**Telecom Churn Prediction**

**Problem description**

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.

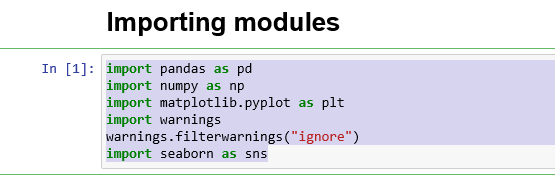
Preventing customer churn is critically important to the telecommunications sector, as the barriers to entry for switching services are so low.

# Steps of the project

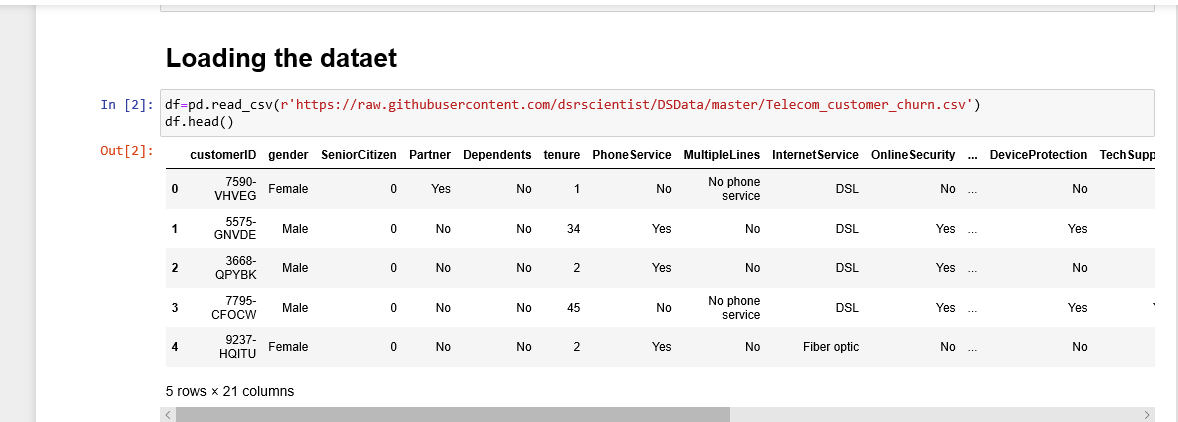
The project consists of the following sections:

* Importing necessary libraries.
* Importing dataset from GitHub.
* Exploratory Data Analysis (EDA) and Data Visualisation.
* Data Preprocessing & Feature Engineering.
* Model building and Saving.

**1. Importing necessary libraries:**

****

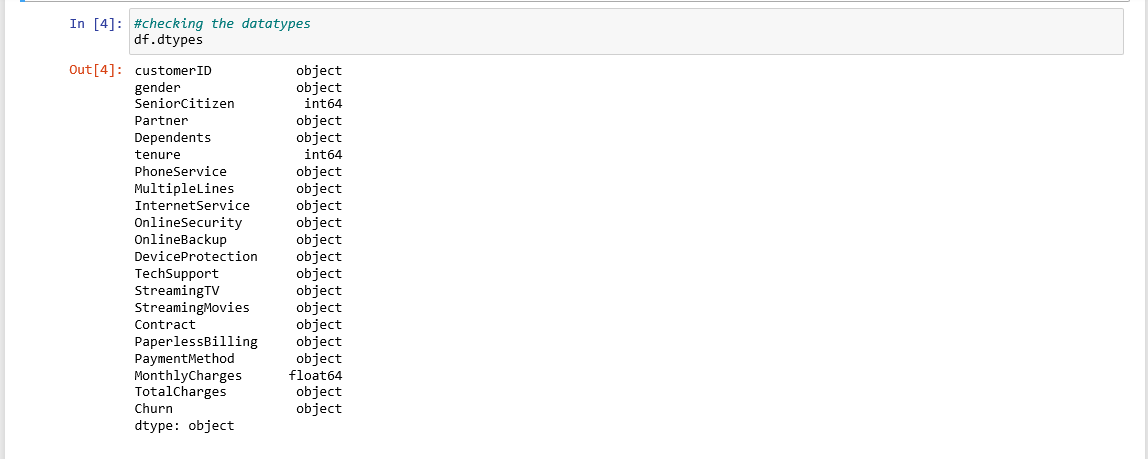
**2. Importing Dataset from Github:**



**3. Exploratory Data Analysis & Data Visualisation:**



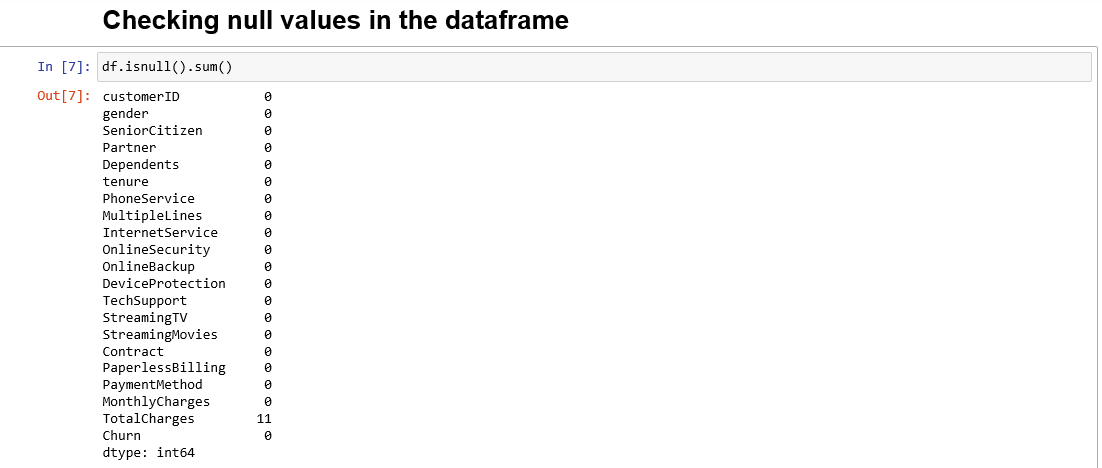
The Dataset contains 7043 rows and 21 columns.



We see that we have a mixture of categorical and numeric columns. We have 3 numerical columns and 18 object type columns. The feature TotalCharges is object datatype but in the data set we can see float values, so we will have to change it to float datatype.



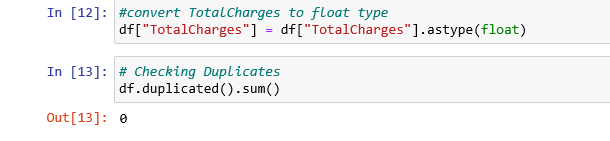
We observe that the feature TotalCharges has 11 rows with missing values are present as empty blank spaces. We will replace the empty spaces with null value .



We can observe that except TotalCharges column there is no other null value present in our dataset. Next we check the percentage of data missing.



Since Only 0.15% of information is lost will dropping the null values we decide to drop the null values.



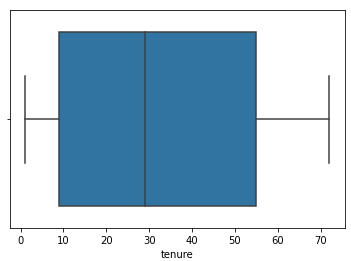
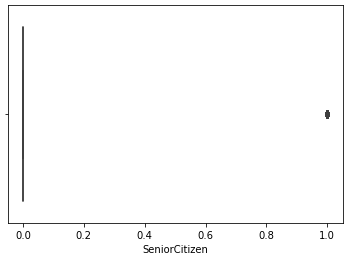
As discussed above, the datatype of TotalCharges is ‘Object’ and we have converted it to float type. Also we checked if any duplicates are present in our data set and we could see that the dataset does not contain any duplicates.

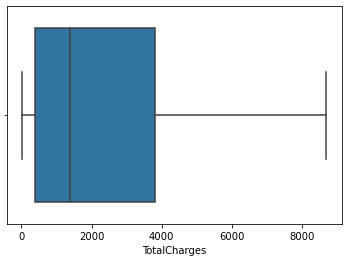
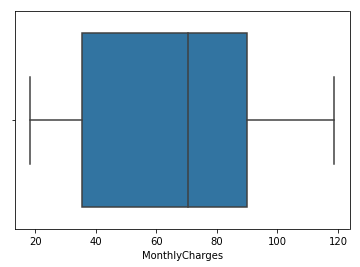
## Remove customerID column

The customer ID column is useless to explain whether not the customer will churn. Therefore, we drop this column from the data set.

Screenshot (62).png

**Univariate Analysis(Numerical Data)**

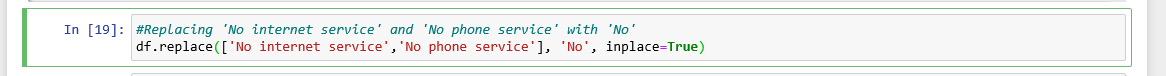
****

****

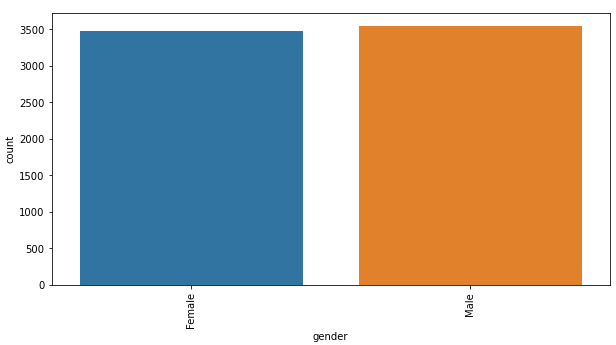
**Observations:**

* Feature Senior citizen has discrete values with 1 indicating the person is senior citizen and 0 indicating the person is not.
* Maximum Tenure period is around 70 months and minimum around 0 months and the average is around 30months.
* Maximum MonthlyCharges is around 120 and minimum around 20 and the average monthly charges is around 70.
* Maximum TotalCharges is around 8500 and minimum around 0 and the average TotalCharges is around 1800.

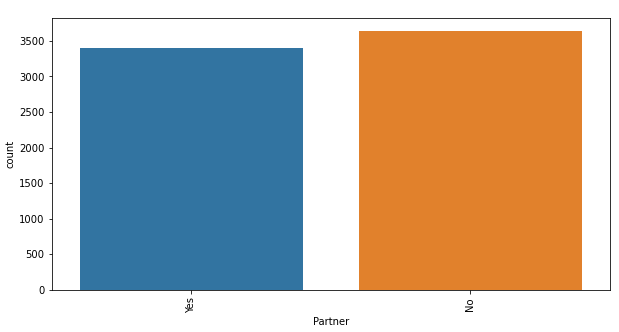
**Univariate Analysis(Categorical Data)**

****

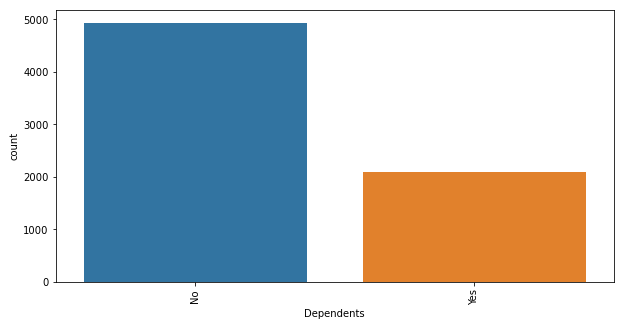
We replace ‘No Internet Service’ and ‘No phone service’ in the dataset with no.



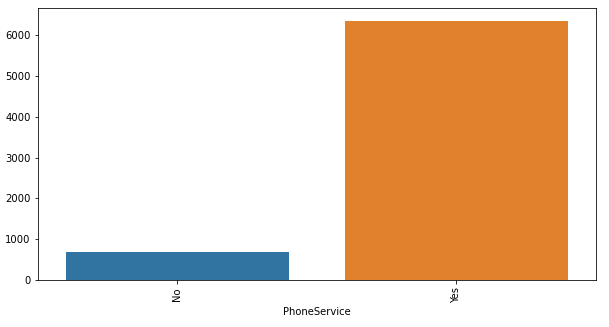
* There is negligible difference in customer percentage/count who changed the service provider. Both genders behaved in similar fashion when it comes to migrating to another service provider/firm.



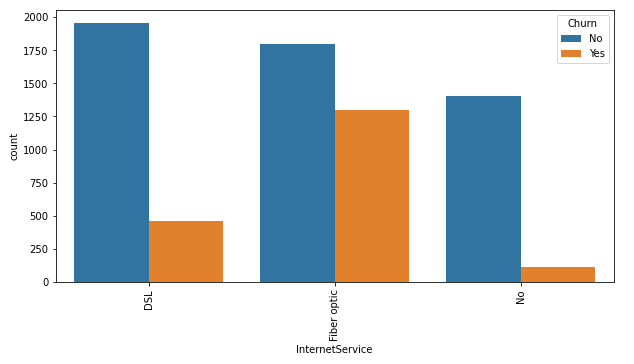
* Customer with partner have slightly higher majority.



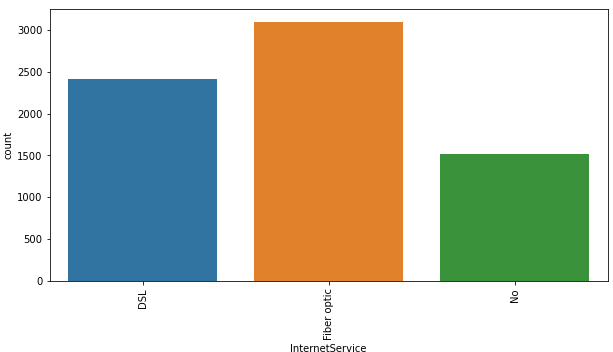
* Majority of customer donot have dependents.

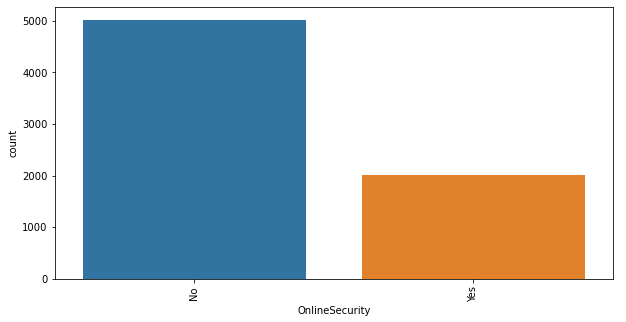


* Majority of customer have Phoneservice and only few are there who donot have.

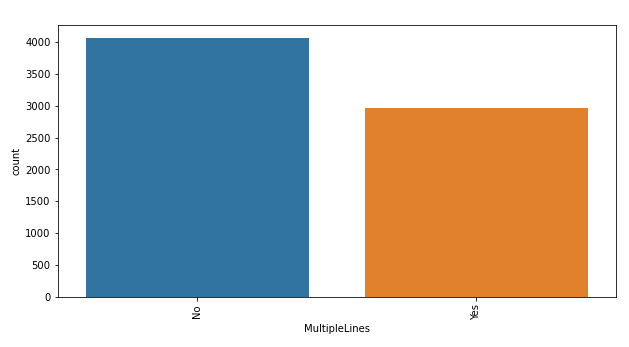


* Majority of customer prefers Fiber optic for InternetService.

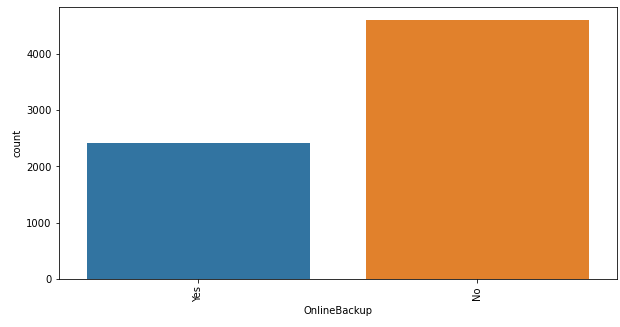


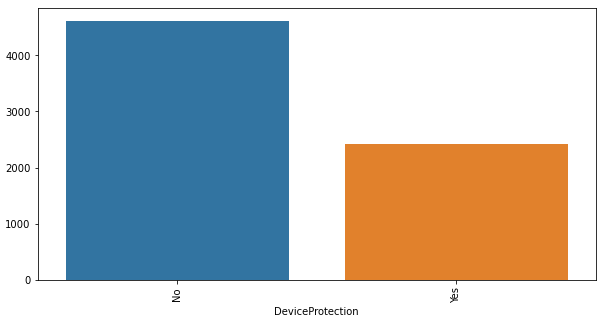


* Majority of service provider donot provide online security and online backup

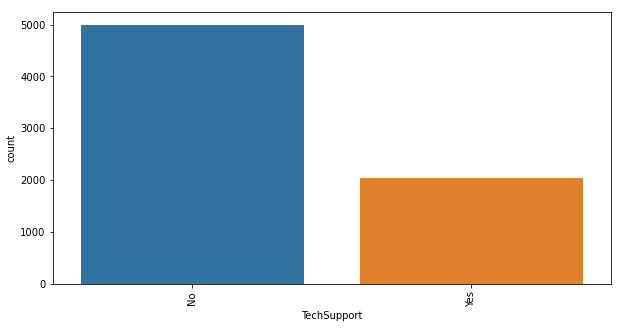


* Majority of Customers doesnot have multiple lines but there are also good amount of customers having multiple lines

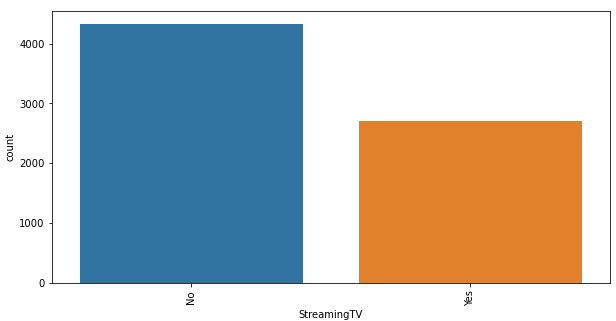


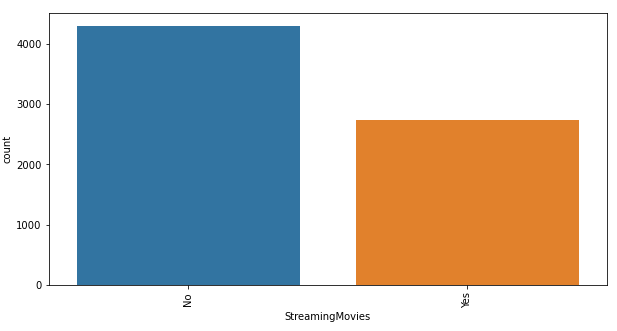


* Majority of service provider donot provide DeviceProtection.

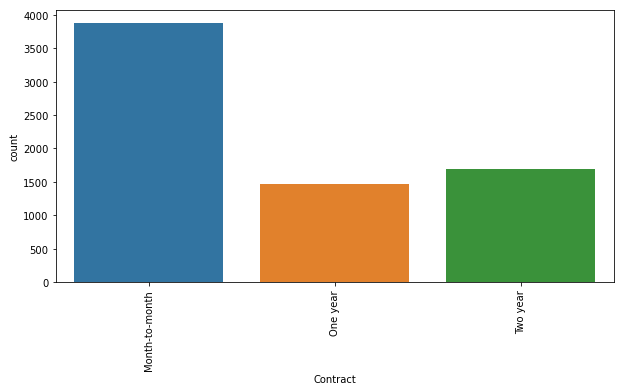


* Majority of service provider donot provide Techsupport

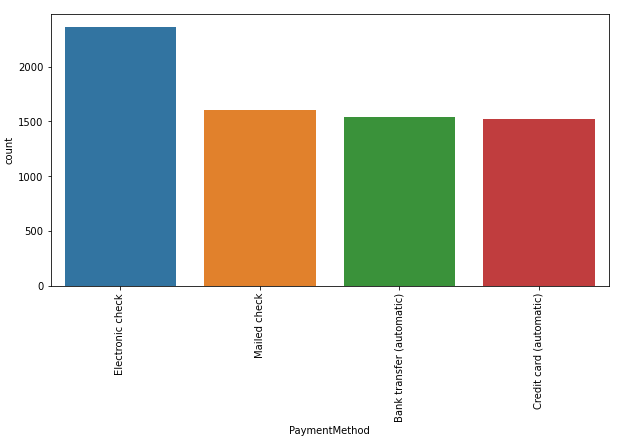




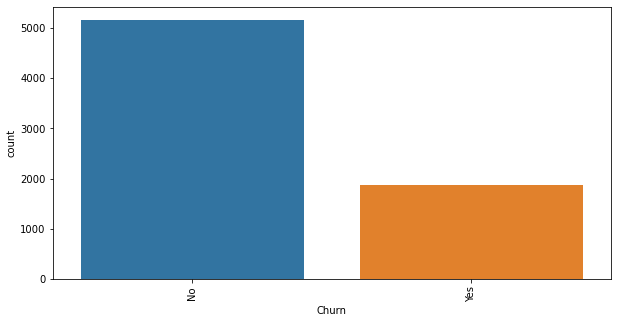
* Majority of service provider doesnot provide streaming services i.e TV or Movies.



* Majority of customer have opted for month to month subscription.

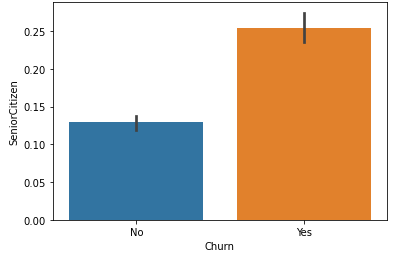


* Majority of customer prefers Electronic Check for payment method.

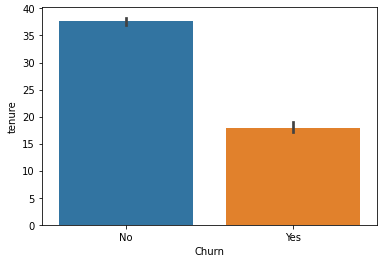


* This is an imbalanced data set because both classes are not equally distributed among all observations, being ‘no’ the majority class . When modeling, this imbalance will lead to a large number of false negatives, we will balance the dataset in the data preprocessing stage.

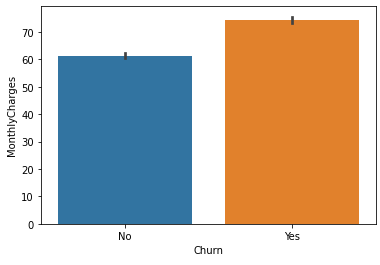
**Bivariate Analysis(Numerical Data-Target)**



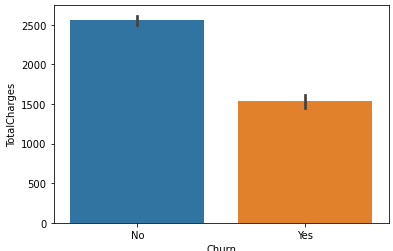
* Senior Citizens have higher churn rate.



* New customers are more likely to churn.

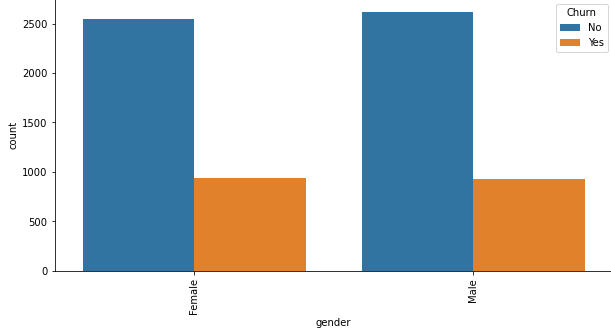


* Customers with higher Monthly Charges are also more likely to churn.

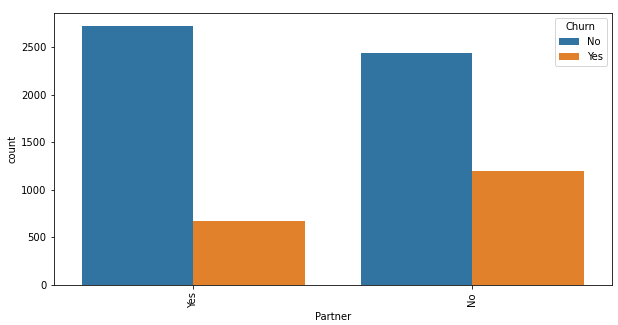


* Clients with high **total charges** are less likely to leave the company.

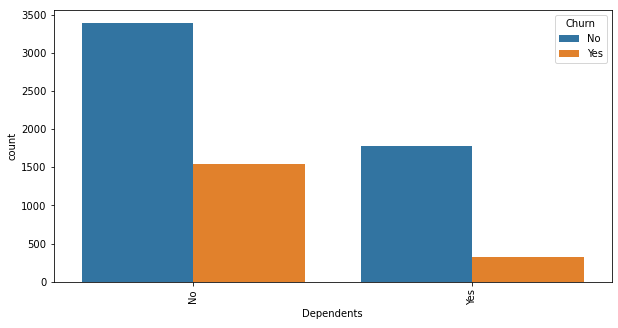
**Bivariate Analysis(Categorical Data-Target)**



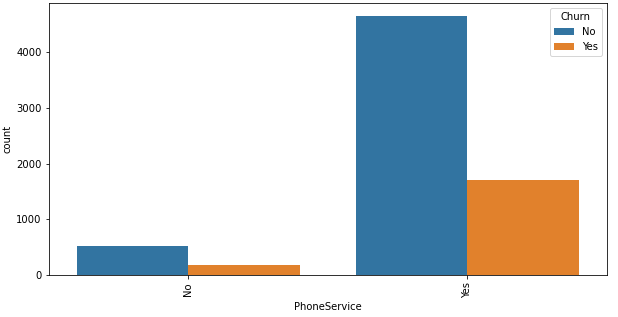
* There is negligible difference in customer percentage/count who changed the service provider. Both genders behaved in similar fashion when it comes to migrating to another service provider/firm.

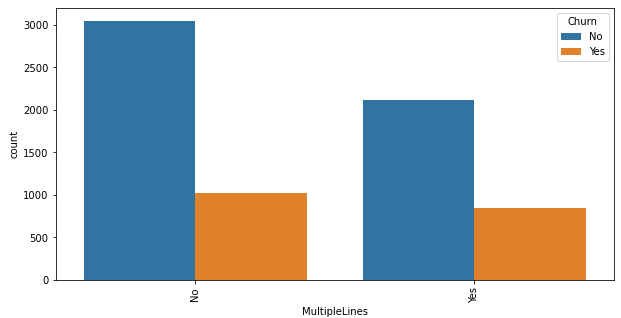


* Customers with a **partner**churn less than customers with no partner.

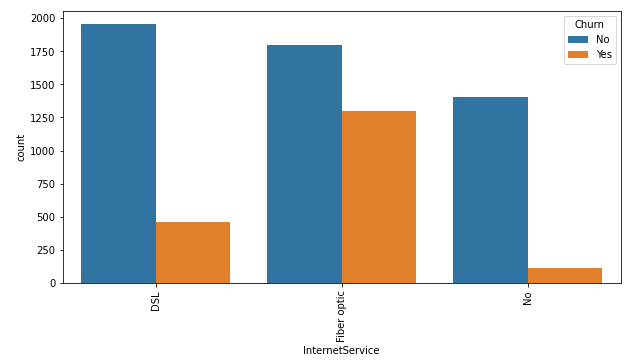


* Customers without dependents are more likely to churn.

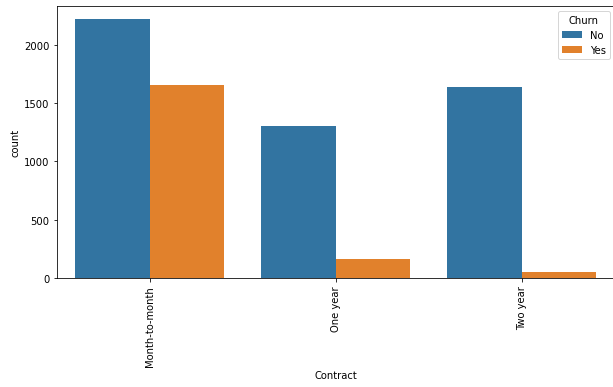




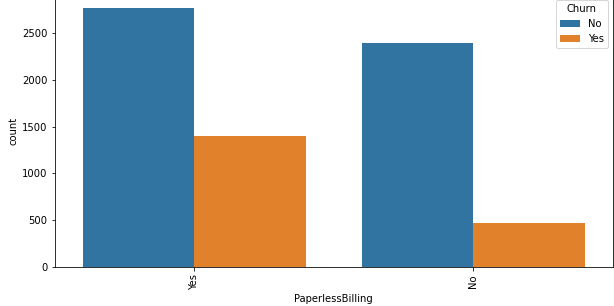
* We do not expect **phone attributes**(PhoneService and MultipleLines)to have significant predictive power. The percentage of churn for all classes in both independent variables is nearly the same.



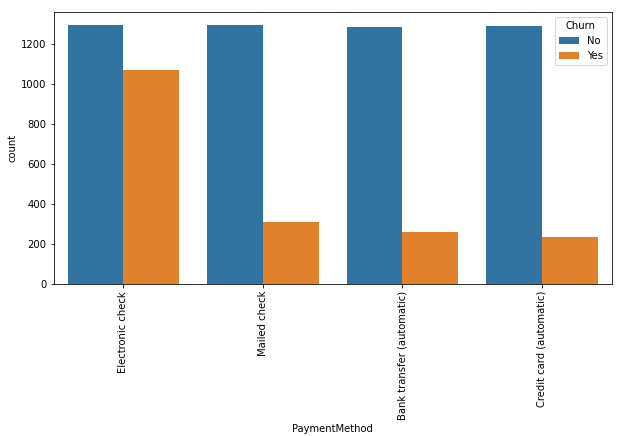
* Customers who use Fiber optic have high churn rate, this might suggest a dissatisfaction with this type of internet service. Customers having DSL service are majority in number and have less churn rate compared to Fibre optic service.



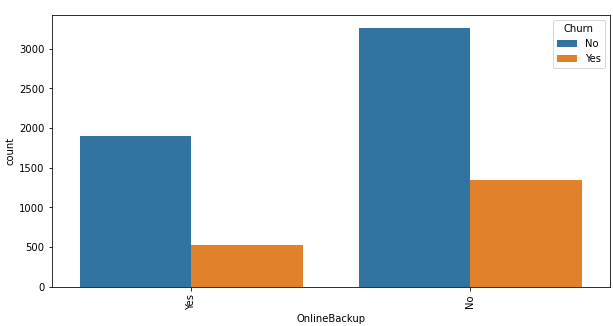
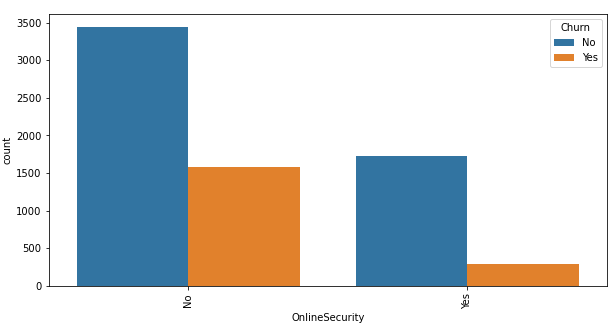
* Customers with **month-to-month contracts** have **higher churn rates** compared to clients with **yearly contracts**.

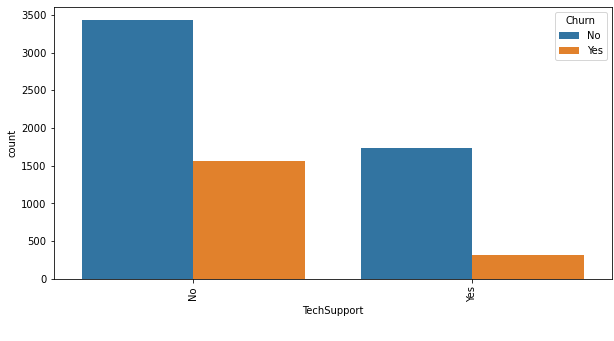
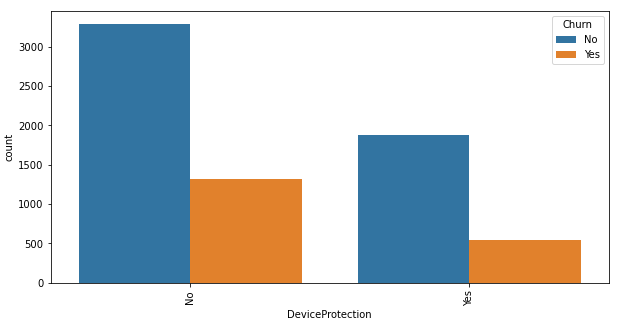


* Customers subscribed to **paperless billing** churn more than those who are not subscribed.



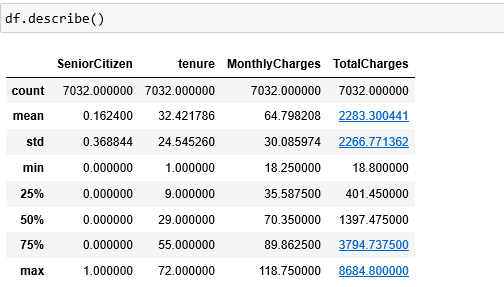
* Customers who opted for an **electronic check** as paying method are more likely to leave the company.





* Clients with **online security,online backup**churn less than those without it.
* Customers with no **tech support and no Device protection** tend to churn more often than those with tech support.

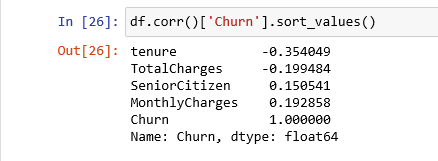
Next we check the numerical statistics of our data using df.describe()

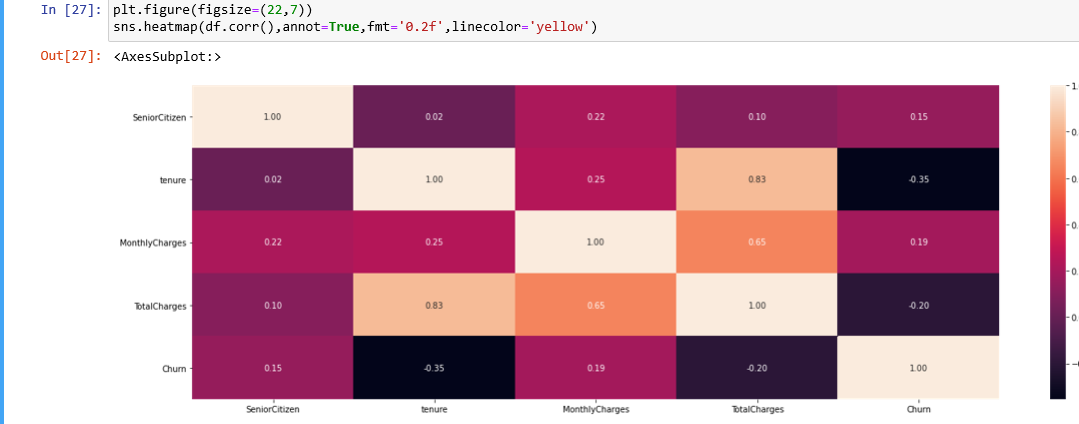


Following observations are made in this step –

* Except TotalCharges(Target) , the mean and median are very close indicating kind of normally distributed or having very less skewness.
* Maximum tenure is 72 and minimum 1.
* Maximum MonthlyCharges is 118.75 and minimum is 18.25.

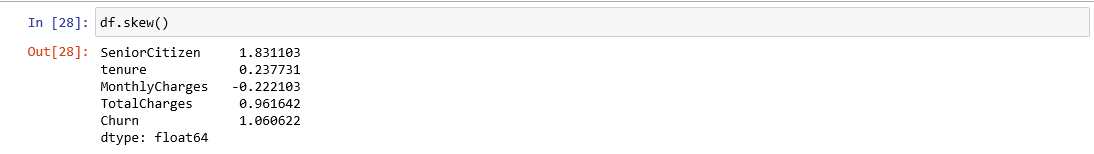
If we check the correlation between the numeric columns and target, we observe that –



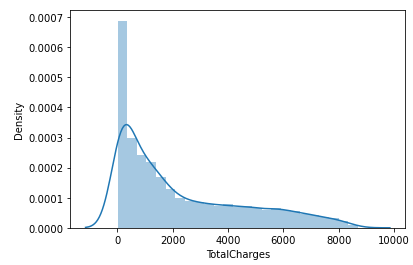


* Monthly Charges is most positively correlated with target i.e increase in monthly charges will increase the churn rate.
* Tenure is most negatively correlated with target i.e increase tenure will decrease the churn rate.

Next we check if any skewness is present in our dataset using df.skew()-

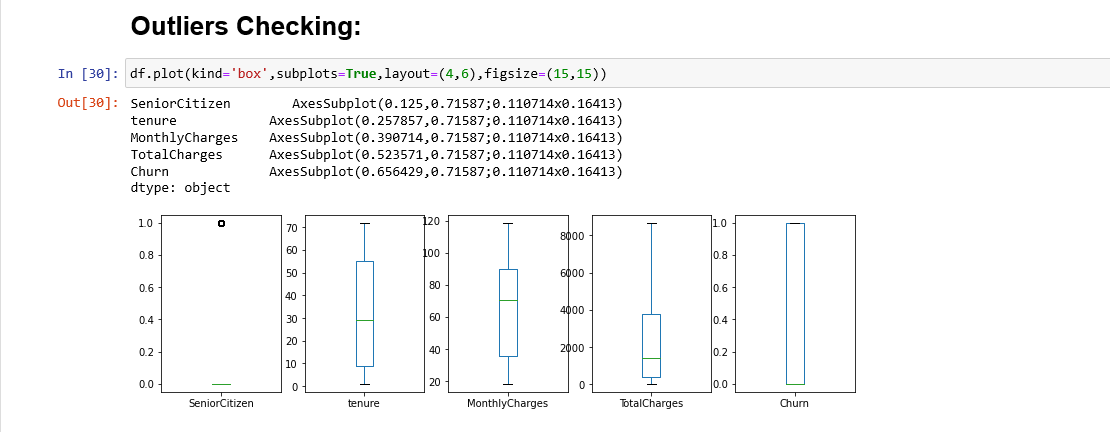


We observe that column TotalCharges is highly skewed. Senior citizen and Churn(Target) are discrete data. Monthly Charges and tenure have very less skewness present. Let’s see the distribution plot of Total Charges.



# Outlier Detection

We further proceed to detect outliers in our data and decide how to deal with them. The best way to interpret outliers using visualizations is boxplot, hence we plot boxplots for our numerical columns, which result in the below visuatization

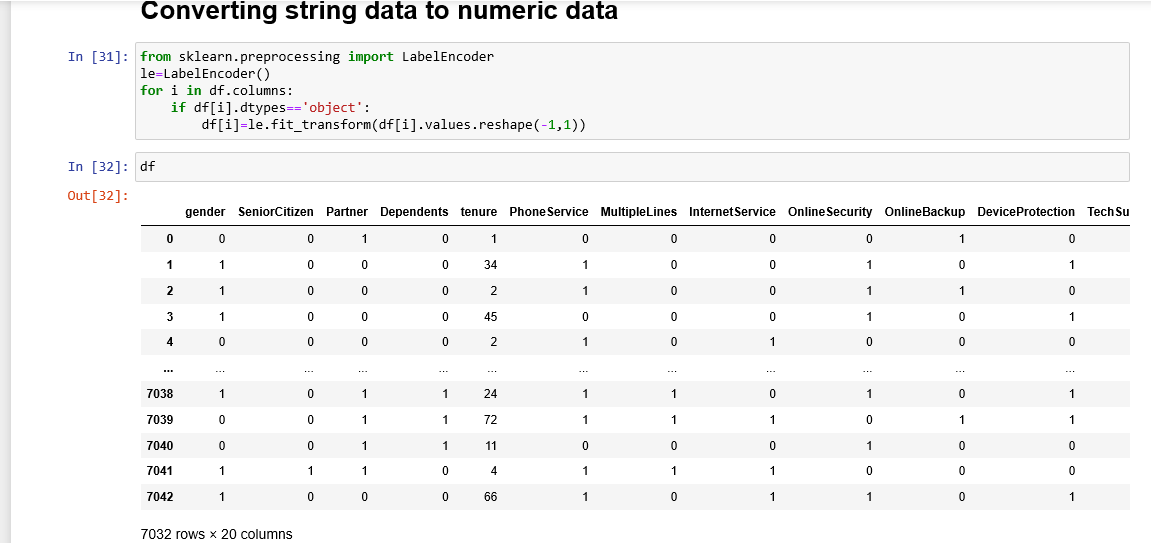


We observe the dataset is free from Outliers.

**4. Data Preprocessing & Feature Engineering:**

# Encoding the data

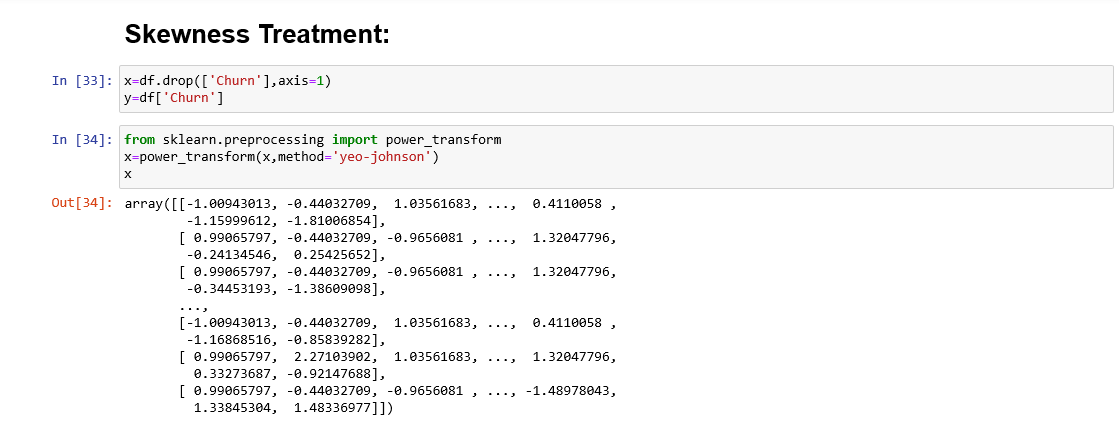
Since majority of the classification models need input as ‘int/float’, and do not work on ‘string’ data, we encode our categorical columns using ‘Label Encoder’



**Skewness treatment**

We now proceed with treating skewness in our data, which allows us to fit our data in a symmetric distribution, which further allows our model to learn better.

We treat it using power transform method and final skewness looks something like this-

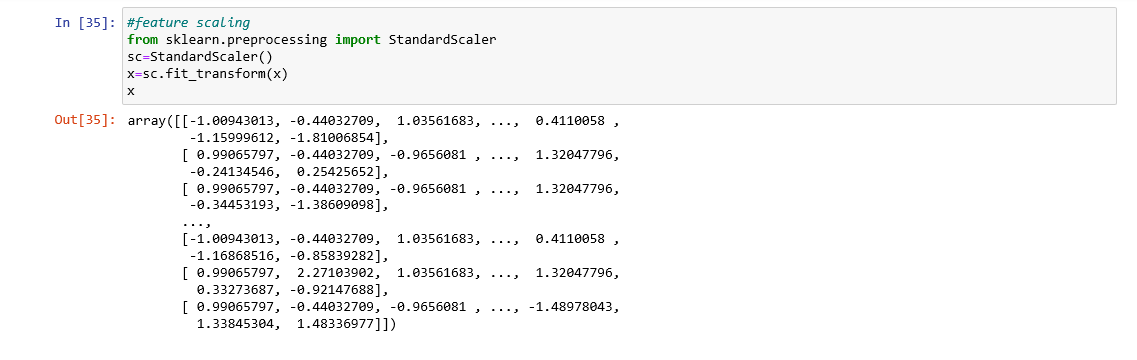


# Scaling the data

The next step is to bring the data to a common scale, since there are certain columns with very small values and some columns with high values. This process is important as values on a similar scale allow the model to learn better.

We use standard scaler for this process –

**‘**StandardScaler follows Standard Normal Distribution (SND). Therefore, it makes mean = 0 and scales the data to unit variance’

**Balancing the data**

As discussed above our target contains imbalanced data. Now we balance it with SMOTE technique.

*‘SMOTE is an oversampling technique where the synthetic samples are generated for the minority class. This algorithm helps to overcome the overfitting problem posed by random oversampling. It focuses on the feature space to generate new instances with the help of interpolation between the positive instances that lie together’.*

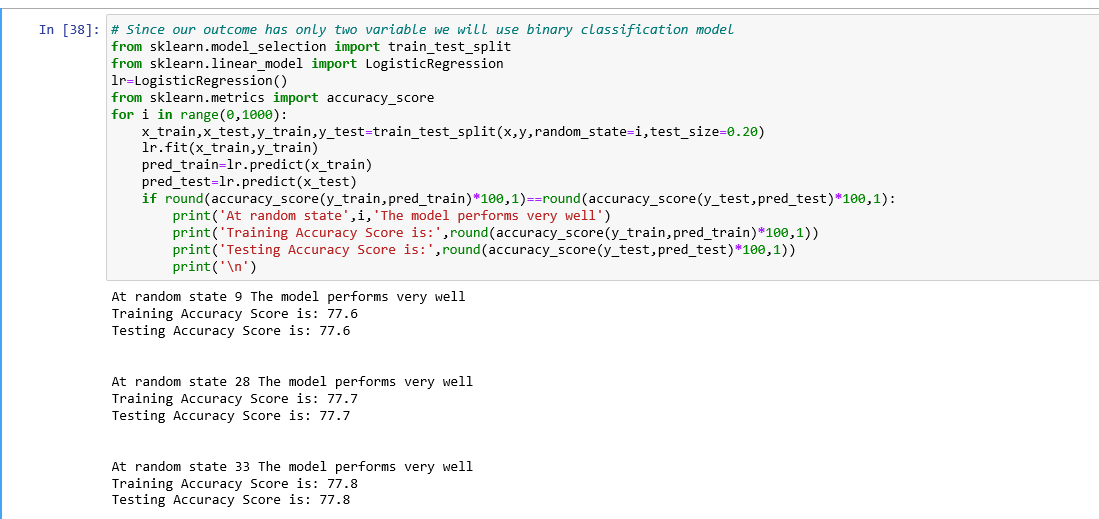


# 5. Model Building

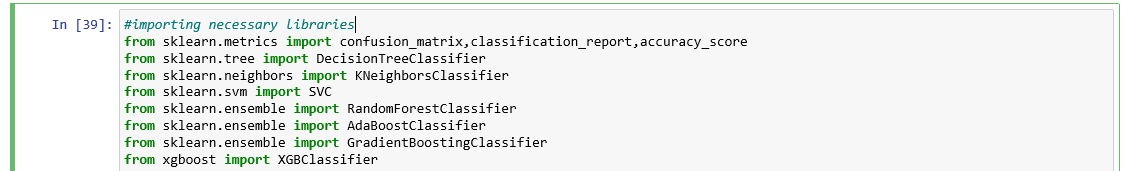
# Fitting data into classification models

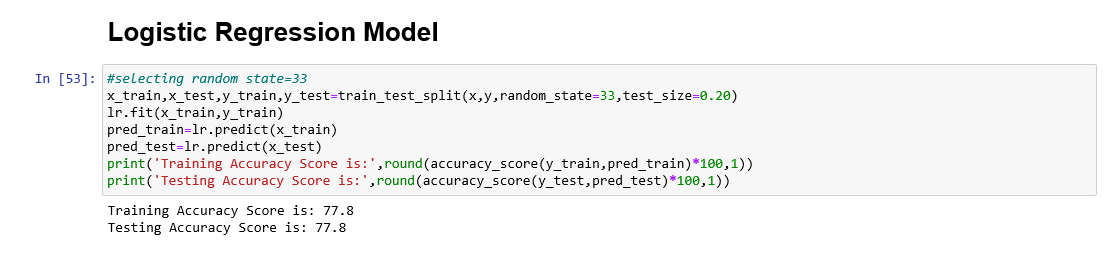
We now proceed to the main step of our machine learning, fitting the model and predicting the outputs. We fit the data into multiple classification models to compare the performance of all models and select the best model –

Our first step is to find out the best random state at which the training and testing accuracy scores are almost equal. Since our target has binary we proceed with the logistic classification model.



At random state equals 33 we see that model is giving highest accuracy scores. So we will build our model with this random state.

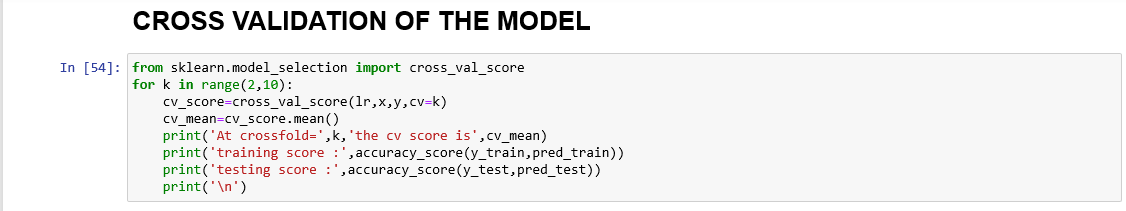


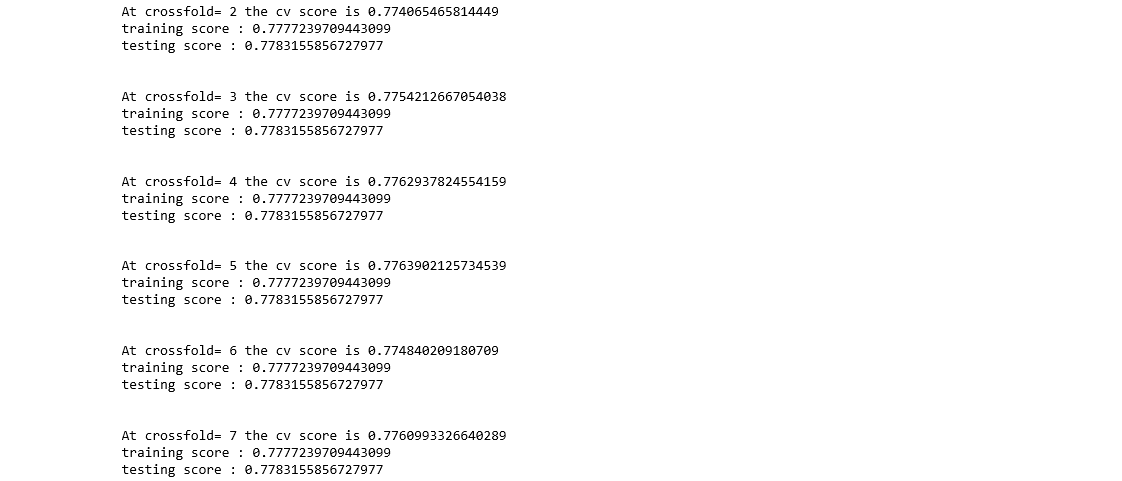


**Cross validation**

The **goal of cross**-**validation** is to test the model’s ability to predict new data that was not used in estimating it, in order to flag problems like overfitting or selection bias and to give an insight on how the model will generalize to an independent dataset (i.e., an unknown dataset, for instance from a real problem).

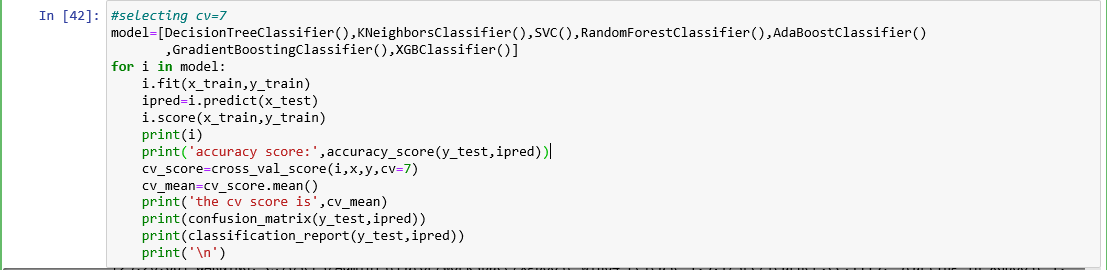
We obtain the following results using cross-validation –



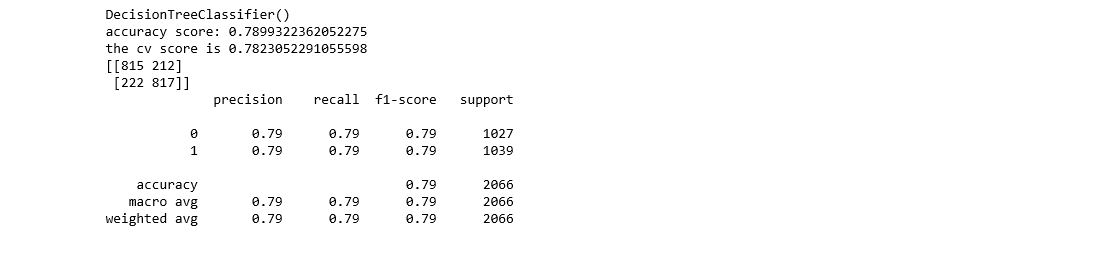


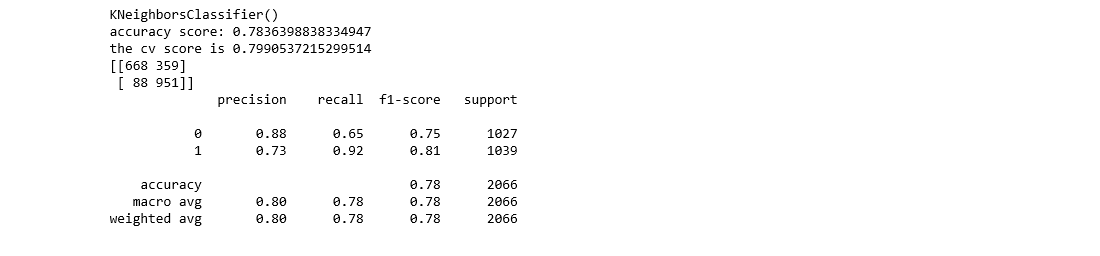
This helps us interpret that the model is not overfitting and will perform well for new data that we feed to our model.

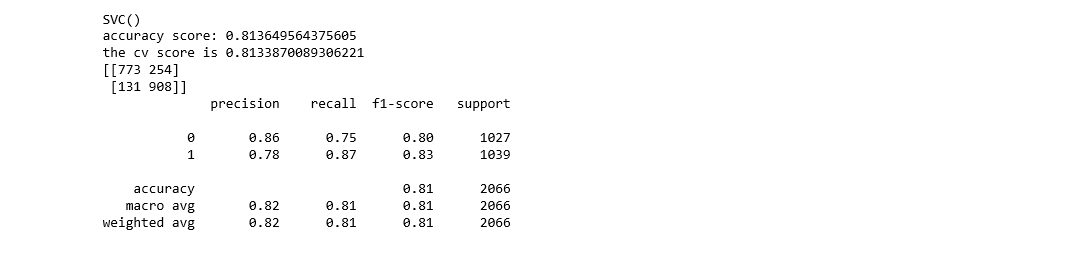
We use the below mentioned code snipped to fit the data into ML models and predict the output –

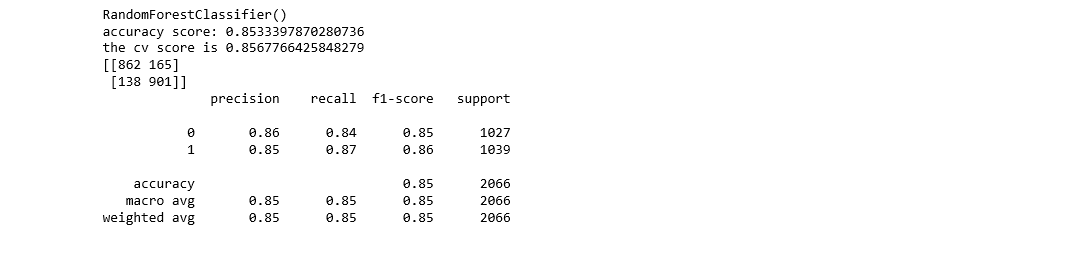


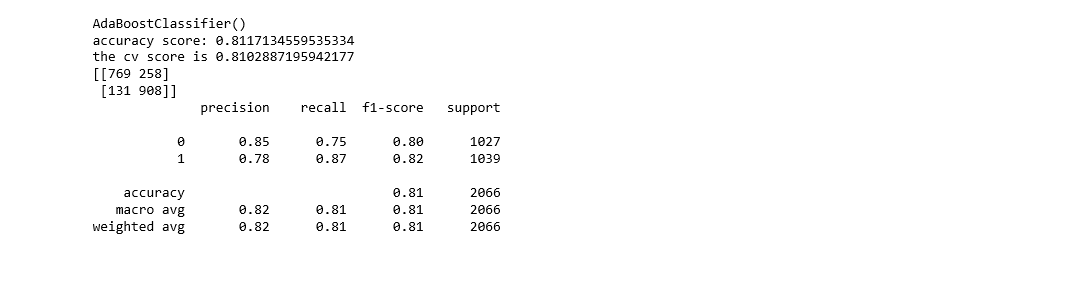
Following outputs are obtained-

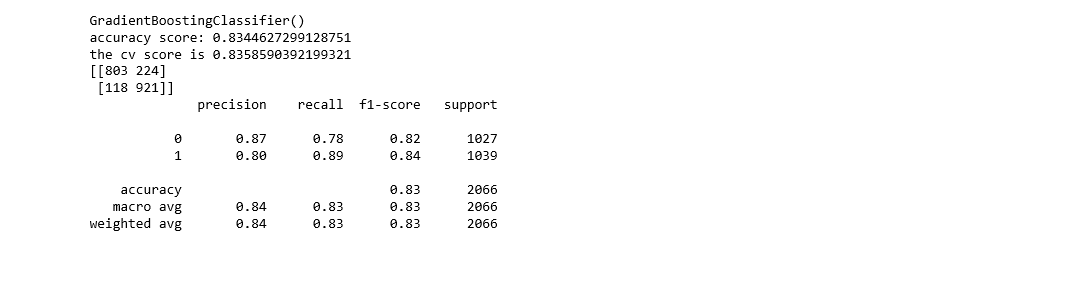


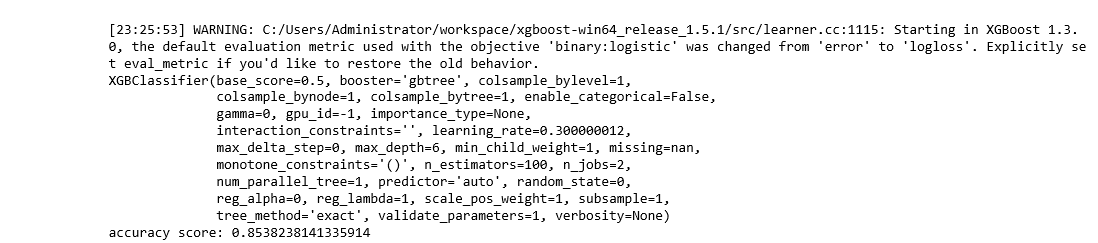


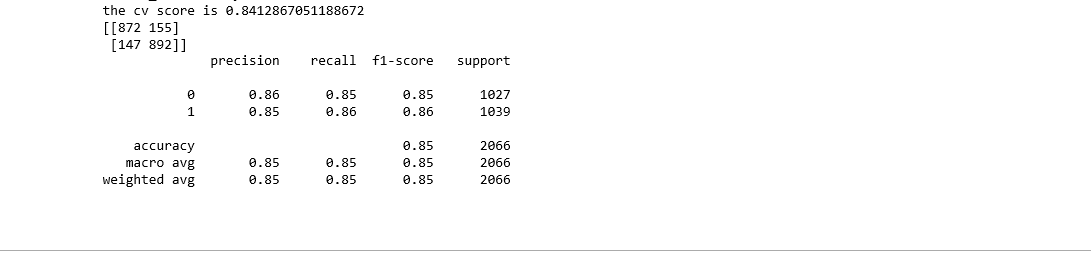






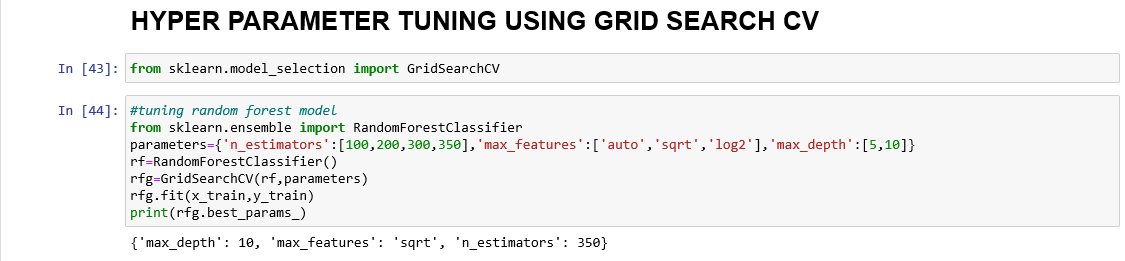


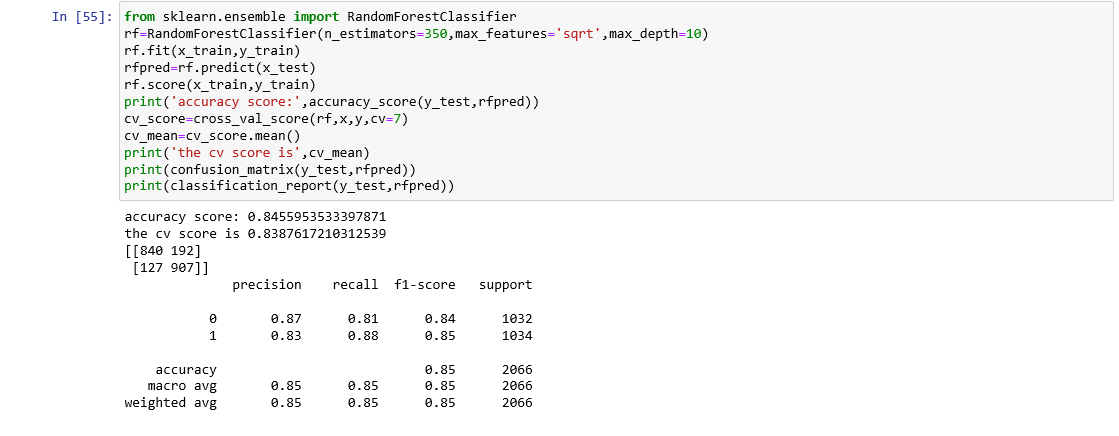


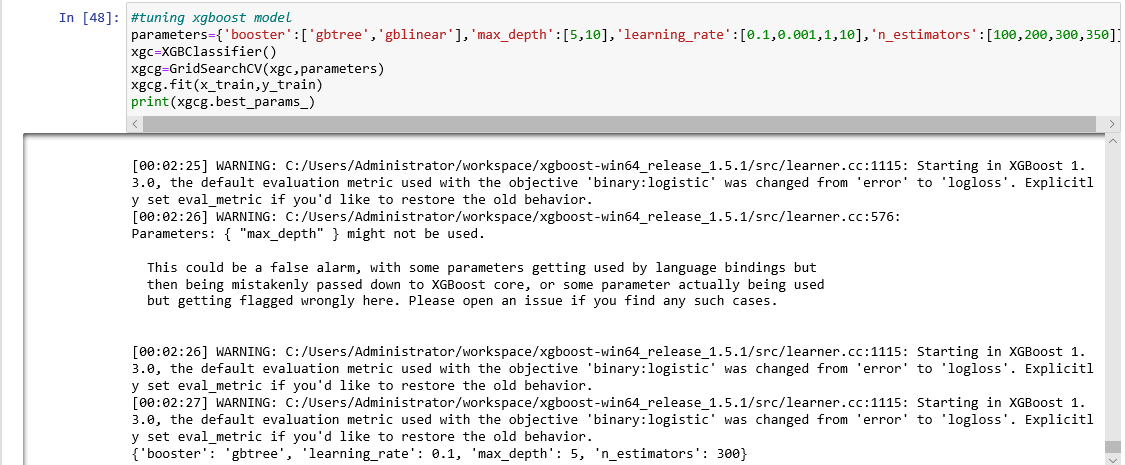


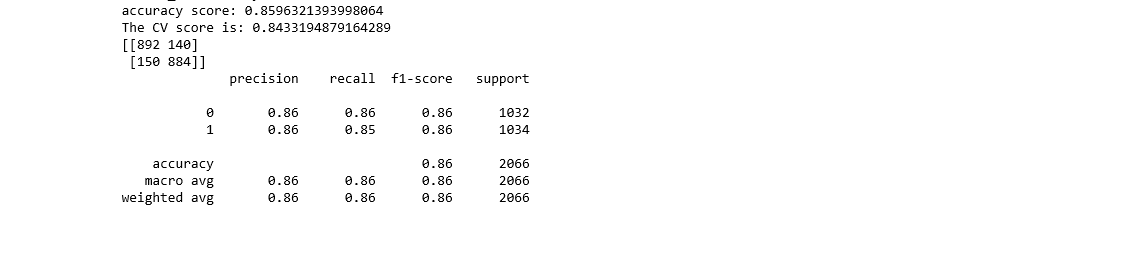
We see that the RandomForest Classifier and XGBoosting Classifier gives us an accuracy of ~85% (higher than SVC,logistic,Decision Tree,Kneighbors,GradientBoosting,Adaboost Classifier), and the CV score also improve. **Hence we choose ‘XGboosting classifier’ and ‘RandomForestClassifier’ as our final model**, and proceed with hyper parameter tuning the model.

**Hyper Parameter tuning the model**

****

****



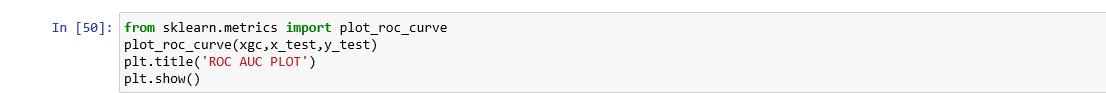


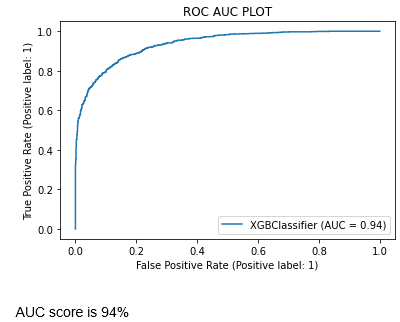
Comparing the performance metrics we select XGBoosting Classifier as our final model. With hyperparameter we observed there is improvement in model performance. We increased our model accuracy by 1% using hyper parameter tuning.

# AUC ROC curve

AUC — ROC curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting 0’s as 0’s and 1’s as 1’s.

We draw the AUC-ROC curve to obtain the following output –

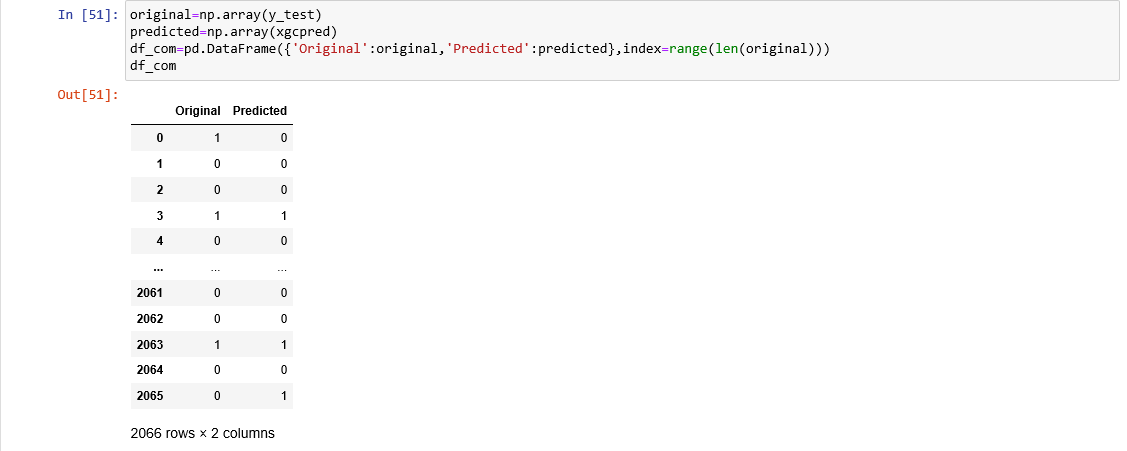




# We obtain an AUC score of 94% with the XGBoost Classifier model.

# Conclusion

We further proceed to test our model, and create a dataframe of predicted values –



**Saving the model**

****

This marks the end of our process; we have successfully trained our model to predict whether a customer will churn or not, with an accuracy of ~86%.

We moved step by step, analyzing, cleaning and modeling the data, and applied various machine learning models to achieve the desired predictions. We also tuned the model to improve the accuracy, and were able to achieve a model with quite a good accuracy.