**Customer Churn Analysis**

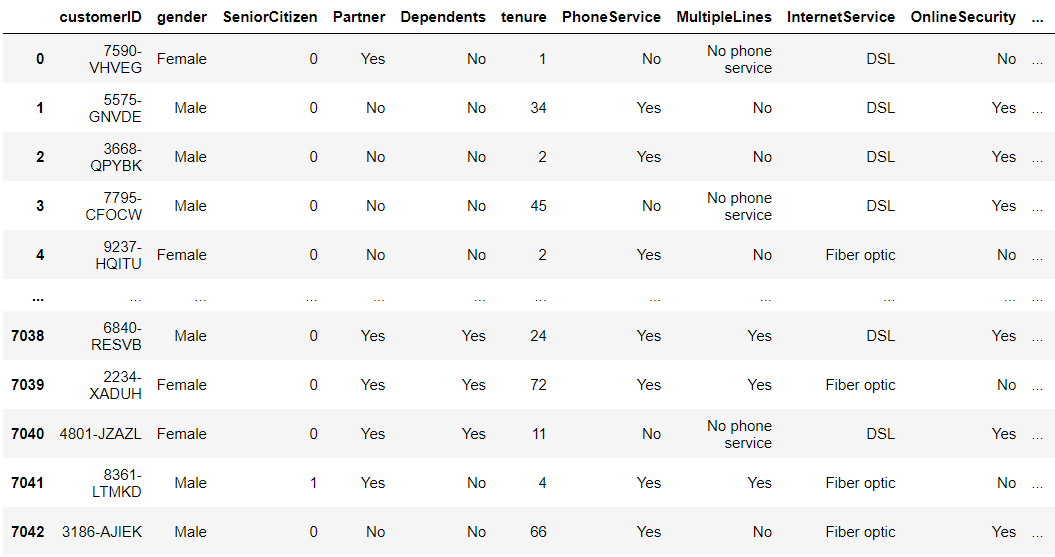
1. Problem Definition

Customer churn is when a company’s customers stop doing business with that company. Businesses are very keen on measuring churn because keeping an existing customer is far less expensive than acquiring a new customer. New business involves working leads through a sales funnel, using marketing and sales budgets to gain additional customers. Existing customers will often have a higher volume of service consumption and can generate additional customer referrals. 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. Our objective is to build a model which is accurate enough to predict customer churn beforehand using a set of services provided to the customer.

2. Data Analysis

2.1 Data Import:

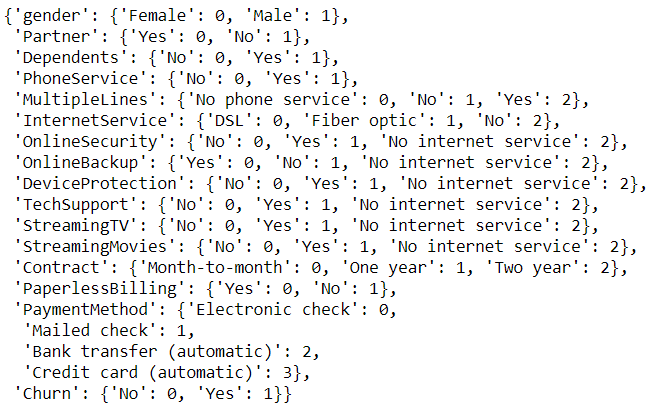
* The dataset is in csv format, we shall import the dataset using ‘read\_csv’ function.
* Once the dataset is imported and converted into a data frame, store the data frame and print it, to analyse the datapoints in rows and column.



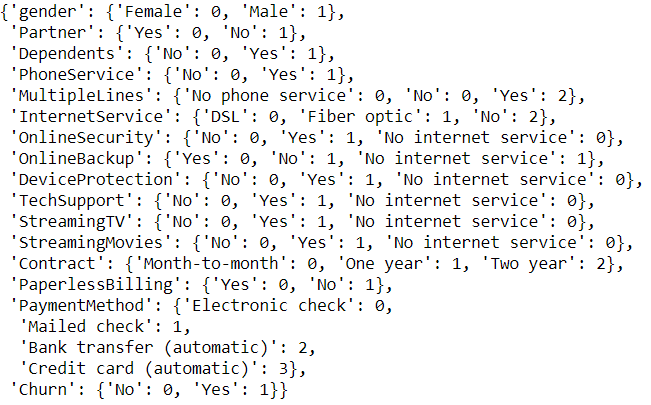
* Unfortunately, due to pandas limitation not all rows and columns are displayed.
* Our project’s objective is to predict customer churn beforehand using a set of services provided to the customer, hence column “Churn” is our target variable and our input features are 'gender', 'SeniorCitizen', 'Partner', 'Dependents', 'tenure', 'PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract', 'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges', 'TotalCharges'.

2.2 Data Analysis:

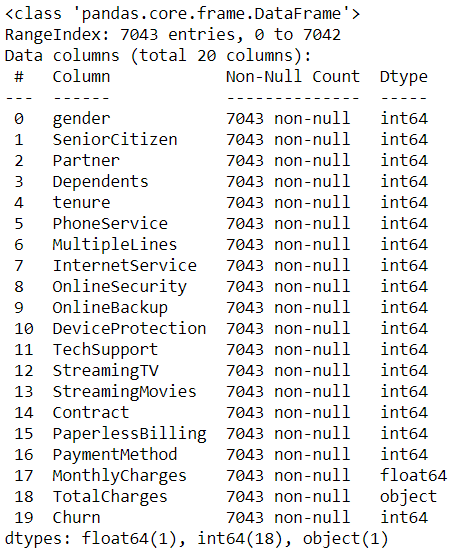
* With the help of pandas inbuilt functions, we can deduce following key points:
  + Number of rows in the dataset are 7043.
  + Number of columns in the dataset are 20.
  + Dataset contains does not any null values.
  + Categorical columns are: 'gender', 'Partner', 'Dependents', 'PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract', 'PaperlessBilling', 'PaymentMethod', 'Churn'.
  + Non-Categorical columns are: tenure, ‘MonthlyCharges’, ‘TotalCharges’.
* In customer ID column all the values are unique, it has nothing to do with model training so let me drop this ID column.
* Further analysis requires our categorical column to be converted into integer datatype column.
* For that we need to find the unique categorical values present in the column and replace the string value by an integer.
* Extract the columns and its unique values and display them.



* Above are the value counts of each column and I can see some duplicate entries in MultipleLines, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV and StreamingMovies.So I have to replace the values for those columns.



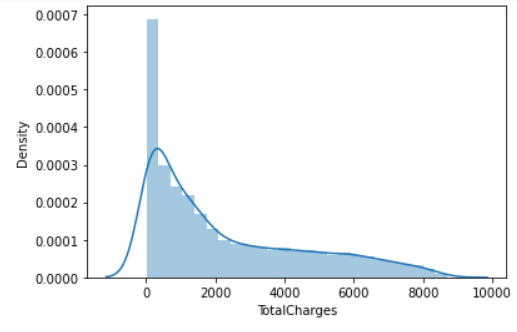
* I have replaced No internet Service with No. If no internet service then there is no MultipleLines, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV and StreamingMovies. So I will replace all these entries with 'No'.
* Execute pandas inbuilt info function to provide detailed information of each and every column.



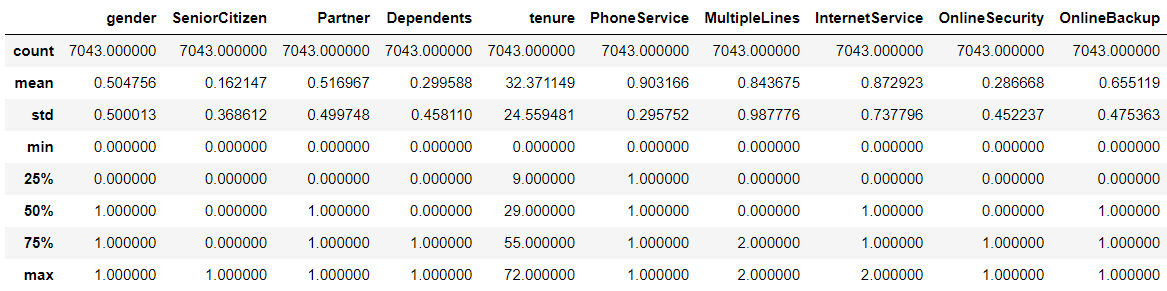
* From above we can take a note of each column and their datatype.
* From above output we can see that column ‘TotalCharges’ is of object datatype instead of float type.
* Check if there are any rows of tota charges filled with blank spaces and display the number of blank rows.

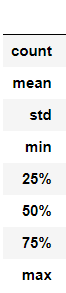
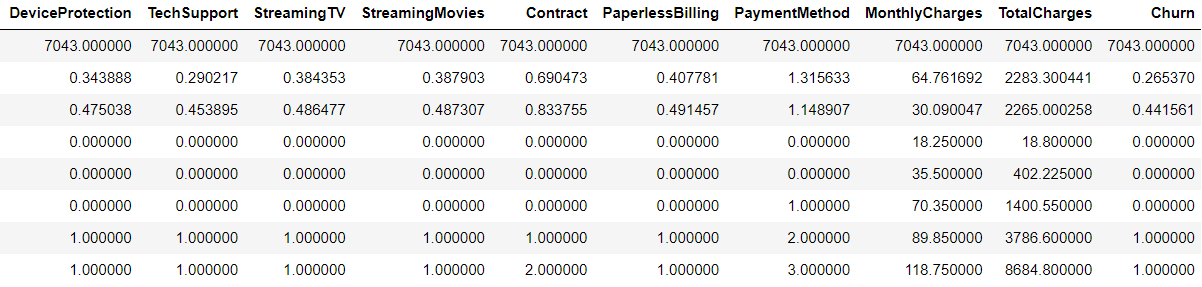


* There are total of 11 blank rows, find the distribution plot of column total charges to find out the value which can be used to replace with the blank spaces.

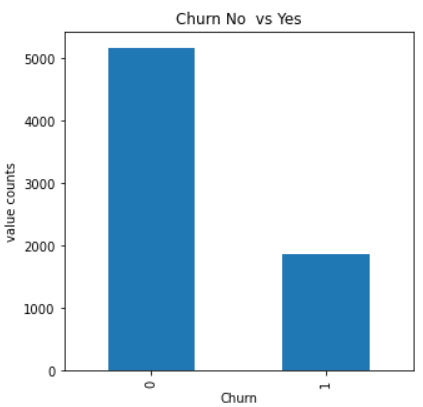


* Distribution plot is of type normal distribution, hence fill the missing values with mean of totalcharges column and then change the dtype of column totalcharges.
* For further analyses of the dataset, we require Statistical Summary of each column such as mean value, median value, max value, min value, standard deviation value.
* Describe function of pandas can provide us with all statistical summary along with the count of non-null rows, lower percentile, upper percentiles.
* In statistics, a percentile is a score *at or below which* a given percentage falls.
* For example, the 50th percentile is the score below which 50% of the scores in the distribution may be found.

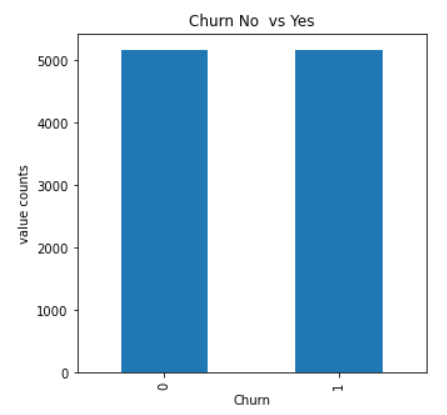


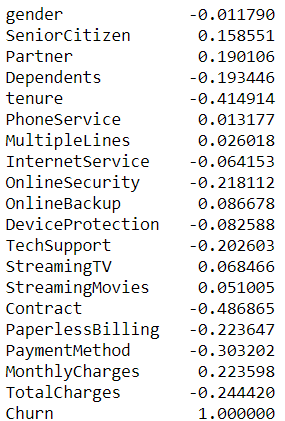
* As this is a classification problem we also need to check if there is data imbalancement.
* Data imbalancement is generally caused when one class datapoints are high compared to that of the other class.
* For example in this project if churn value counts of yes/No is higher than the other, then there is data imbalancement problem.
* Evaluate the value counts of churn datapoints.



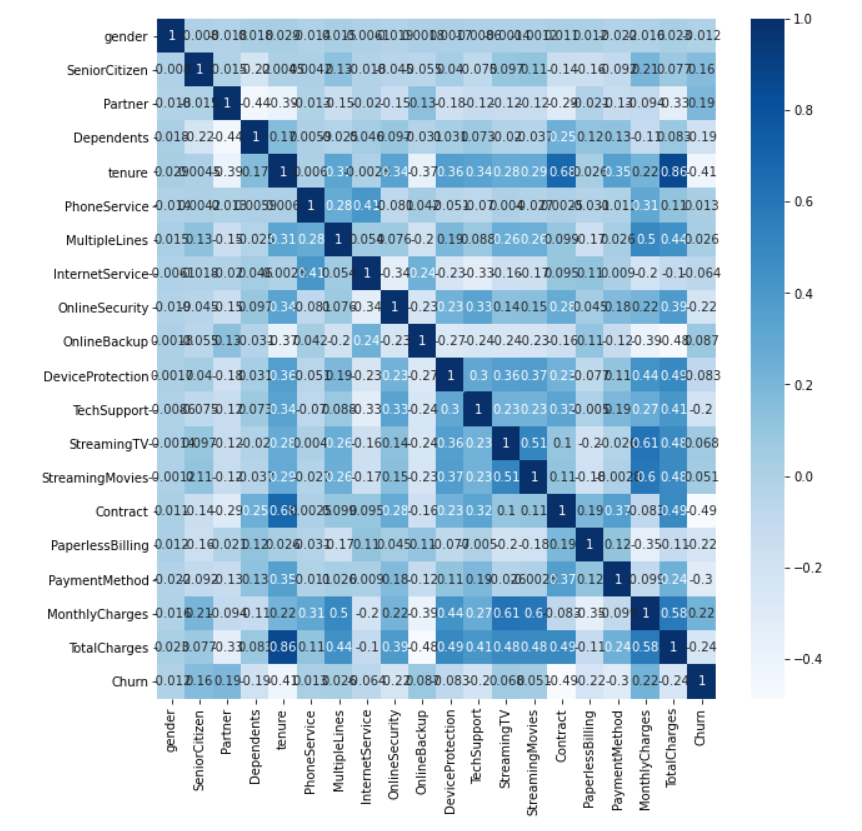
* Number of Churn ‘no’ datapoints are 5174.
* Number of Churn ‘yes’ datapoints are 1869.
* Thus we can deduce that we have an unbalanced dataset.
* Balance the dataset using resample function and plot bar graph.



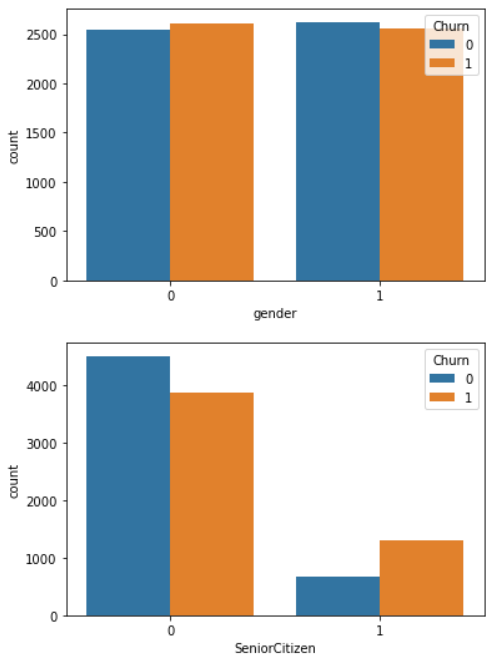
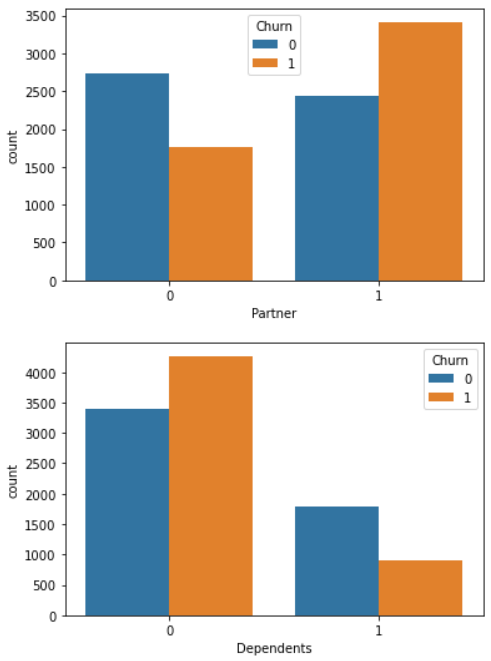
* Next Step is to check and analyse the correlation of input features w.r.t output column.

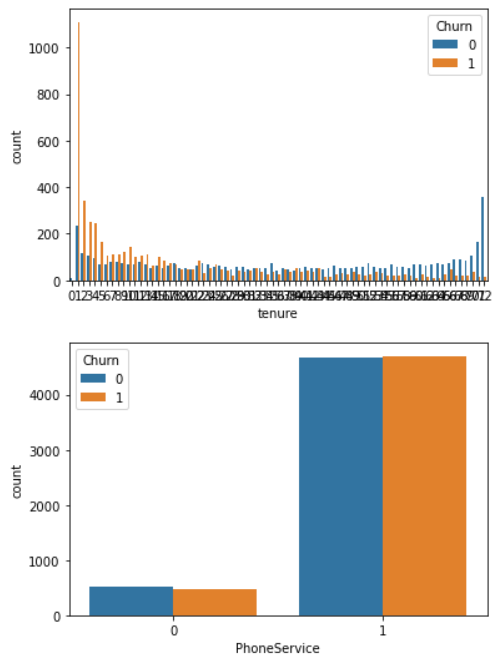
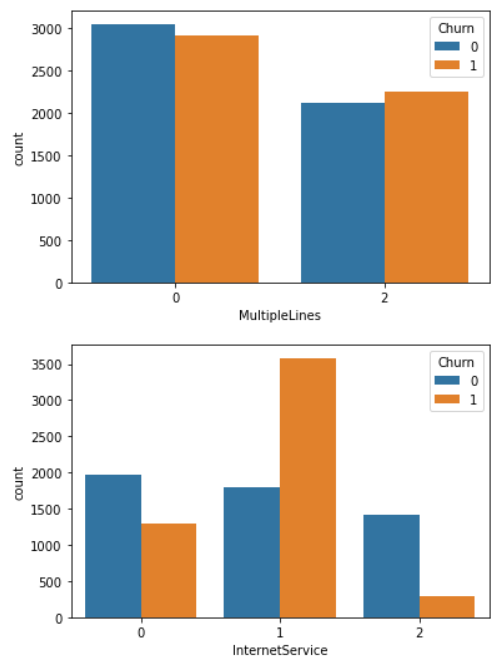


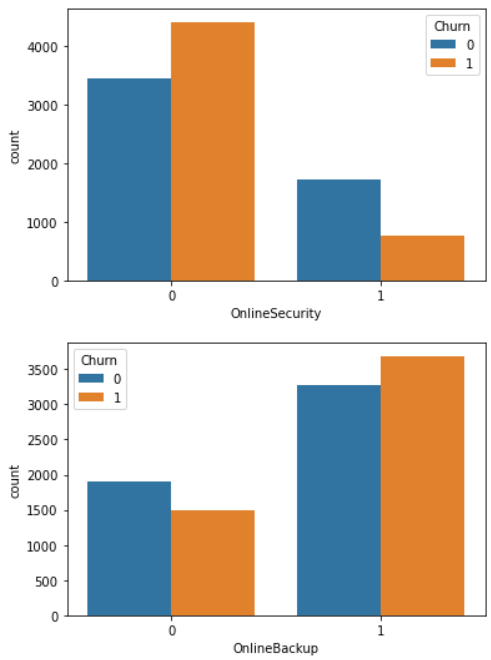
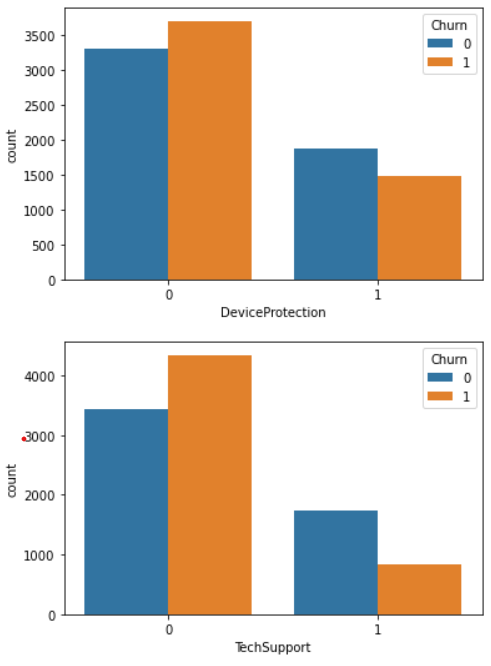
* We can deduce which columns are highly correlated to the target column and which columns are least correlated. We perform these analyses to determine which columns can dropped based on their correlation w.r.t target variable.
* Visualising correlation of the entire dataset using Heatmap.

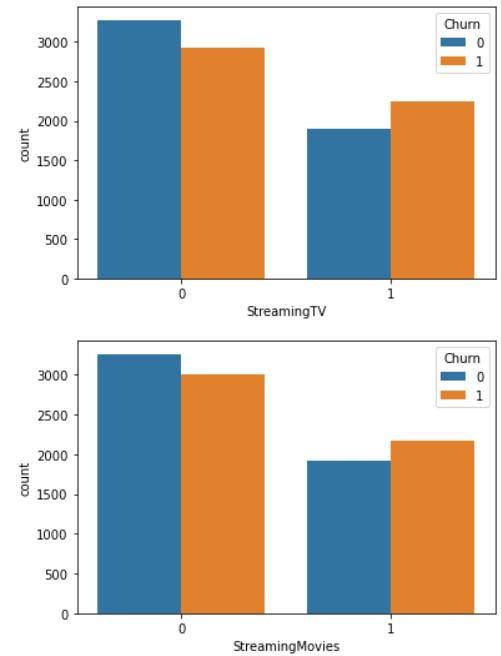
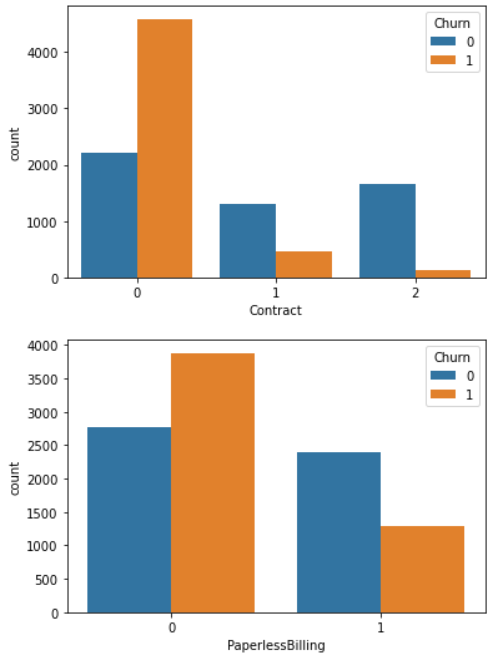


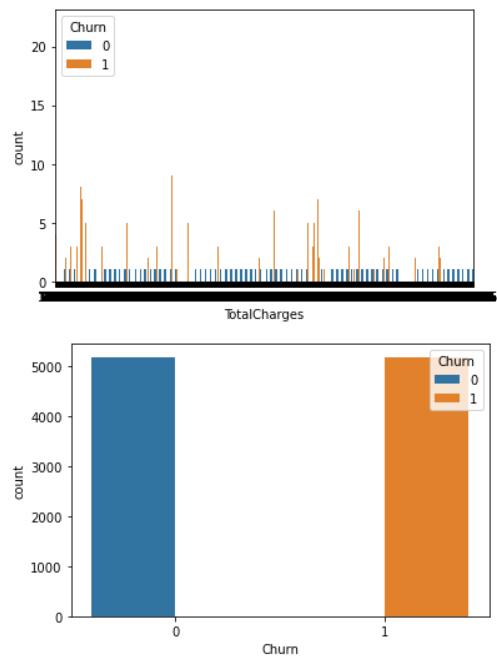
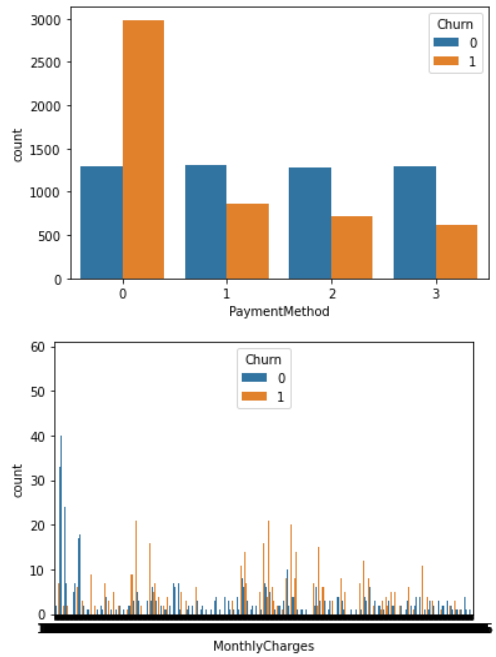
* Check the graphical representation of input feature column w.r.t target variable to analyse their relation.



### Acc to above plot, gender, phoneservice, multiplelines columns are unable to distinguish between churn 0 (no) and churn 1(yes). Meaning it does not provide much impact to the model. Therefore, we shall drop those columns.

3. Exploratory data analysis

### Check distribution plot of each column to determine whether a column has datapoints representing normal distribution or bimodal distribution or rectangular distribution plot.

### 

### 

### From above we can deduce that only column totalcharges has normal distributed datapoints with positive skewness, rest all have bimodal distribution plots.

### We must also check for outliers using boxplot in our dataset as it can hamper our model performance.

### Outliers of only non-categorical data should be checked.

### 

### 

### In the column of totalcharges outliers are shown because of biasing issue that will solved in further steps. so as far now no need to remove outlier.

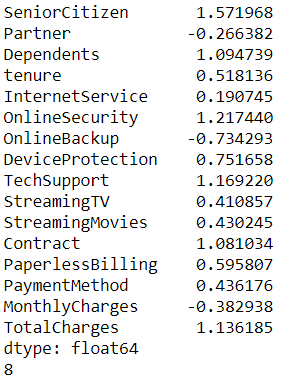
EDA Concluding Remark

* Gender, phoneservice, multiplelines & customerID columns have least correlation & they can be dropped for the purpose of training the model.
* Totalcharges has normal distributed datapoints with positive skewness, rest all have bimodal distribution plots.

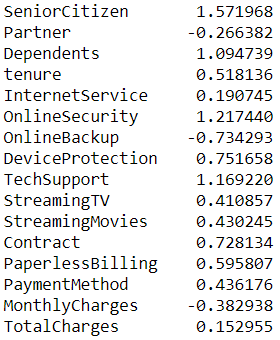
### Totalcharges has outliers present because of biasing issue that will solved in further steps. so as far now no need to remove outlier.

4. Pre-Processing Pipeline

* To train the model we need to split target variable and input features.
* To avoid biasing issue, we need to check for biasing using skew method of input features and remove if required.

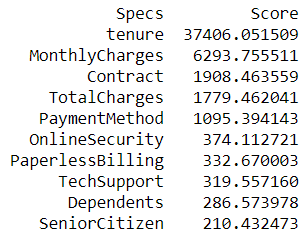


* Skewness is present if skew value of a column is more than 0.55.
* From above table we can see biasing exists in column SeniorCitizen, Dependents, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, Contract, Paperbilling, TotalCharges.
* Remove biasing of specific column by replacing its data points by cube root of its data points.
* Check for biasing again.



### There was biasing mostly present in totalcharges column which is now fixed. hence the output is acceptable.

### To improve on the model accuracy, we need to find the best features using Univariate Selection.



* From above we can observe the columns w.r.t its score regarding the column’s importance for model training. By evaluating columns according to its score we can obtain the best features required to train the model.
* According to the score we will pick tenure, MonthlyCharges, Contract, TotalCharges & PaymentMethod features to train the model.
* Perform feature scaling using standard scaler algorithm on non-categorical columns only.
* The objective of performing feature Scaling is to standardize the independent features present in the data in a fixed range. It is performed during the data pre-processing.

5. Building Machine Learning Models

### Now we will start with model selection and fine-tuning process.

### First, we need to find the most optimum model.

### We shall evaluate model on roc auc score & cv score.

### The types of models through which we need to iterate are:

### LogisticRegression

### DecisionTreeClassifier

### KNeighborsClassifier

### RandomForestClassifier

### SVC

### RidgeClassifier

### BaggingClassifier

### GradientBoostingClassifier

### SGDClassifier

### LGBMClassifier

### XGBClassifier

### ExtraTreesClassifier

### AdaBoostClassifier

### QuadraticDiscriminantAnalysis

### CalibratedClassifierCV

### LinearSVC

### NuSVC

### LinearDiscriminantAnalysis

### RidgeClassifierCV

### GaussianNB

### BernoulliNB

### PassiveAggressiveClassifier

### Perceptron

### DummyClassifier

### After iterating through all the above algorithms, we obtained the following roc auc accuracy, cv score and the difference between roc auc score and cv score.

### Display the top 5 models metrics.

|  |  |  |  |
| --- | --- | --- | --- |
| Model | CV Score | Roc Auc score | Difference |
| Decision Tree Classifier | 85% | 88% | 3% |
| KNeighbors Classifier | 84% | 78% | 5% |
| Random Forest Classifier | 93% | 89% | 4% |
| Bagging Classifier | 92% | 88% | 4% |
| Extra Trees Classifier | 95% | 90% | 5% |

### From above we can conclude that Extra Trees Classifier is the best model without any issues of underfitting or overfitting.

### Perform fine tuning on Extra Trees Classifier model and find the best parameters to be used for the model by using GridsearchCV algorithm.

### For grid search use of the following parameters for extra trees classifier:

### 'criterion': ['gini', 'entropy']

### 'max\_features': ['auto', 'sqrt', 'log2']

### 'max\_depth': list(range(3,36))

### Best parameters for Extra Trees Classifier obtained by executing Grid Search are:

* + 'criterion': entropy
  + 'max\_features': 'auto'
  + 'max\_depth ': 33

### Now we need to find the most optimum random state in train test split for Extra Trees Classifier model to get best score, in this case best random state is 99.

### We are going to split the dataset by keeping 20% for testing and 80% for training.

* Using the model extra trees classifier along with the best parameters obtained from grid search, fit the model wrt the train dataset.
* Evaluate the model based on test dataset and obtain all the metrics of the currently trained model:
  + f1 score is: 91.33%
  + AUC ROC Score: 90.96
  + CV mean is: 96.86
  + Std error is: 0.36%
  + Confusion matrix: 49 False Negative

138 False Positive

897 True Negative

986 True Positive

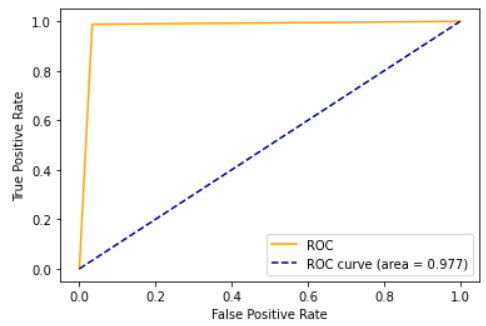
* As observed, the accuracy of the model varies from 91.4% to 91%.
* We shall evaluate the model on the whole dataset to re-check it’s performance.
  + f1 score is: 97.74%
  + AUC ROC Score: 97.71%
  + CV mean 96.82%
  + Std error is: 0.37%
  + Confusion matrix: 61 False Negative

175 False Positive

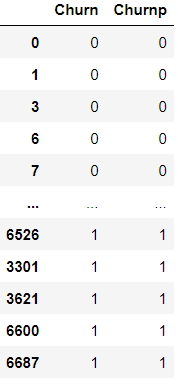
4999 True Negative

5113 True Positive

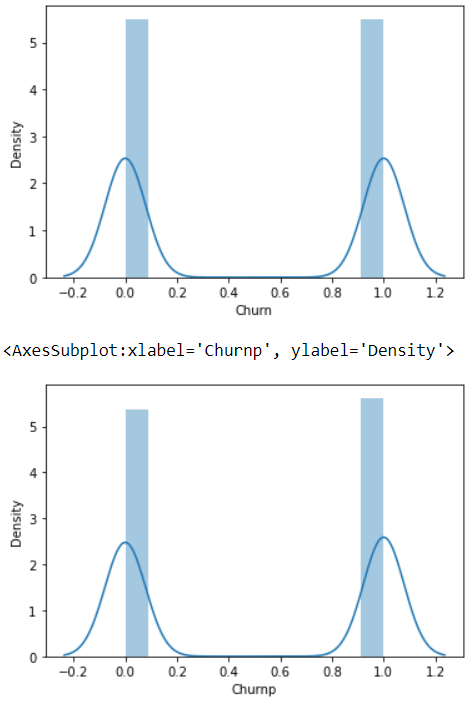
* As observed, the accuracy of the model varies from 97.74% to 97.71%.
* Plot roc curve of the model to visualize the accuracy of the model.



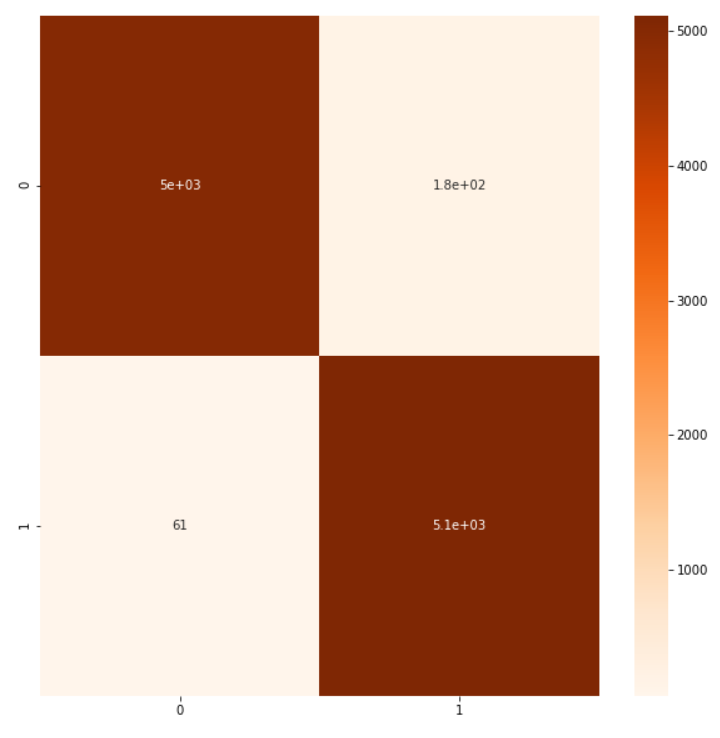
### Put target variable ‘Churn’ column & predicted “Churn” column side by side to observe the difference between the actual and predicted datapoints.



* Unfortunately, we cannot observe the difference between the actual and predicted because the dataset is too large for us to go through the values, hence plot appropriate graphs to check the similarity between actual and predicted values.
* First check distribution plot of both the columns.



* They look very similar, but plot more graphs to evaluate model more accurately.
* Display the heatmap of confusion matrix.



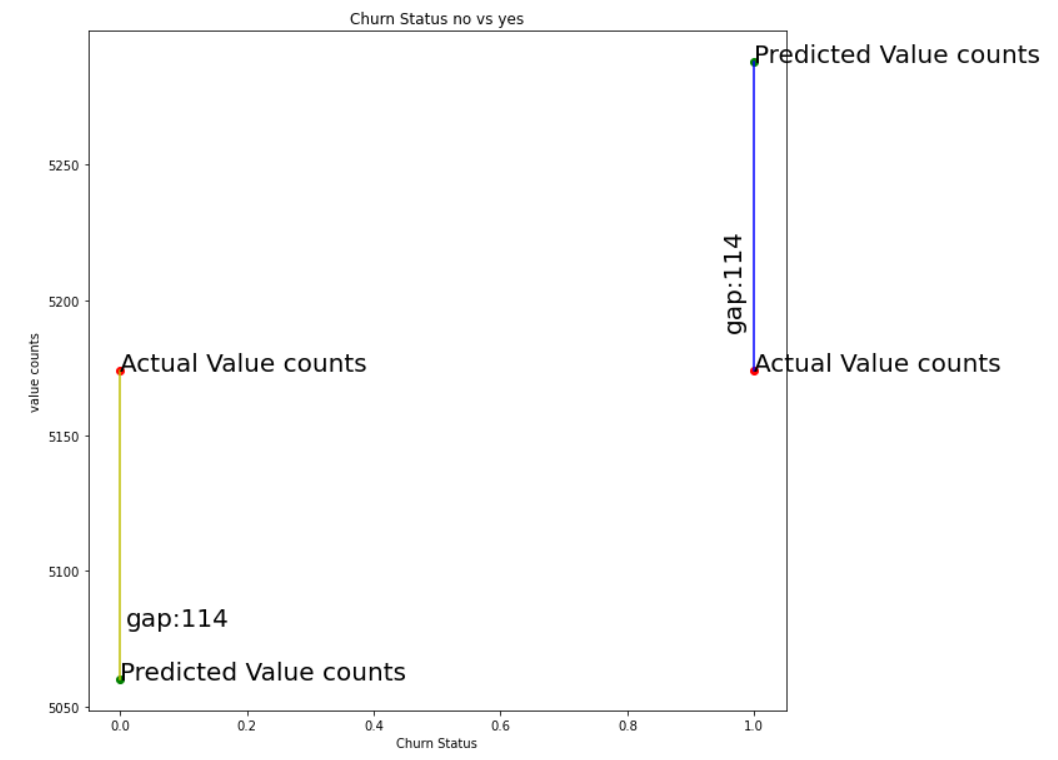
* Confusion matrix: 61 False Negative

175 False Positive

4999 True Negative

5113 True Positive

* Upon executing value counts function on predicted churn values we know that value counts is 5288 for churn=1 and for churn=0, value count is 5060.
* Using custom graph, we shall now plot a graph where we can see the difference in value counts of actual and predicted datapoints for both classes.



### Finally, we save the model using “joblib” library which can be reused for further prediction.

6. Concluding remarks

We were able to build a model having 97% accuracy in predicting the customer churn beforehand using a set of services provided to the customer. We used Extra trees classifier model obtain high accuracy, hence we obtained low standard deviation error along with no overfitting or underfitting issue. We also observed the difference between actual and predicted value. By further visualizing of datapoints we can conclude that the model is accurately able to predict customer churn. The saved model can be loaded and used again to predict customer churn. Model’s accuracy can increase if more training data is provided.

### Click the link below to go through the jupyter notebook:

### [Customer Churn Analysis](https://github.com/genos1998/datatrained-project/blob/main/Customer%20Churn%20Analysis.ipynb)