# **Customer Churn Analysis using Machine Learning**

Customer retention is critical to a company's success. Customers are a company's most precious asset, impacting business today and growing in value over time as they continue to engage in products and services. Customer churn can be costly for both new and established businesses. The true cost of churn is much larger than what corporate leaders assume. Not only does this result in lost revenue in the short term, but it also means that your team will have to double down on gaining new customers to replenish those revenue streams in order to assure future success. It is widely acknowledged that acquiring a new customer might cost up to 5 times as much as keeping an existing one.

**Problem Definition**

According to the introduction, the main challenge is predicting whether or not a specific customer would churn. Machine learning models are trained based on 80% of the sample data to achieve this. The remaining 20% is utilised to put the trained models to work and evaluate their prediction power in terms of "churn / not churn." Which features actually promote customer churn will be a separate question. This data can be utilised to detect client "pain points" and address them by providing incentives to keep customers coming back.

**Understanding Features:**

A general idea of the features can be gained by looking at the columns and their unique values. The features can also be classified into several groups for further evaluation:

* Churn: Weather the customer stopped business or not (Yes/No)

**Customer Services:**

* PhoneService: Weather the customer using phone service (Yes/ No)
* MultipleLines: Weather the customer has multiple lines (Yes/ No/ No phone service)
* InternetService: Customer internet services provider (DSL/ Fiber optic/ No)
* OnlineSecurity: Weather the customer has online security (Yes/ No/ No internet service)
* OnlineBackup: Weather the customer has online backup (Yes/ No/ No internet service)
* DeviceProtection: Weather the customer has drive protection (Yes/ No/ No internet service)
* TechSupport: Weather the customer has tech support (Yes/ No/ No internet service)
* StreamingTV: Weather the customer has streaming TV (Yes/ No/ No internet service)
* StreamingMovies: Weather the customer streaming movies (Yes/ No/ No internet service)

**Customer account:**

* Tenure: Number of months a customer stayed with company
* Contract: Contractual term of customer (month to month, one year, two year)
* PaperlessBilling: Weather the customer using paperless billing (Yes/ No)
* PaymentMethod: paying mode of customer (Electronic check, mailed check, bank transfer (automatic), credit card (automatic)
* MonthlyCharges: customer pays monthly amount.
* TotalCharges: total charges paid by customer

**Customer profile:**

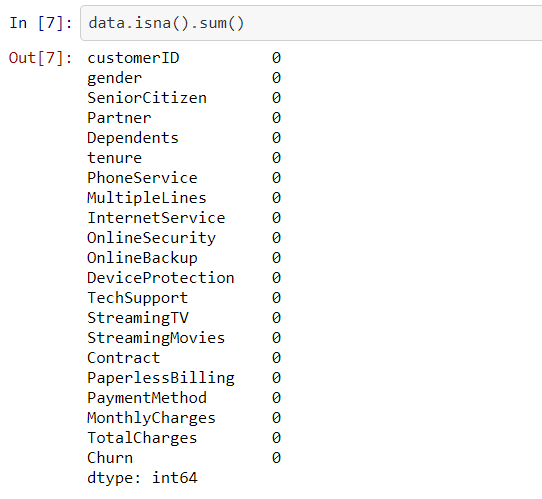
* Customer ID: unique customer identification number
* Gender: Either the customer is Male or Female
* SeniorCitizen: Weather the customer is Senior Citizen (Yes/ No)
* Partner: Weather the customer has partner (Yes/ No)
* Dependents: Weather the customer has dependents (Yes/ No)

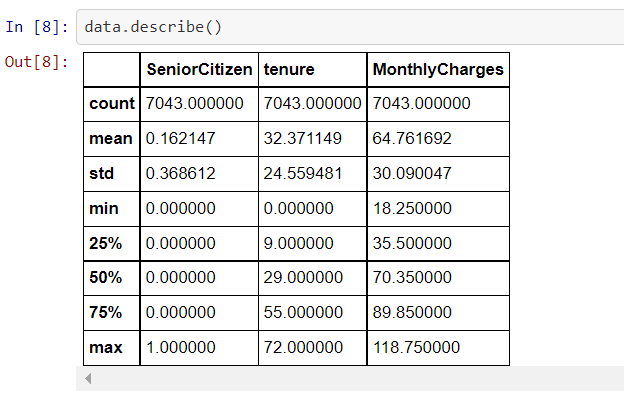
**Libraries to import:**

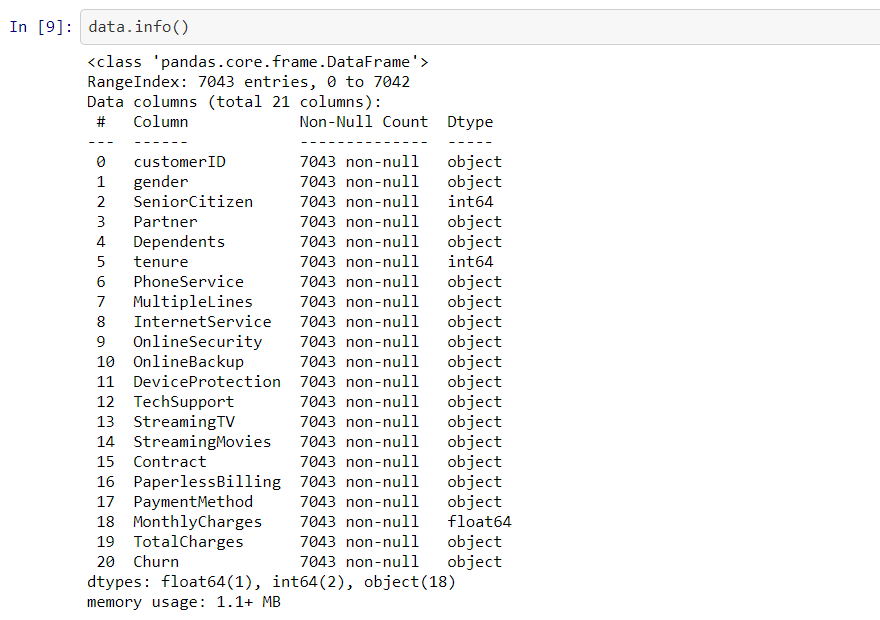
* import numpy as np
* import pandas as pd
* import seaborn as sns
* import matplotlib.pyplot as plt
* import warnings
* from sklearn.preprocessing import OrdinalEncoder, power\_transform, StandardScaler, MinMaxScaler
* from sklearn.model\_selection import train\_test\_split, cross\_val\_score, GridSearchCV, KFold, StratifiedKFold
* import scikitplot as skplt
* import sys
* import six
* sys.modules['sklearn.externals.six']=six
* import mlrose
* from yellowbrick.classifier.rocauc import roc\_auc
* from kmeans\_smote import KMeansSMOTE
* from sklearn.metrics import confusion\_matrix,classification\_report,accuracy\_score
* import pyfiglet
* import chart\_studio.plotly as py
* import plotly.graph\_objects as go
* import plotly.tools as tls
* from plotly.offline import iplot,init\_notebook\_mode
* import cufflinks as cf
* import plotly.figure\_factory as ff
* from sklearn.ensemble import ExtraTreesClassifier,RandomForestClassifier
* from sklearn.linear\_model import LogisticRegression
* from sklearn.tree import DecisionTreeClassifier
* from xgboost import XGBClassifier
* from lightgbm import LGBMClassifier

**Exploratory Data Analysis**

The data frame is analysed for structure, columns provided, and data types in the first step of EDA. The objectives of this step are to gain a general overview of the data set, evaluate domain knowledge, and understand firstly columns to deal with. Some common Pandas functions are applied in this step:







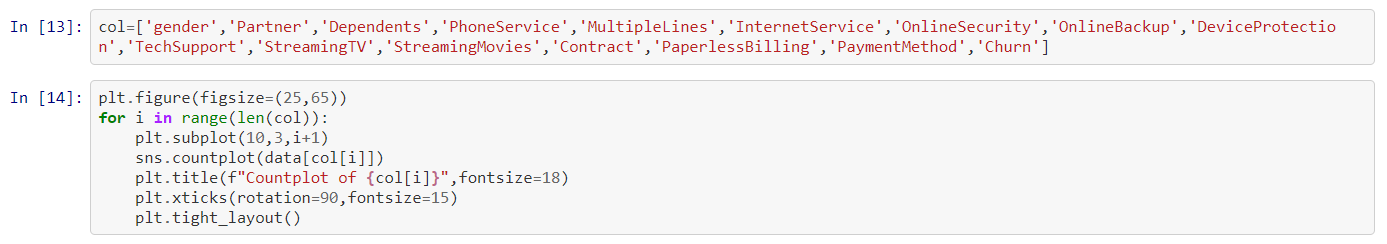
Selecting all the object data types for further use:

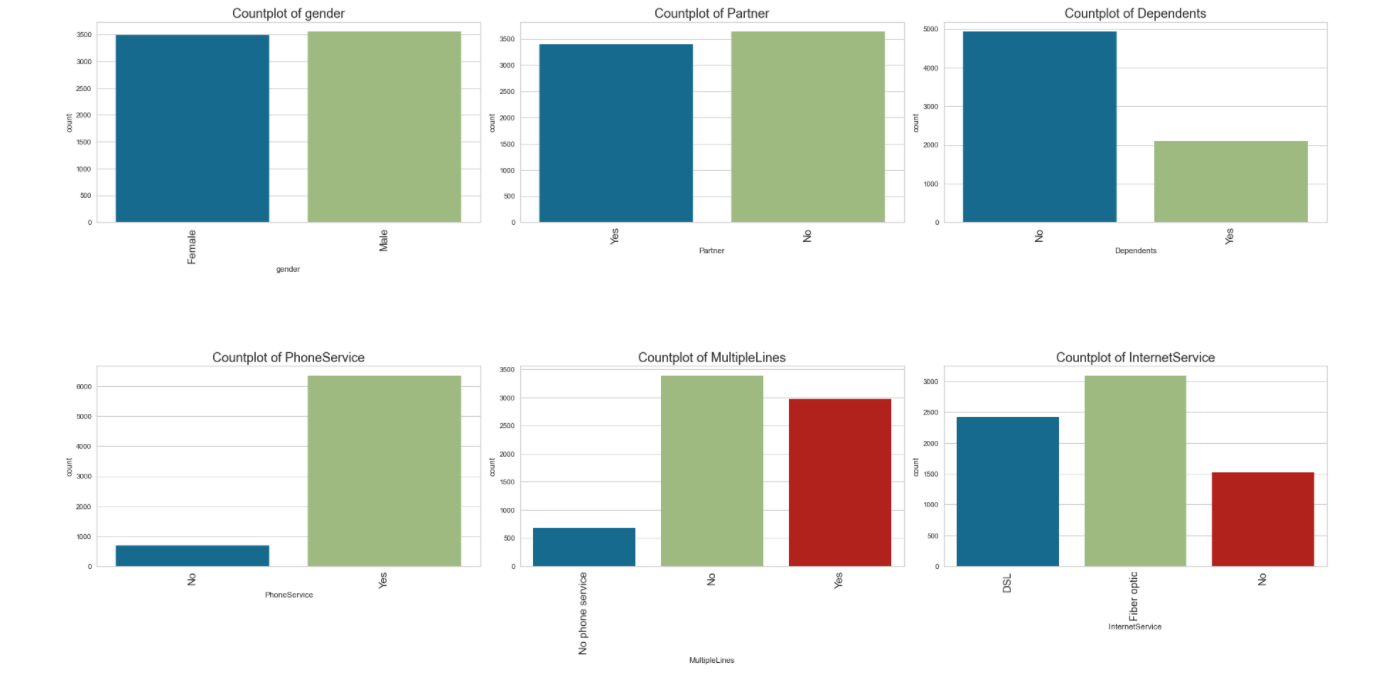


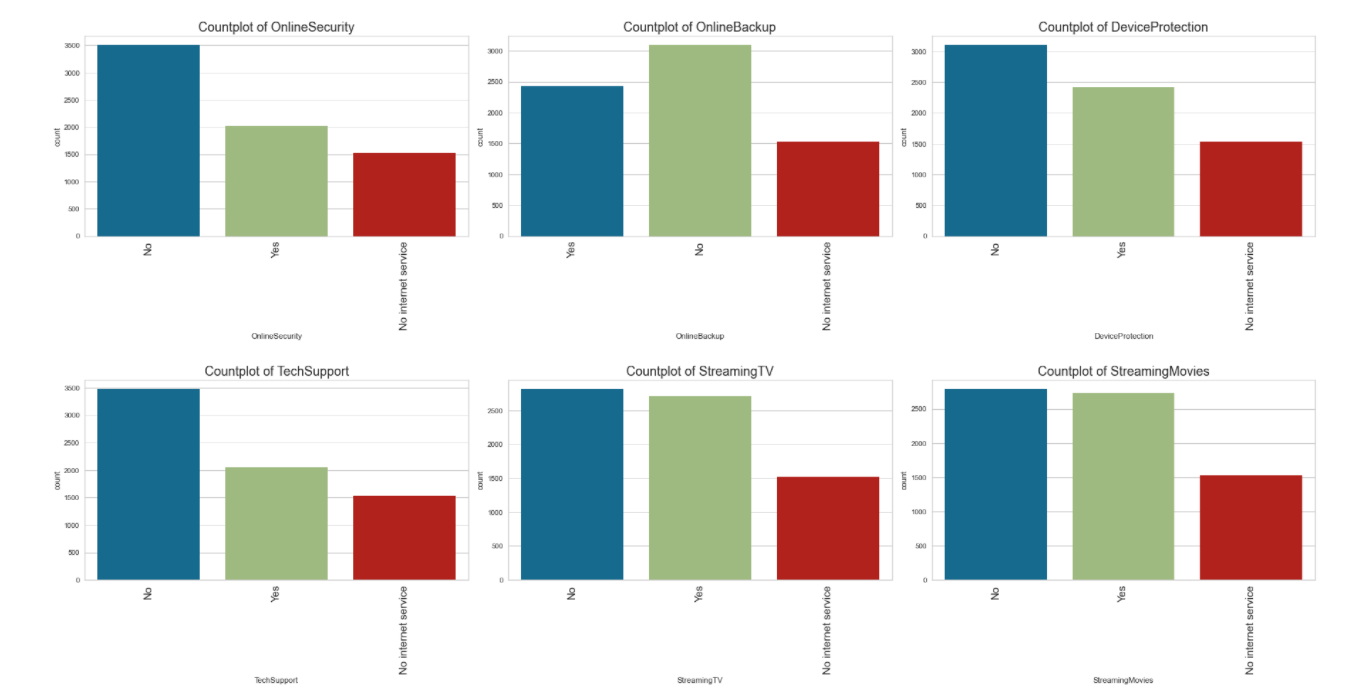
TotalCharges are in object data type, we need to change:

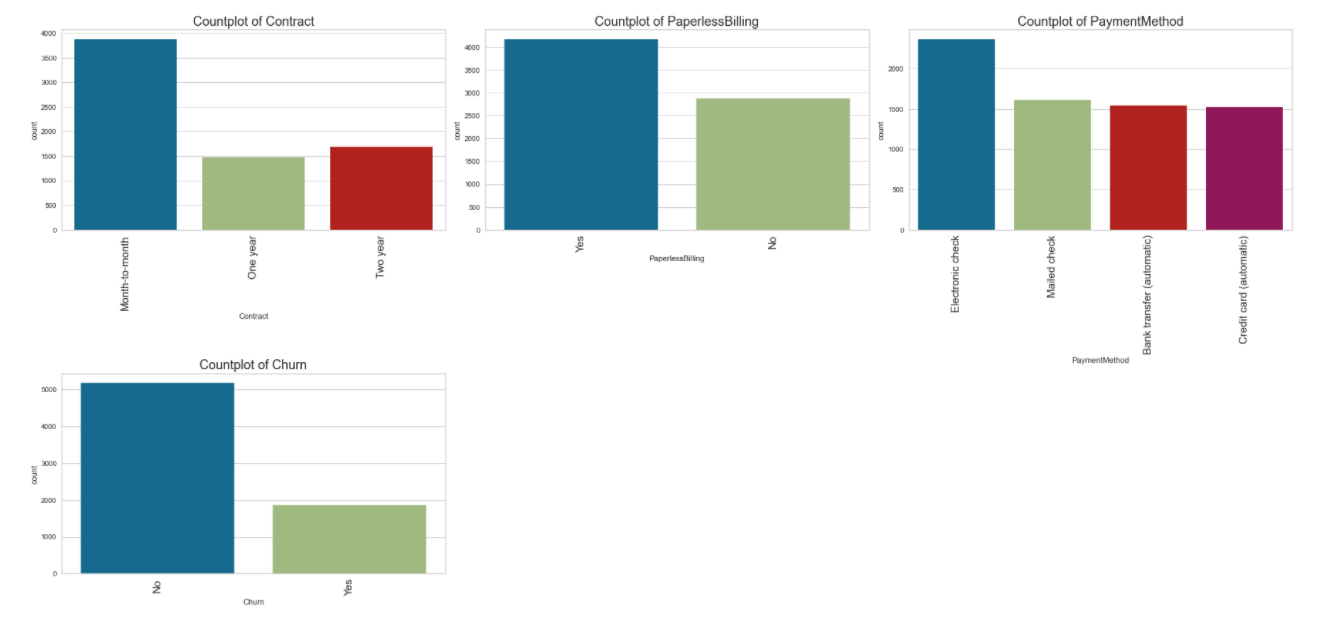


**Univariate Analysis for categorical columns:**





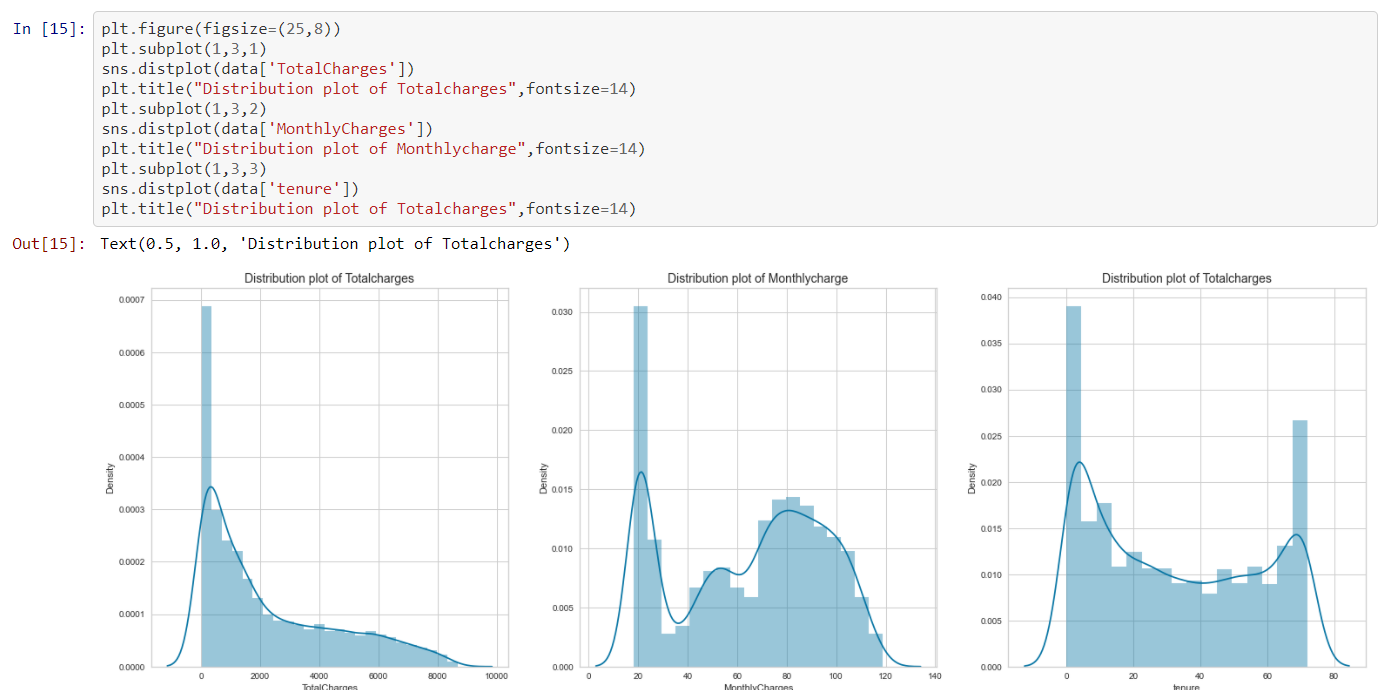




Above we have done univariate analysis of categorical columns and observations are:

* Gender: Number of Males is higher than Females.
* Partner: Most of the customers don’t have a partner.
* Dependence: Most customers are independent - getting a minimum dependent customer.
* Phone Service: Mostly customers using phone services.
* MultipleLine: Mostly customers using phone service but not multiple lines - few customers don’t have phone service even.
* InternetService: Mostly customers using Fiber optics rather than DSL.
* OnlineSecurity: Maximum customers don’t have online security.
* OnlineBackup: Maximum customers don’t have online backup.
* DeviceProtection: Maximum customers don’t have device protection.
* TechSupport: Maximum customers don’t have tech support.
* StreamingTV: Maximum customers don’t stream TV.
* StreamingMovies: Maximum customers don’t stream movies.
* Contract: Most customers have month to month contracts.
* PaperlessBilling: Maximum customers do paperless billing.
* PaymentMethod: Mostly use Electronic check then mailed check then bank transfer and credit card.
* Churn: this data shows imbalance which needs to be balanced for most accurate results. We will do it later in the process using SMOTE.

**Univariate Analysis for continuous columns:**

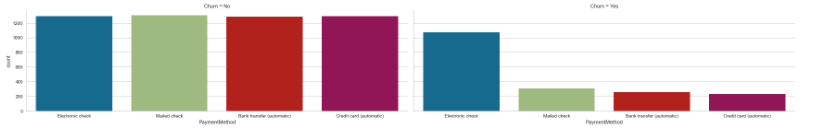
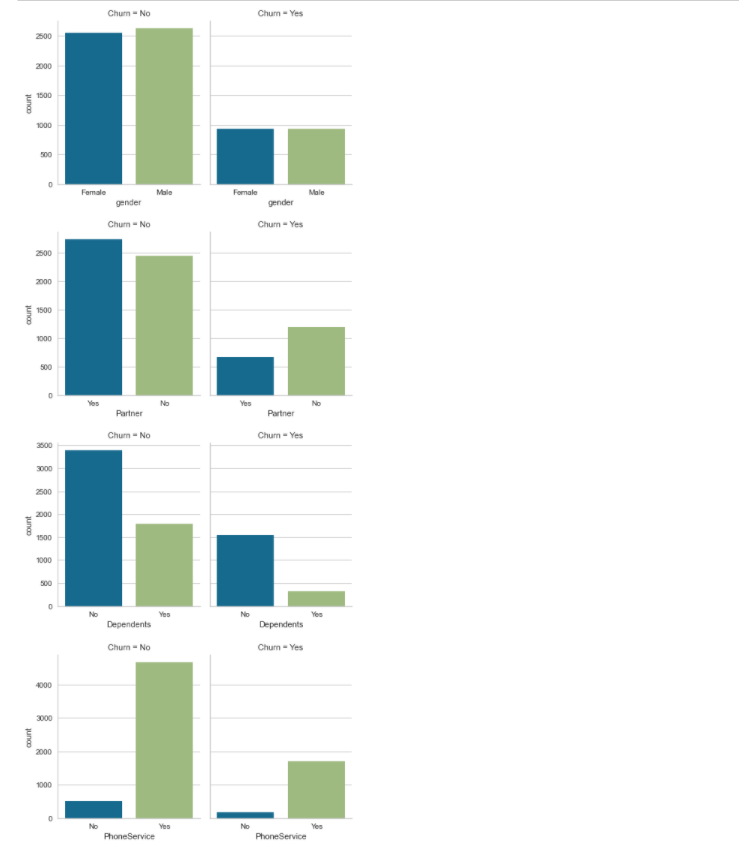


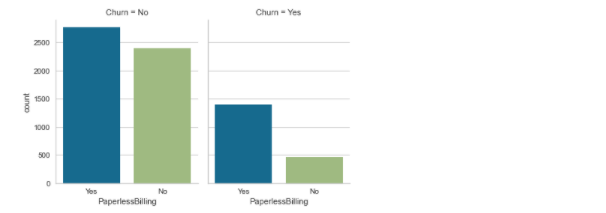
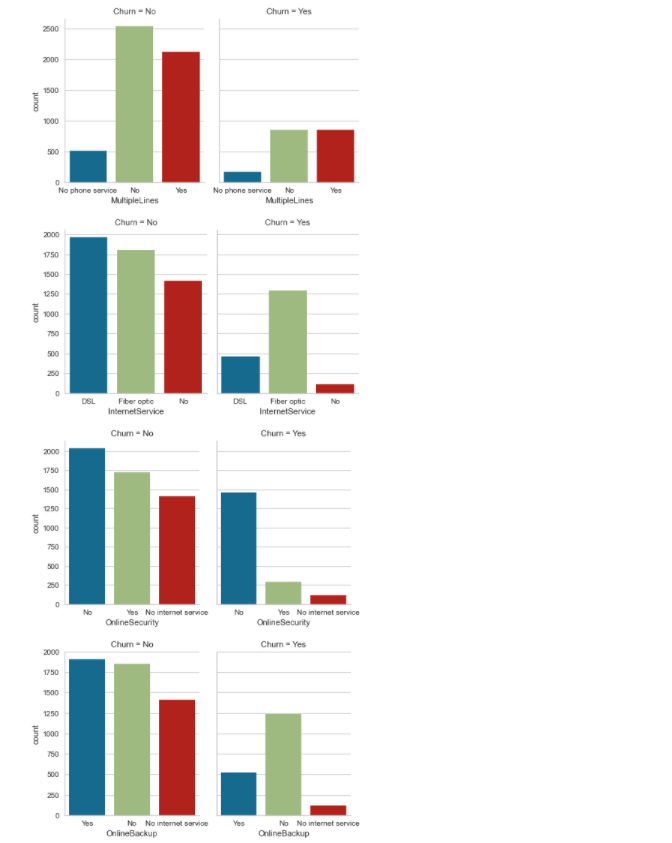
Above we have done univariate analysis of continuous columns and observations are:

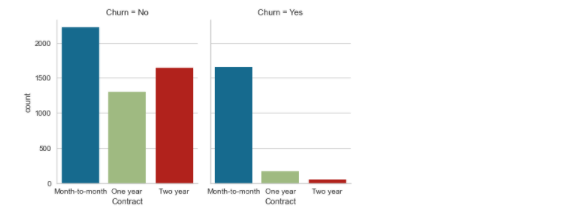
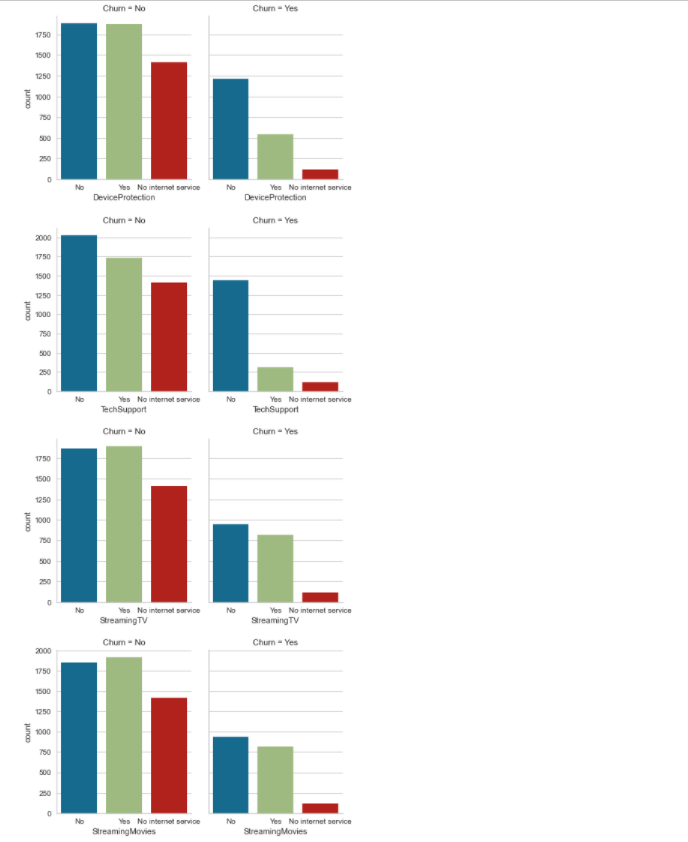
* TotalCharges:
  + Data is positively skewed.
  + Majority of the population spent close to $1,100 dollars
  + Customer have spent upto $8000 dollars
* MonthlyCharges:
  + The graph is not normal distributed
  + Most of the people spend between 18-24 dollars. Must be the service charge for basic service. Majority of customers have subscribed to the basic package
  + Between 70-100 dollars - there are quite good no. of customers
* Tenure:
  + It’s a Bi-Model distribution having 2 peak, which means data is concentrated across two different group
  + We have a major chunk of customers in the 0-1 month period. lot of them might be customers who tried the service and left or liked the service and continue
  + Between 10-65 months we can see flat distribution of data
  + There are lot of customers in 69-72 months range.They are the royal customers

**Bivariate Analysis:**







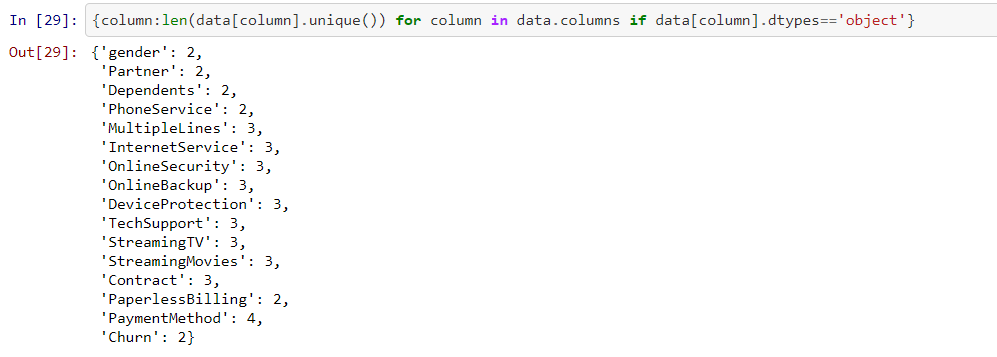


Above we have done bivariate analysis churn column with other columns:

* Male and female both churn values are almost equal.
* Customers with partners having less chance of churn
* Customers with no dependence have high chances of churn.
* Customers having phone service have higher chances of churn.
* From multiple lines - it can be seen that customers with no phone service have less chances to churn whereas customers having multiple lines or not both have the same chances of churn.
* People with fiber optics have higher chances of churn.
* Customers not having online security have higher chances of churn and customers having no internet services are less churn.
* Customers not having online backup have higher chances of churn and customers having no internet services are less churn.
* Customers not having device protection have higher chances of churn and customers having no internet services are less churn.
* Customers not having tech support have higher chances of churn and customers having no internet services are less churn.
* Customers not StreamingTV or StreamingMovies have higher chances of churn and customers having no internet services are less churn.
* Customers who have month to month contracts have higher chances of churn whereas 2years contracted customers having less churn.
* Customers using paperless billing have more chances to churn.
* Customers using electronic checks have higher chances of churn.

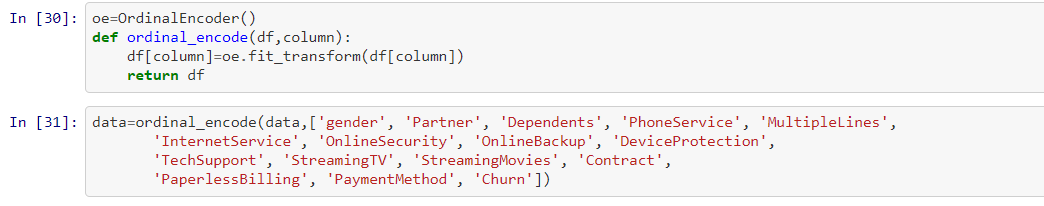
**Pre-Processing of data:**

Checking for unique values in every column for encoding.

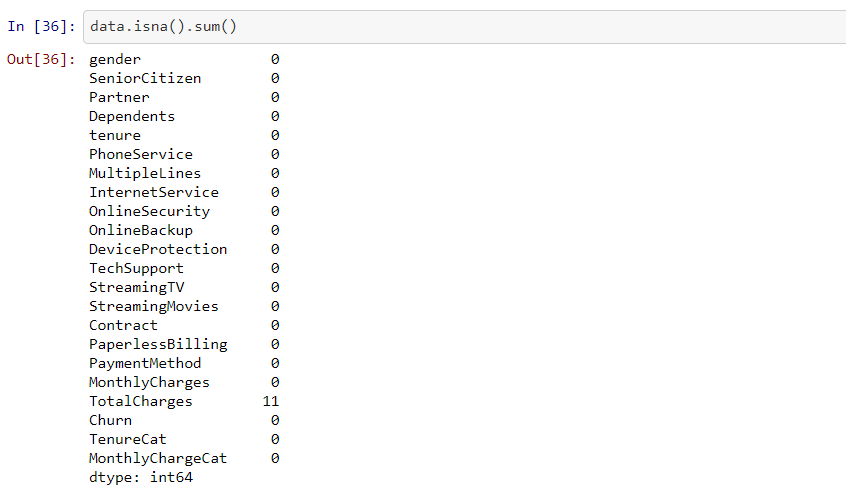


Now doing feature encoding using ordinal encoder:

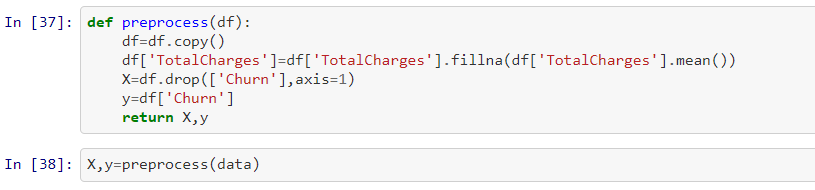
Ordinal Encoder: Each unique label is mapped to an integer value..



Again checking for null values:

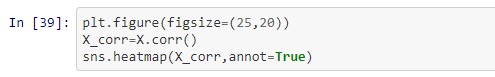


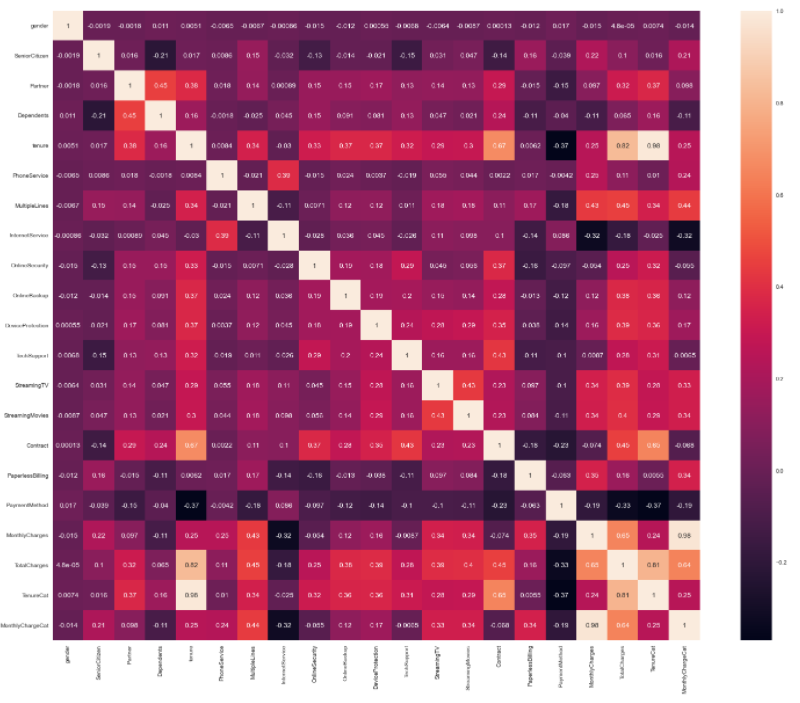
We found that Total charges are having null values as we earlier converted total charges column from object data type to float data type, so we’re putting the mean of the column with below code and separating the dependent and independent variables:



**Correlation Analysis:**

Multivariate Analysis to check the correlation between the columns:

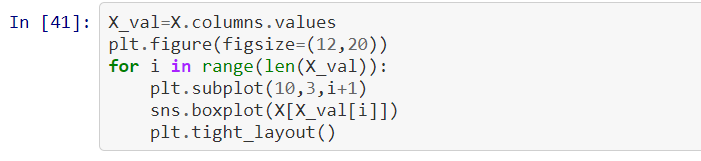




The heatmap shows a good correlation between columns, we don’t have any columns with correlation of more than 90%. So we can move ahead without any problem.

**Check for outliers:**

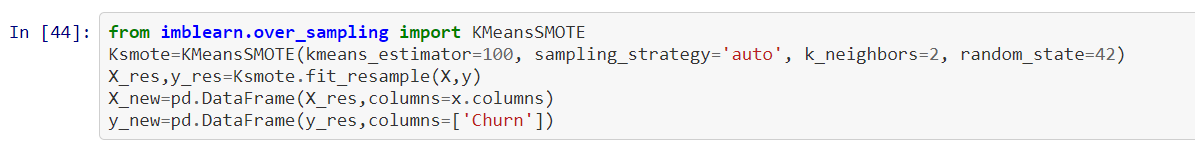
A data item/object that deviates greatly from the rest of the (so-called normal)objects is referred to as an outlier. Errors in measurement or execution can cause them.



Boxplots are plotted to visualize the outliers.

As we can see on the boxplot that there are no outliers, so we are proceeding further with the balancing of data.

**Balancing the data:**

****

Applying KMeansSMOTE here in this case. Why?

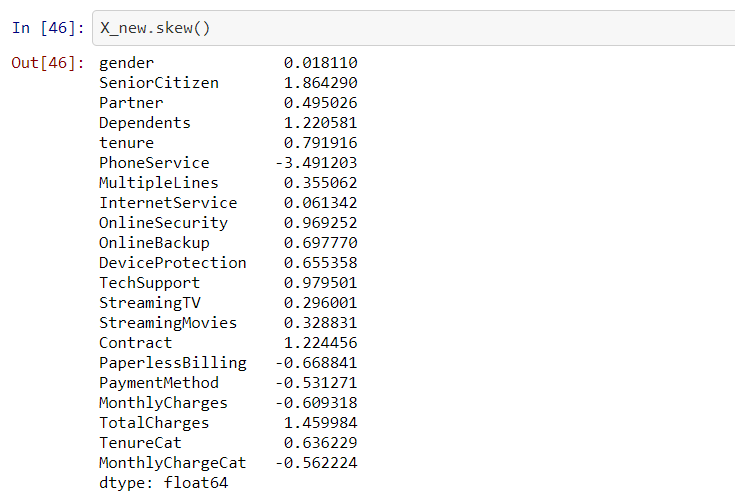
* K-Means SMOTE is a class-imbalanced data oversampling approach. It assists classification by producing minority class samples in safe and critical regions of the input space. The approach reduces noise while resolving imbalances between and within classes.



As we can see in the picture, using KMeansSMOTE - now the data is balanced and we can proceed further to check skewness and to the machine learning process.

**Check Skewness:**

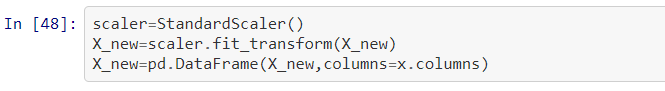
Skewness is a distortion or asymmetry in a set of data that deviates from the symmetrical bell curve, or normal distribution. The curve is said to be skewed if it is displaced to the left or right.



We can apply log transformation to a categorical column only, so apply the same in TotalCharges. We also have a few more columns skewed but those all are categorical, so leaving them as it is and proceeding further.

**Feature Scaling:**

Feature scaling is a technique for putting the data's independent features into a fixed range. It is used to handle significantly changing magnitudes, values, or units during data pre-processing.



We are using standardscaler to scale our complete data.



Splitting our dataset for training and testing with the ratio of 70:30 whereas 70% for training and 30% for testing.

**Training Multiple Models:**

As we know that our target is binary (Yes,No or 0,1) - so we are going to use classification models here and check for accuracy score, cross validation score, confusion matrix, classification report, AUC ROC curve and model learning curve.

We are splitting the training dataset in 5 folds.

We'll start by putting different models to the test and evaluating their performance using a variety of criteria. Later, the hyperparameters of those models will be tuned to improve their performance. The following models were used:

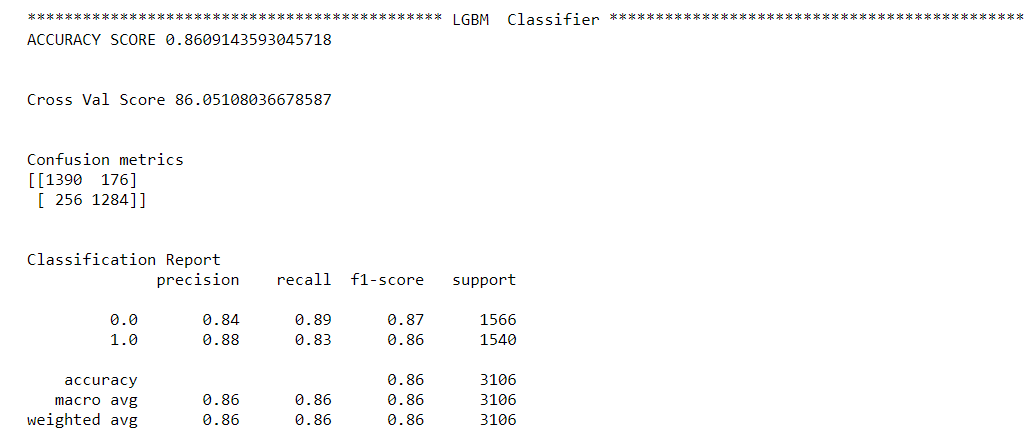
* Logistic Regression
* Random Forest Classifier
* ExtraTrees Classifier
* DecisionTree Classifier
* XGBoost Classifier
* LGBM Classifier

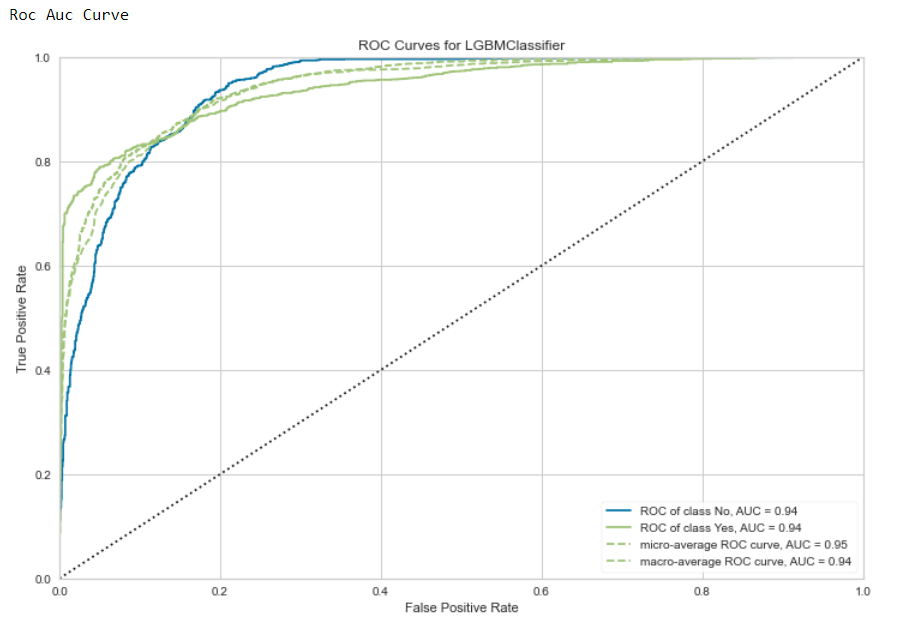


Our best model came out: LGBM Classifier

With Accuracy: 86.09%

Cross Validation Score: 86





**Hyperparameter Tuning:**

The task of selecting a set of ideal hyperparameters for a learning algorithm is known as hyperparameter optimization or tuning. A hyperparameter is a value for a parameter that is used to influence the learning process.

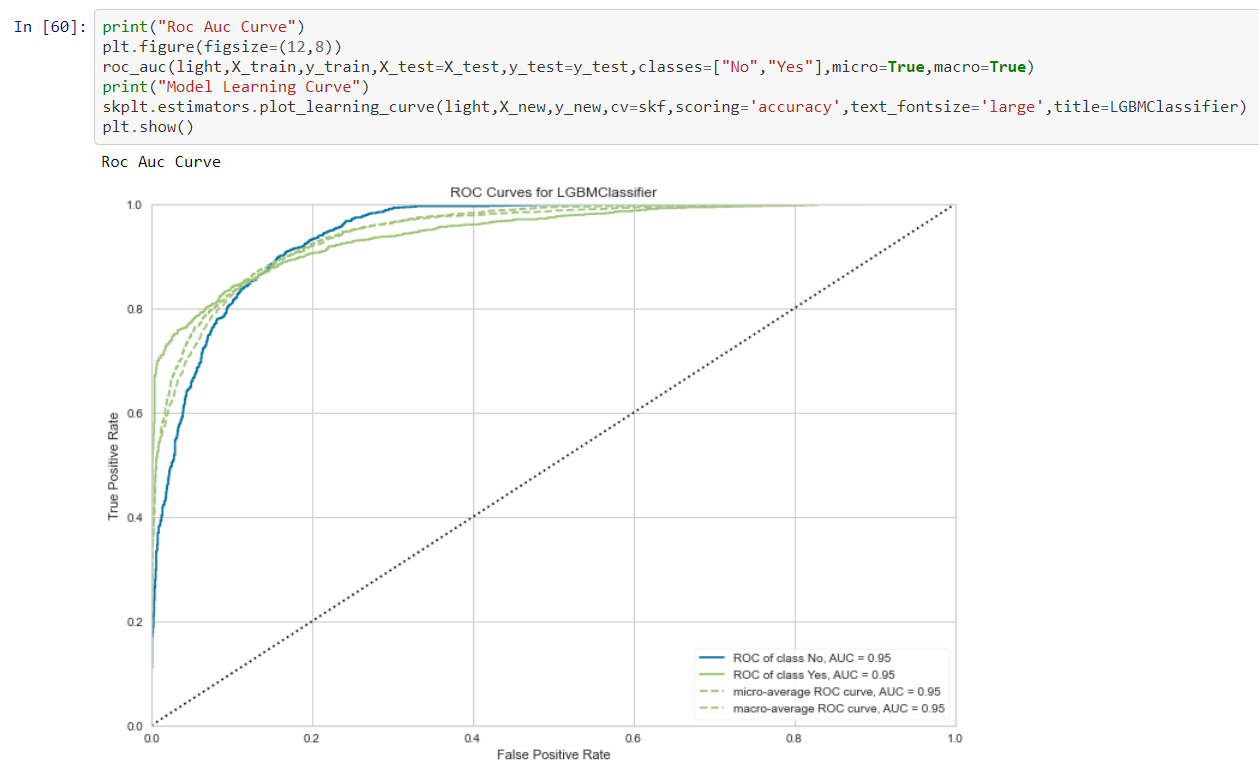
For LGBM GridSearchCV, the ideal parameters that lead to the best model performance is determined.

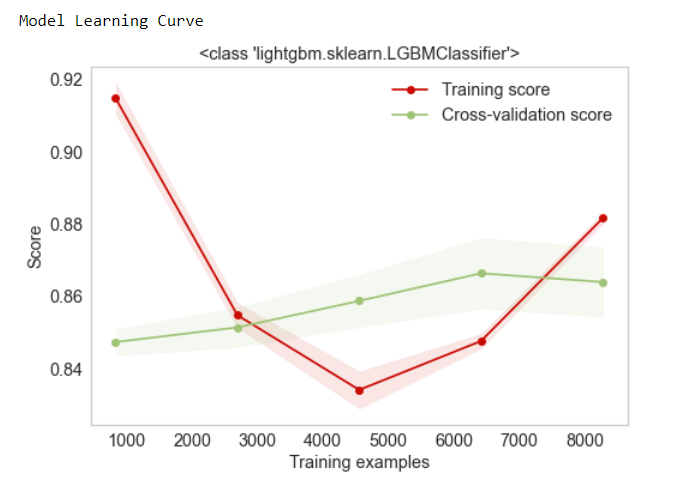


On doing hyperparameter tuning, we improved the model from 86.01 to 86.20 with the best parameters as:

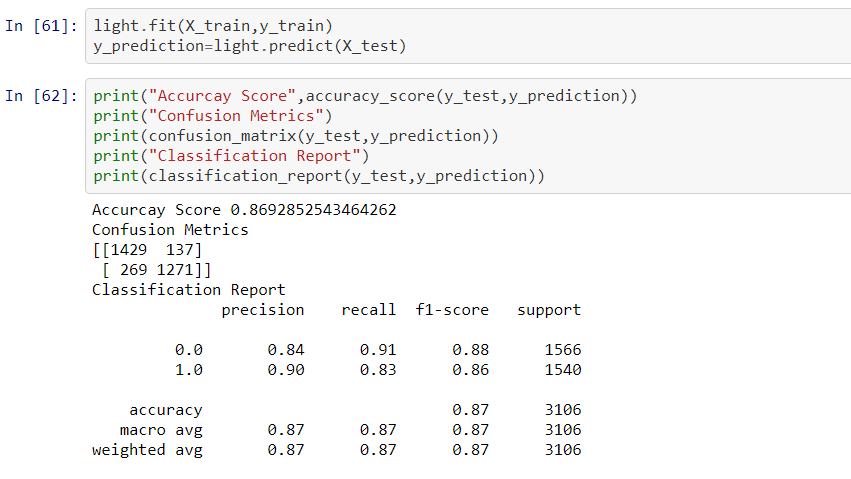


**Final Model ROC AUC & Learning Curve:**





**Final Model Metrics:**



**Conclusion:**

* With the help of these final model metrics, it is clearly seen that the model accuracy is increased from 86.01 to 86.92 using LGBM Classification.
* The customers have partners and dependents will have less chances of churn.
* The customers have online security and tech support also will have less chances to churn.
* The customers using paperless billing and electronic check - they have high chances of churn and having month to month contracts.
* So on that basis we have the data and can forecast the churn probability of customers and provide the services which can retain them and build loyal customers.
* This can reduce the customer churn.