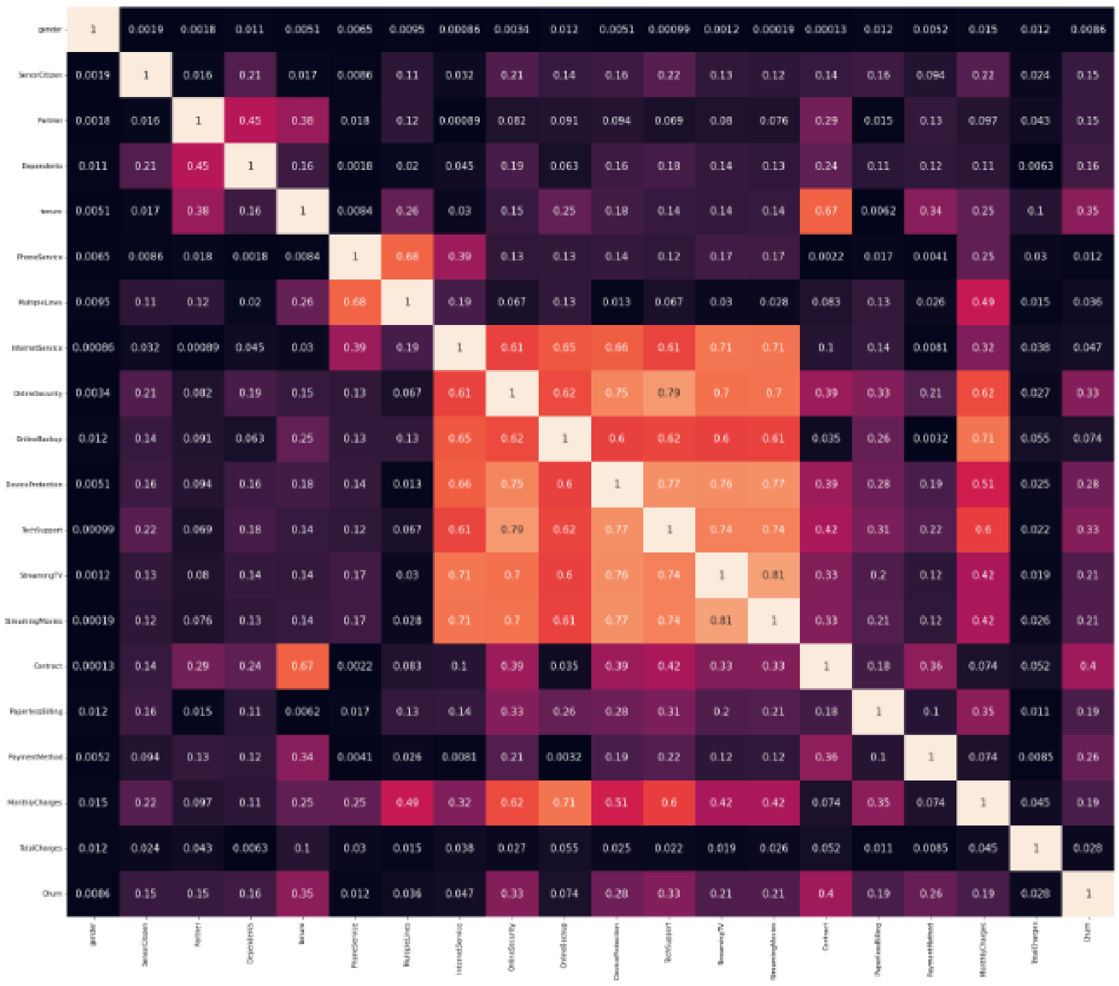
Sample box plot as below, no significant outlier has been found.



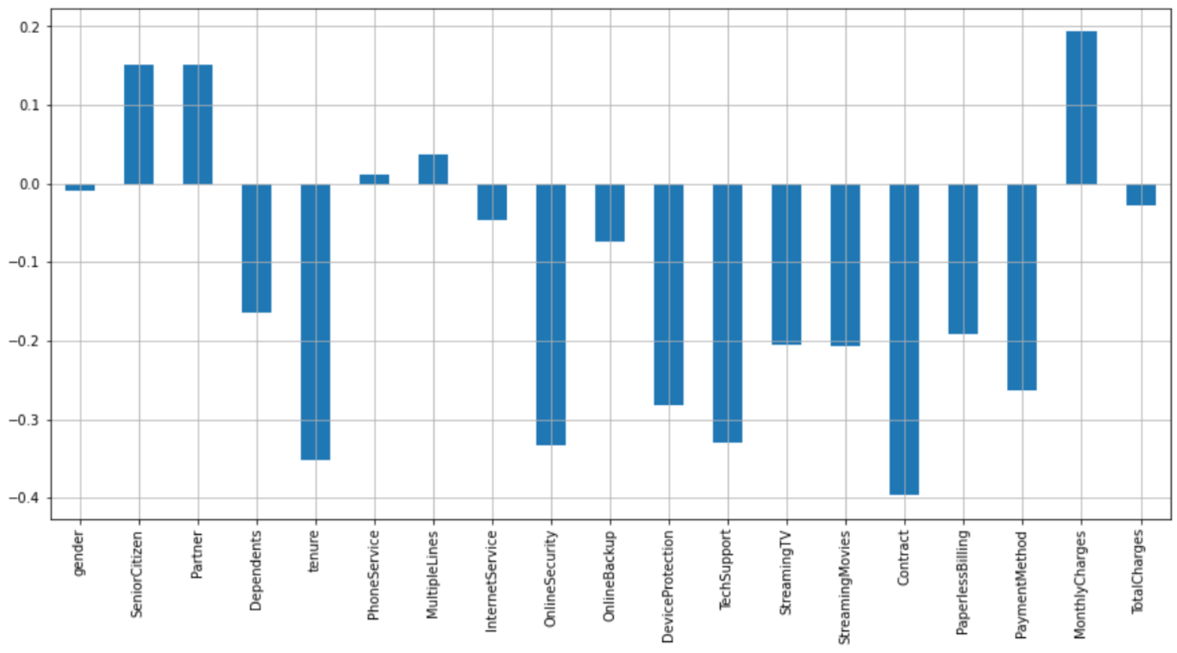
3. Feature Selection

   Features which have significant relation with target label can be found using heat plot and the corelation bar chart.

HeatPlot



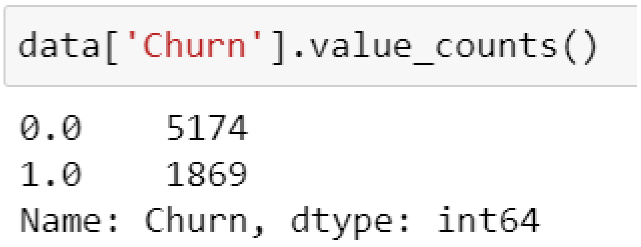
Co Relation Bar plot



- From the above two charts , dropping the corelated and less impact features
- Dropping StreamingTV, device protection, gender, PhoneServices and TotalCharges
- This will leads to no features will have corelation more than 75% with each other features.

4. Balancing Dataset

   We need to check for balance of target labels.

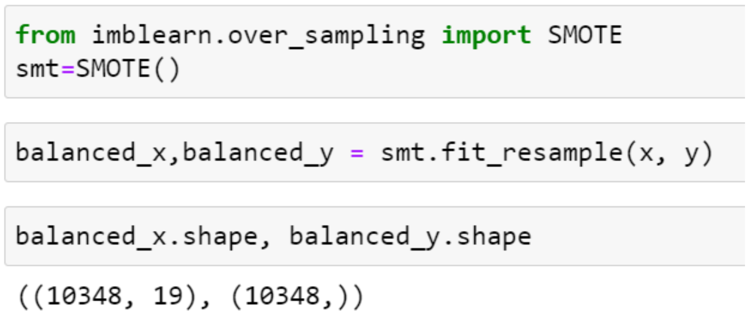


```
data['Churn'].value_counts()
```

```
0.0    5174
1.0    1869
Name: Churn, dtype: int64
```

   The target label is not balanced from the above image. We need to balance the dataset based on target label.

   Using SMOTE, we have balanced the dataset.



```
from imblearn.over_sampling import SMOTE
smt=SMOTE()
```

```
balanced_x,balanced_y = smt.fit_resample(x, y)
```

```
balanced_x.shape, balanced_y.shape
```

```
((10348, 19), (10348,))
```

   Now, the dataset is balanced having 10348 records on both the category.

5. Feature Scaling

   Before model building, all variables scaled a range of 0 to 1 and transformed. We have used Standard Scaler for scaling.

   We have checked for VIF Score for Multicoliinearity problem. All the feature have VIF score less than 5. So, its clear that there is no multi collinearity problem exists in the features.

MODEL BULIDING:

1. Train Test Split

   The train-test split is a **technique for evaluating the performance of a machine learning algorithm**. It can be used for classification or regression problems and can be used for any supervised learning algorithm. The procedure involves taking a dataset and dividing it into two subsets.

   We have considered 75% of data set as train data set and 25% as test data set.

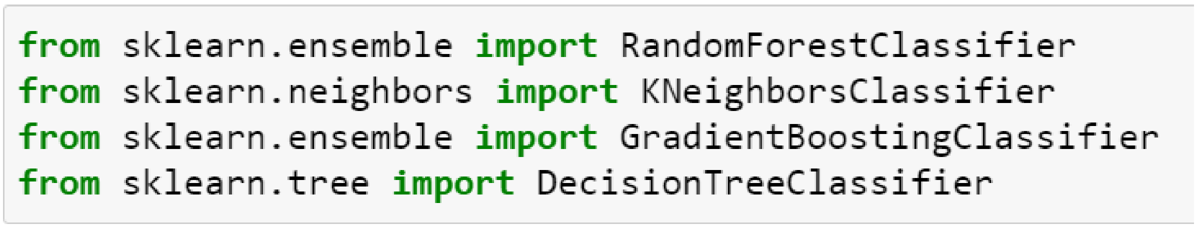
2. Random State

   We can use any random state for model build up. But we can select best random state using Linear Regression.

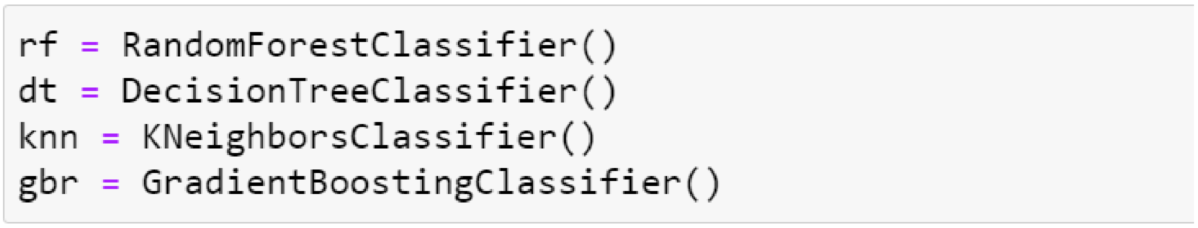
   Finally, we have selected 55 as random state as it gives best performance as 77%.

3. Model Selection

   Many Classification models can be builded for best performance for prediction.



```
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.tree import DecisionTreeClassifier
```



```
rf = RandomForestClassifier()
dt = DecisionTreeClassifier()
knn = KNeighborsClassifier()
gbr = GradientBoostingClassifier()
```

Logistic regression is basically a supervised classification algorithm. In a classification problem, the target variable(or output), y, can take only discrete values for a given set of features(or inputs), X.
Contrary to popular belief, logistic regression is a regression model. The model builds a regression model to predict the probability that a given data entry belongs to the category numbered as “1”. Just like Linear regression assumes that the data follows a linear function, Logistic regression models the data using the sigmoid function.

The bootstrapping Random Forest algorithm combines ensemble learning methods with the decision tree framework to create multiple randomly drawn decision trees from the data, averaging the results to output a result that often times leads to strong predictions/classifications.

The K-Nearest Neighbour or the KNN algorithm is a machine learning algorithm based on the supervised learning model. The K-NN algorithm works by assuming that similar things exist close to each other. Hence, the K-NN algorithm utilises feature similarity between the new data points and the points in the training set (available cases) to predict the values of the new data points. In essence, the K-NN algorithm assigns a value to the latest data point based on how closely it resembles the points in the training set. K-NN algorithm finds application in both classification and regression problems but is mainly used for classification problems.

Gradient boosting classifier is a set of machine learning algorithms that include several weaker models to combine them into a strong big one with highly predictive output. Models of a kind are popular due to their ability to classify datasets effectively. Gradient boosting classifier usually uses decision trees in model building.

4. Cross Validation of Model

The Model mat give high performance due to overfitting. So we need to validate the model.

We have used the K-Fold Cross Validation for model validation.

The mean of the validation score has been calculated to consider the model validation score.

5. Best Model

The below table shows the model performance

Model	Model Performance	Cross Validation score
Logistic Regression	76.8%	75.3%
Random Forest	86.2%	70.6%
Decision Tree	80.4%	59.8%
KNN Classifier	75.9%	55.6%
Gradient Boost DT	86.0%	76.7%

From the above table, we can select GRADIENT BOOST DECISION TREE as the final model.

6. Hyperparameter Tuning

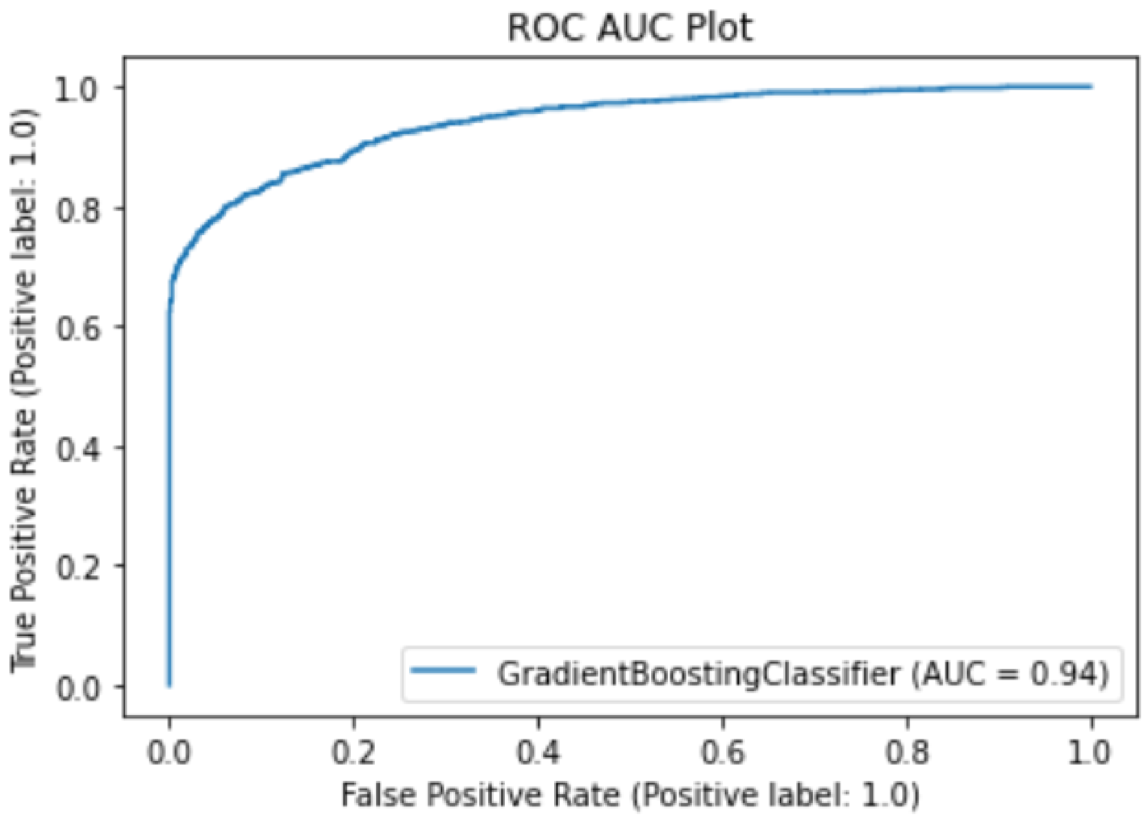
Parameter tuning has been done on final model and model performance is improved to 87% having improvement on train dataset as well with 92% performance.

7. ROC AUC Curve

It is computed based on the receiver operating characteristic (ROC) curve that illustrates the diagnostic ability of a given binary classifier system as its discrimination threshold is varied. The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.

The area under the ROC curve (AUC)[1][2] is often used to summarize in a single number the diagnostic ability of the classifier. The AUC is simply defined as the area of the ROC space that lies below the ROC curve.

Here, in this dataset we have attained ROC AUC plot with 94% coverage as shown in below image



CONCLUSION:

1. We have saved the final model using Pickle library.
2. From exploratory data analysis we can conclude that, recent clients are more likely to churn and the clients with higher MonthlyCharges are also more likely to churn. Senior cit izens are only 16% of customers, but they have a much higher churn rate: 42% against 23% for non senior customers. There are no special relations between this categorical values and the main numerical features. Additional services variables (OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies) are plotted with churn rates and these results obtained; customers with the first 4 additional services (OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport) are more unli kely to churn and

streaming service is not predictive for churn. Payment methods plotted with

3. churn rates and these results obtained; customers with paperless billing are more probable to churn and the preferred payment method is electronic check with around 35% of customers.
4. We have chance of improvement by making dataset more informative and having the balanced dataset with actual data. We have balanced the dataset. But we can have originally balanced dataset for more accuracy.
5. However, when a new model was created, its accuracy was only slightly better than that given by a logistic regression.
6. For neural networks as well, accuracy is only more with a very tiny difference with Gradient Boost Algorithm.

END OF PROJECT