**CUSTOMER CHURN PROJECT**

**Developing Machine Learning Model for predicting ‘Customer Churn'**

INRODUCTION

Customer churn is one of the most important metrics for a growing business to evaluate. While it's not the happiest measure, it's a number that can give your company the hard truth about its customer retention.It's hard to measure success if you don't measure the inevitable failures, too. While you strive for 100% of customers to stick with your company, that's simply unrealistic. That's where customer churn comes in.

# What is ‘CUSTOMER CHURN’?

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.

# The Problem Statement

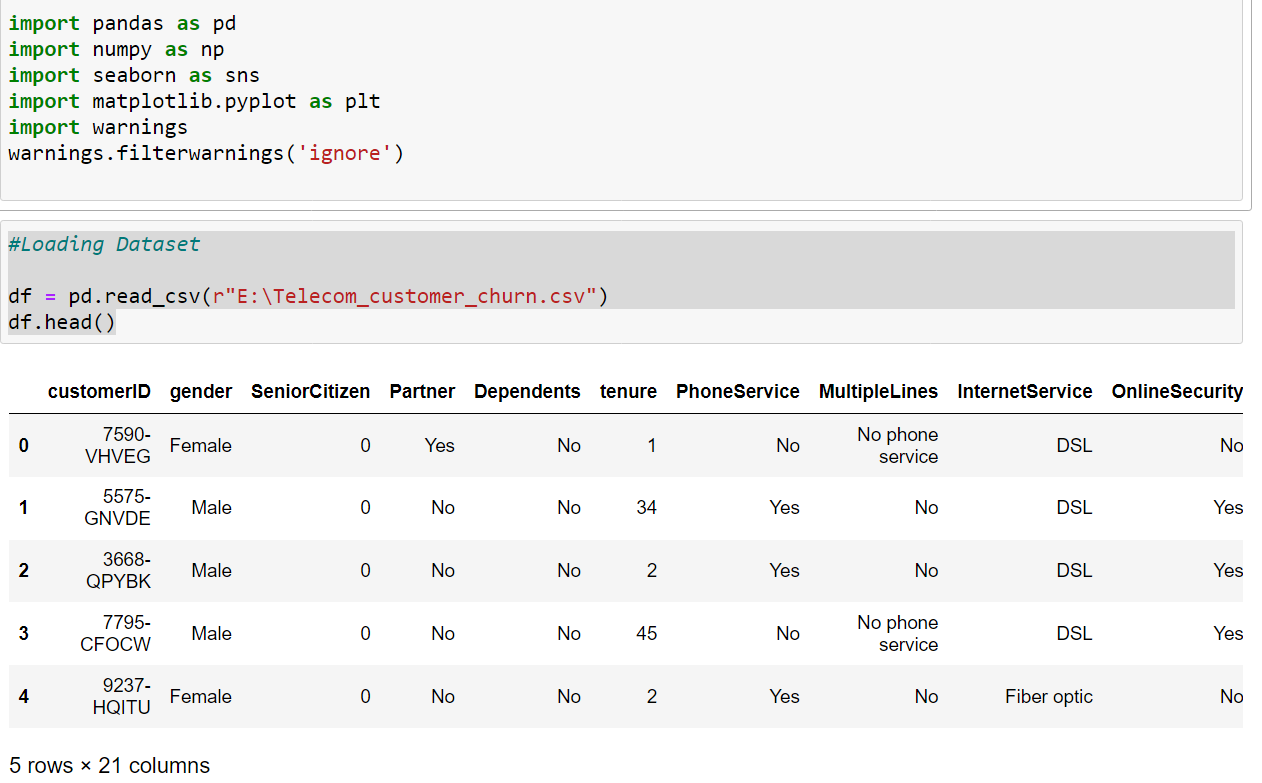
Preventing customer churn is critically important to the telecommunications sector, as the barriers to entry for switching services are so low.

We will examine customer data from IBM Sample Data Sets with the aim of building and comparing several customer churn prediction models.

# About Dataset

**Dataset Link:** https://github.com/dsrscientist/DSData/blob/master/Telecom\_customer\_churn.csv

# Load Dataset



After giving a close look, we found that all there are 21 features in our dataset. Here, it is not necessary that all features will contribute equally to our target variable.

We need to think practically for each and every feature with respect to relationship with our target feature and hence, deal accordingly.

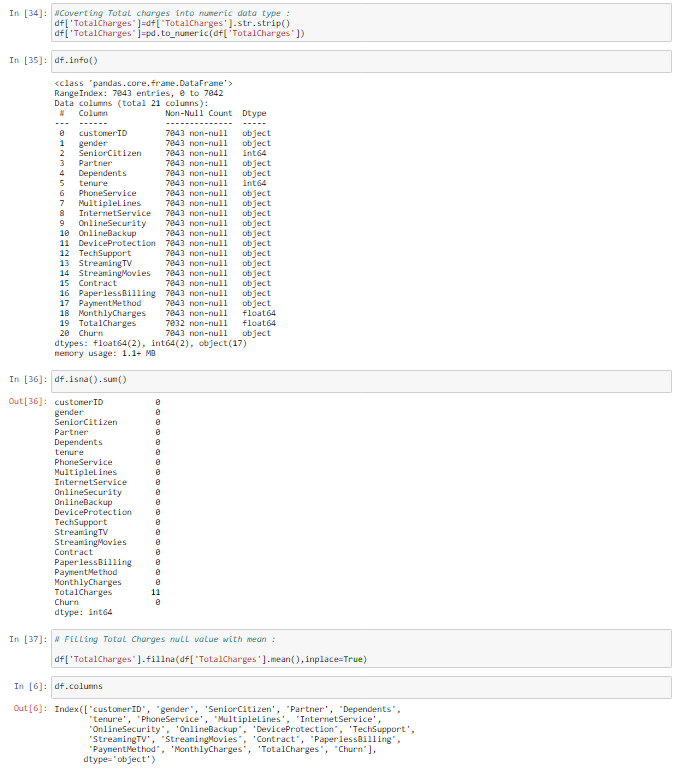
For example – As you can see above, at 0 index there is one feature called ‘customerID’.We can remove that column because it will not help in defining our target.

# Data Analysis

In this part, we analyse the data by checking its data type, data info, missing values or null values , statistics , std. deviation ,value count and uniqueness in every column etc.

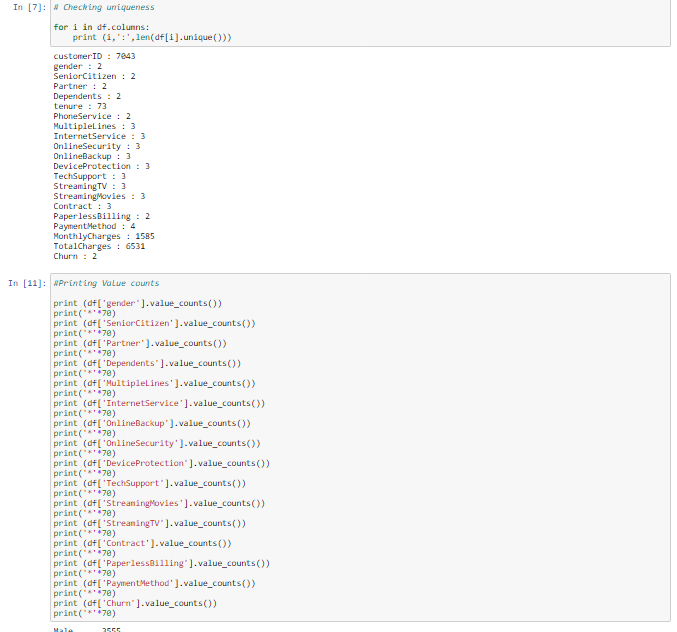


* We can clearly see , the dataset consists of 7043 rows and 21 columns.
* At index 2,5 and 18 the dataset is numeric (float or integer) and rest all columns consisting object data type. Hence, we need to convert them into numeric for our machine to get it understand.
* There are no null values present in our dataset.



As you can clearly analyse, we changed column ‘TotalCharges’ data type into numeric and found there are 11 null values.

Data in column ‘TotalCharges’ is continuous so, filling missing values with mean of whole column.



Male 3555

Female 3488

Name: gender, dtype: int64

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

0 5901

1 1142

Name: SeniorCitizen, dtype: int64

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

No 3641

Yes 3402

Name: Partner, dtype: int64

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

No 4933

Yes 2110

Name: Dependents, dtype: int64

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

No 3390

Yes 2971

No phone service 682

Name: MultipleLines, dtype: int64

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Fiber optic 3096

DSL 2421

No 1526

Name: InternetService, dtype: int64

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

No 3088

Yes 2429

No internet service 1526

Name: OnlineBackup, dtype: int64

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

No 3498

Yes 2019

No internet service 1526

Name: OnlineSecurity, dtype: int64

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

No 3095

Yes 2422

No internet service 1526

Name: DeviceProtection, dtype: int64

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

No 3473

Yes 2044

No internet service 1526

Name: TechSupport, dtype: int64

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

No 2785

Yes 2732

No internet service 1526

Name: StreamingMovies, dtype: int64

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

No 2810

Yes 2707

No internet service 1526

Name: StreamingTV, dtype: int64

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Month-to-month 3875

Two year 1695

One year 1473

Name: Contract, dtype: int64

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Yes 4171

No 2872

Name: PaperlessBilling, dtype: int64

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Electronic check 2365

Mailed check 1612

Bank transfer (automatic) 1544

Credit card (automatic) 1522

Name: PaymentMethod, dtype: int64

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

No 5174

Yes 1869

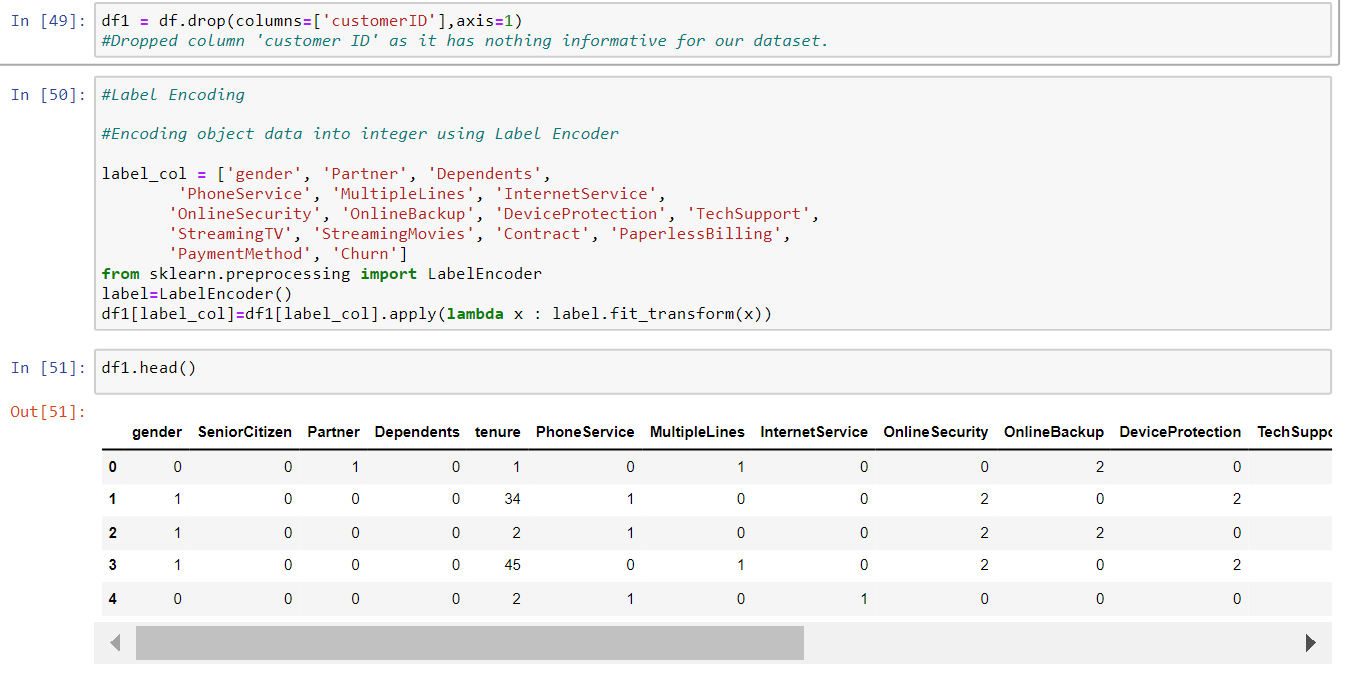
Name: Churn, dtype: int64

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

* We analyse uniqueness and value count in every column and get information inside from our dataset.

Encoding

The performance of a machine learning model not only depends on the model and the hyperparameters but also on how we process and feed different types of variables to the model. Since most machine learning models only accept numerical variables, preprocessing the categorical variables becomes a necessary step. We need to convert these categorical variables to numbers such that the model is able to understand and extract valuable information.



* Here we use label encoder to encode our categorical columns.

# Exploratory Data Analysis (EDA) and Visualisation Concluding Remarks

Exploratory Data Analysis (EDA) is an approach to analysing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods. This can divided into three parts namely Univariate, Bivariate and Multivariate Analysis.

**Importing visualisation libraries**

**import** **matplotlib.pyplot** **as** **plt**

**import** **seaborn** **as** **sns**

## Univariate Analysis

Univariate Analysis means analysis of one variable or one feature at a time and it basically tells us how data in each feature is distributed and also tells us about central tendencies like mean, median, and mode as well as presence of outliers in the dataset.

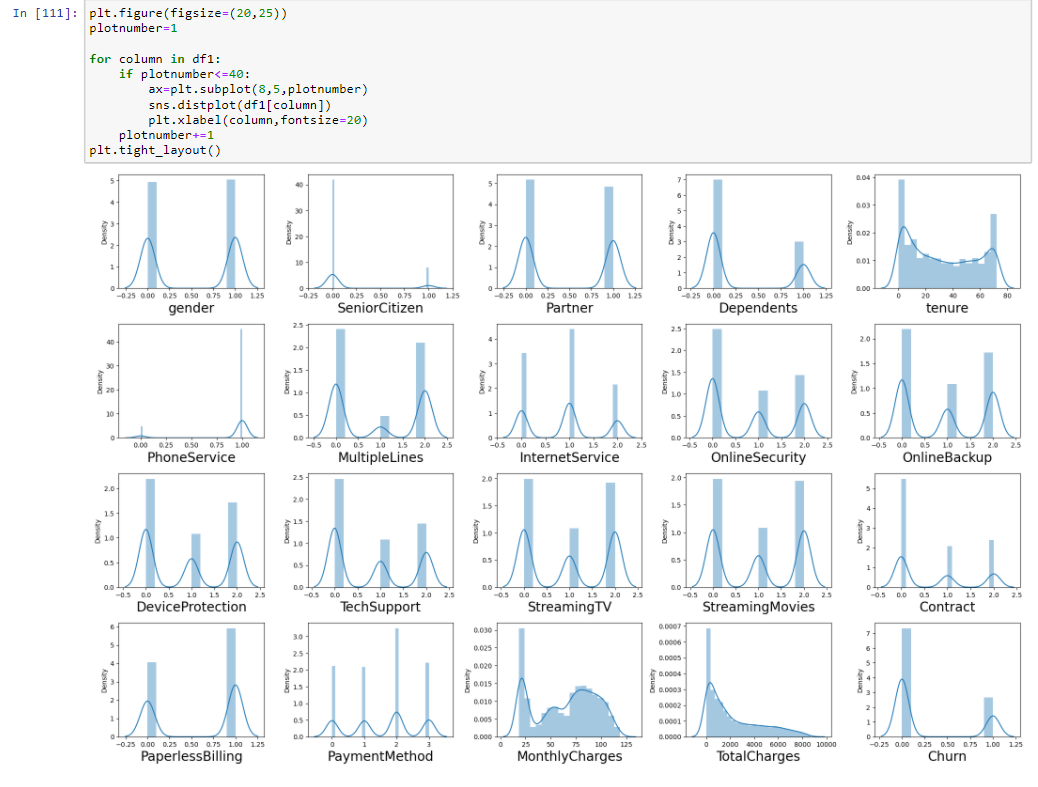
**Data Distribution using distplot:** Distribution of data basically tells us about the mean, median, mode, maximum, minimum, standard deviation and skewness of data. We can get and visualize them using distribution plot of seaborn library as shown below:

**import** **warnings**

warnings.simplefilter('ignore')

*#Checking data distribution using distplot.*

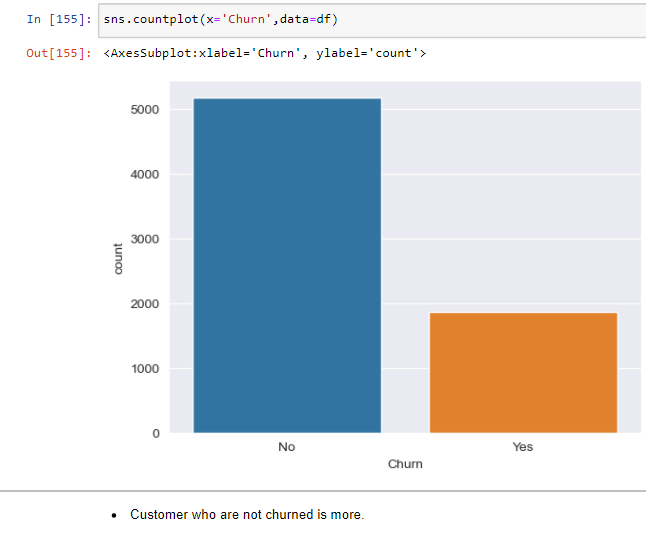
We can see that distribution plot shows, the data is not distributed normally in all of the features and this is understandable because all of the features are of categorical type. Now moving further with Bivariate analysis to check the relationship between features.



* Looks like there is some skewness present in Total Charges, Monthly Charges and Tenure so we will treat it using log transformation.

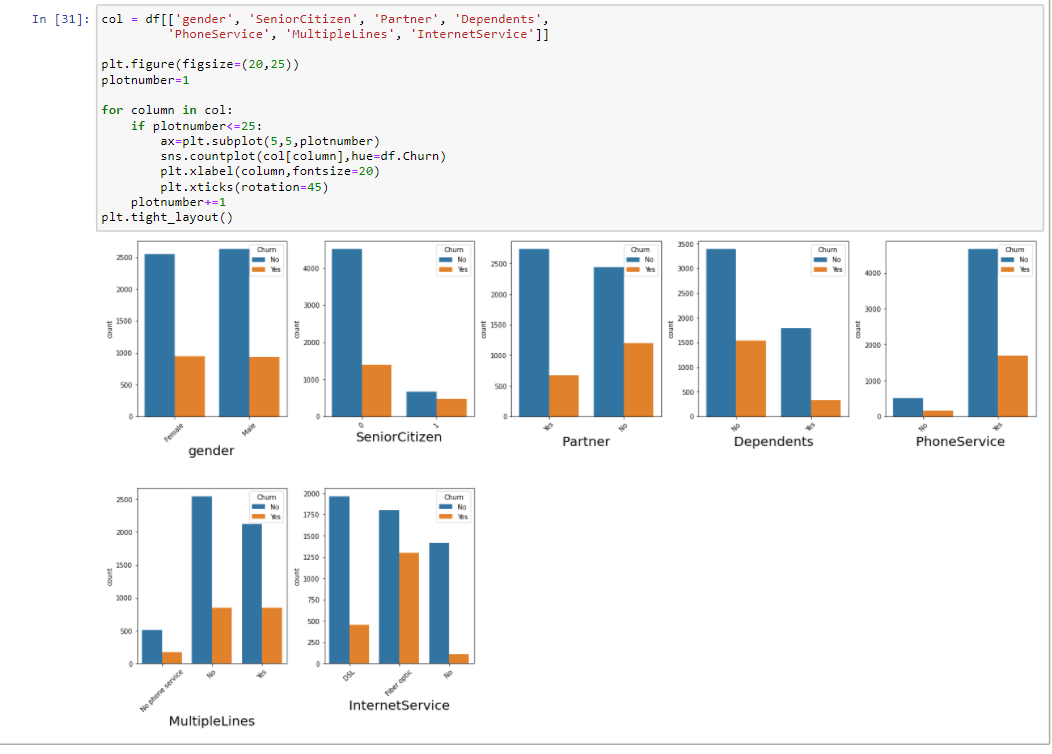


* Seems to be there are no outliers present in our data so we do not need to treat them.



## Bivariate Analysis

Bivariate analysis is one of the simplest forms of quantitative analysis. It involves the analysis of two variables (namely X and Y), for the purpose of determining the empirical relationship between them. Therefore, checking the relationship of some feature such as ‘gender’ , ‘SeniorCitizen’ etc. with target variable ‘CHURN’ to get more information on it.



* Both Male and Female are almost equally churned.
* Churn rate of Senior Citizens are high.
* Customer without partner has more Churn rate.
* Churn Rate is high for no dependents.
* Churn Rate is high for Phone Service
* For those who have fiber optic Internet , churn rate is very high.
* Customers are churned equally for having or not having multiple lines but, less chured if they do not have phone service

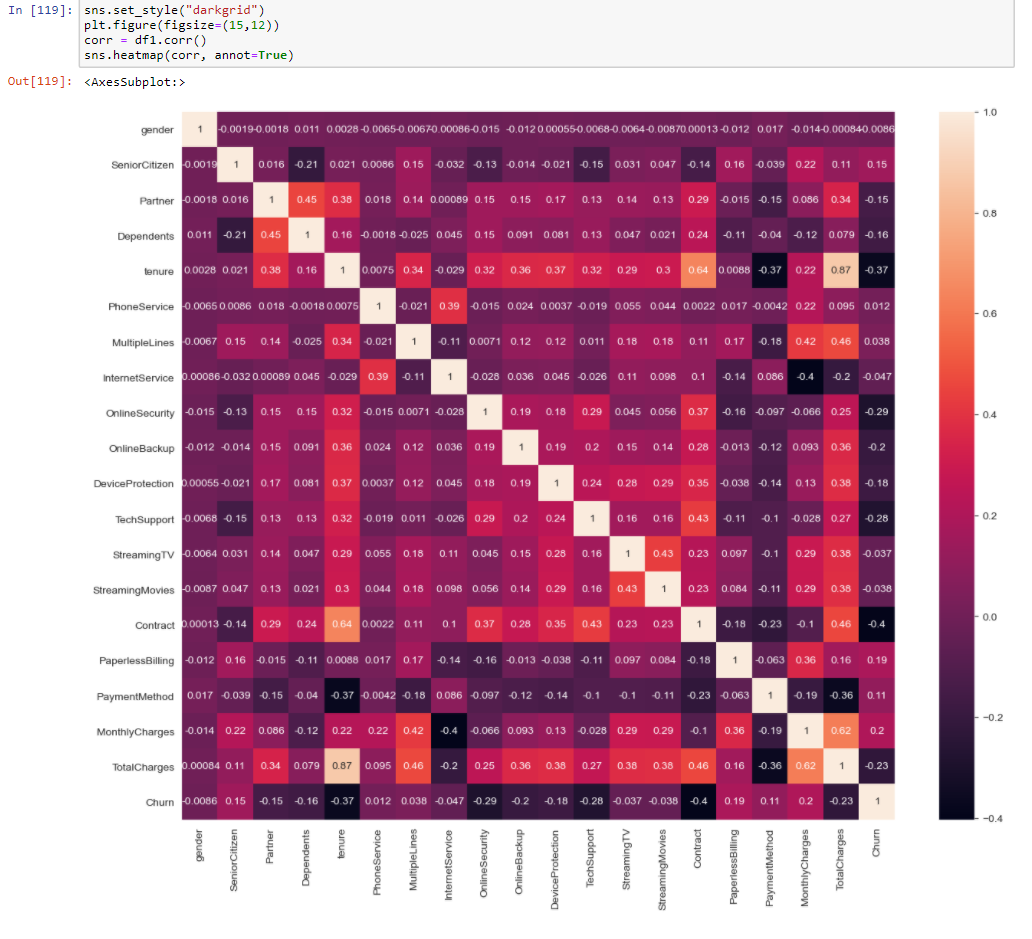
## Multi Variate Analysis

Multivariate analysis refers to any statistical technique used to analyse more complex sets of data.



* Above, we are comparing various features with gender and our target column ‘Churn’.

### Correlation of Features



From the above correlation depiction by heatmap, we can clearly infer that:

* The more brighter the colour , more will be the correlation between features.
* Correlation value near or equal to 1 means features are highly correlated among each other. Then, this is the condition of multicollinearity.
* As a Data scientist, we should avoid multicollinearity by dropping suitable column.

# Prepare Dataset for Model Training

Preparing dataset Model training is the one of core part of machine learning model building and includes different types data modification and transformation to achieve the better model performance.

**Importing Libraries:**

**from** **sklearn.preprocessing** **import** OrdinalEncoder, StandardScaler, power\_transform

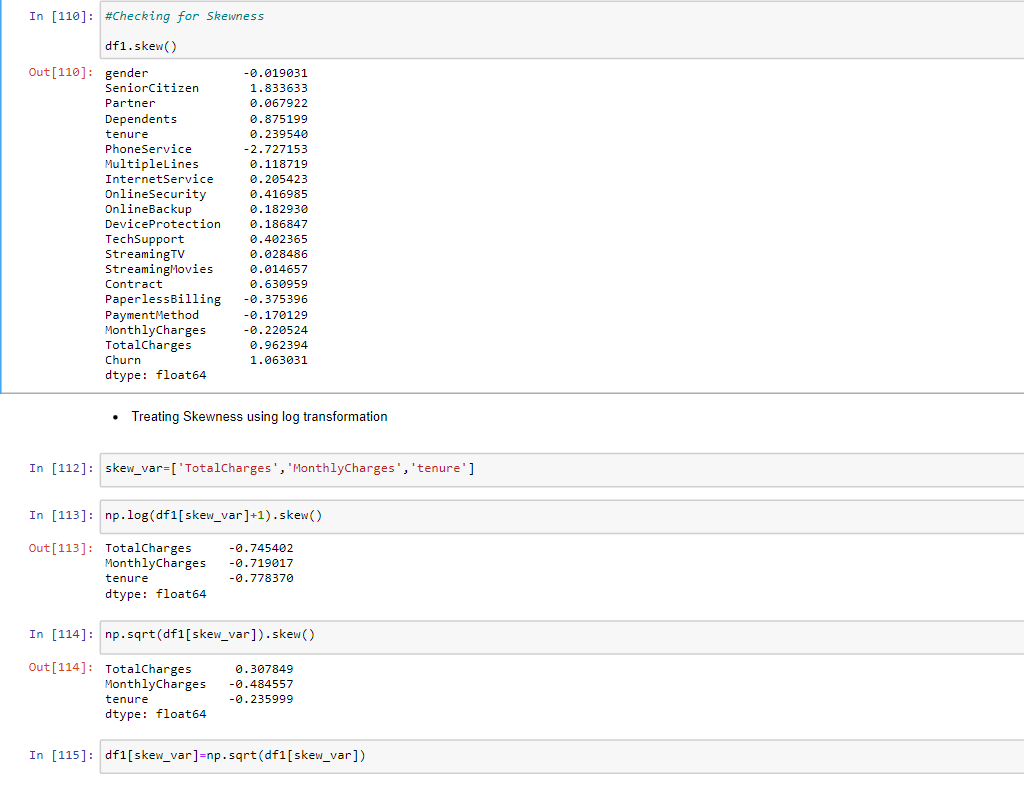
**from** **scipy.stats** **import** zscore

## Outlier(s): Detection & Removal

Outliers are extreme values that fall a long way outside of the other observations. It can be detected and removed using either Z-Score or Interquartile Range (IQR) methods. Here we are going to use z-score for this purpose.

From above box plot, we can see that there are no outliers present in our dataset. Hence, we will not remove any.

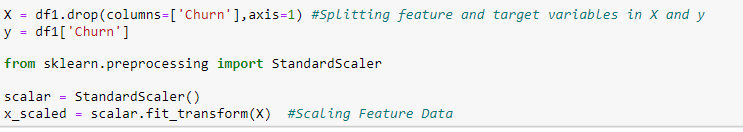
## Skewness: Detection & Treatment



* Above, we removed skewness from data using log transformation.

## Separate Input and Output/Target Variable and Scale Feature Data for Model Training

Now, we can separate the features into input as X and output/target as Y to continue further with data preparation.

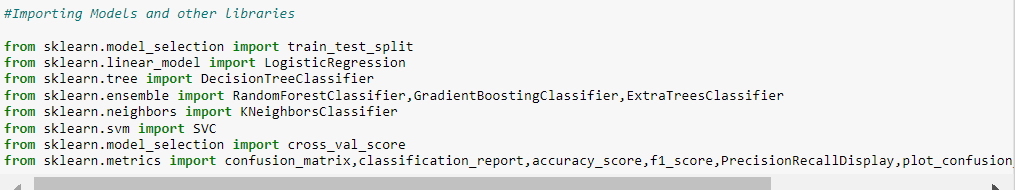


* Scaling data for model training is refers to normalize the range of independent variables or features of data. Here we are using StandardScaler for this purpose:

# Model Training: Finding the best model

The models that I have decided to train for this dataset are LogisticRegression, RandomForestClassifier, DecisionTree, KNeighboursClassifier, SVC (Support Vector Classifier) models. The goal here is to find the best hyper-tuned models for further processing.

**Importing Libraries:**



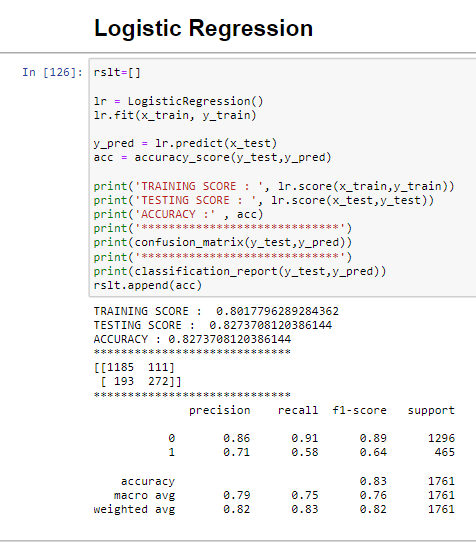
## Applying loop for getting best random\_state

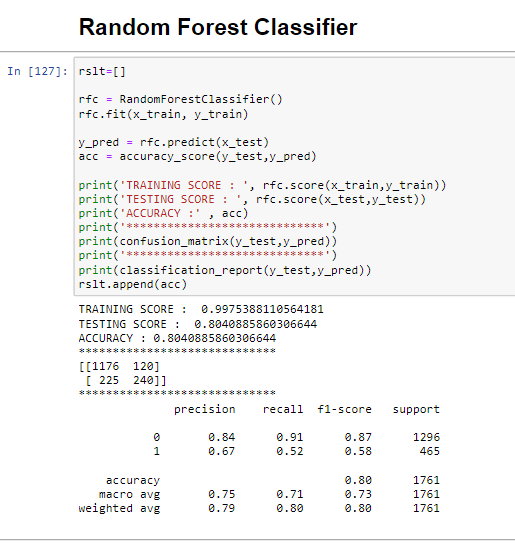
## 

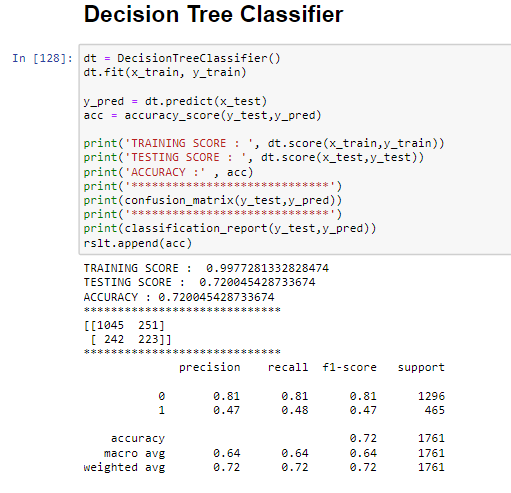
## Prepare Model List and Test to get Best Model

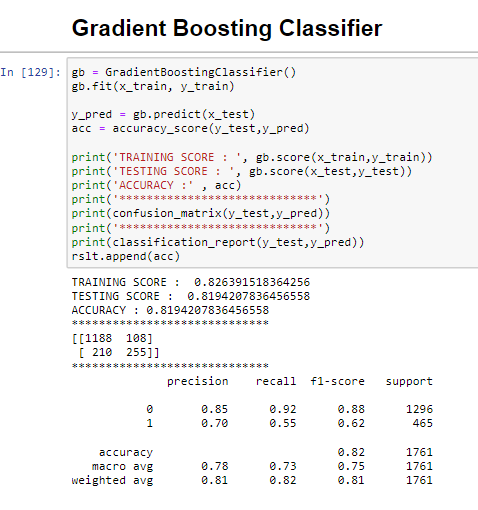
This part includes the preparation of model list with parameters and then train and test with best parameter to get the best hyper-tuned model performances.

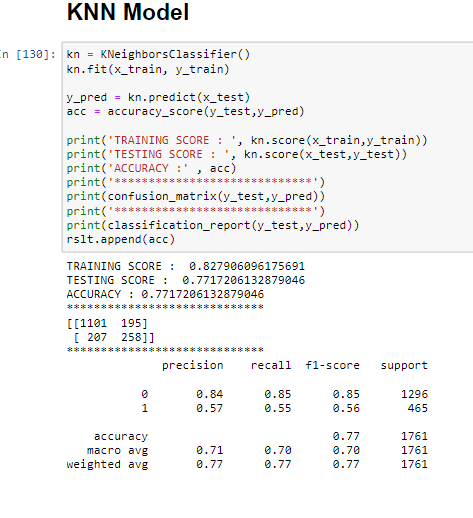
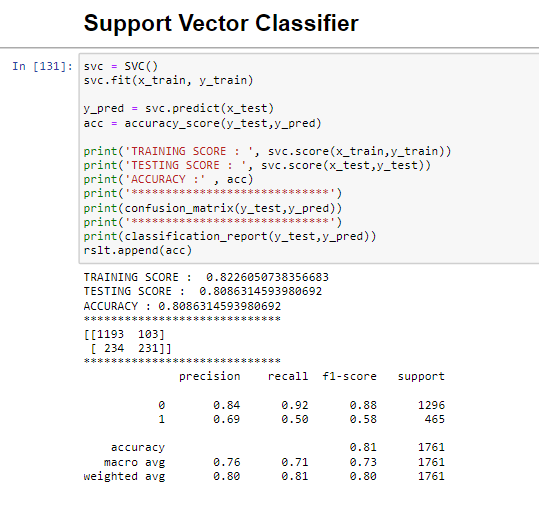
### 

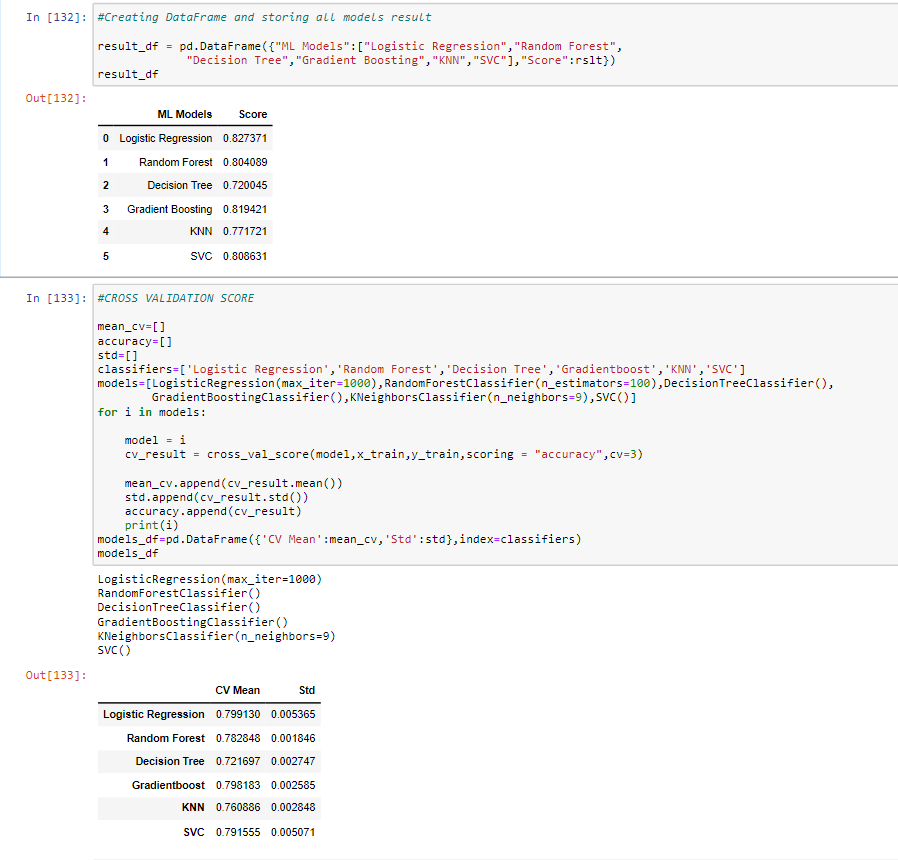








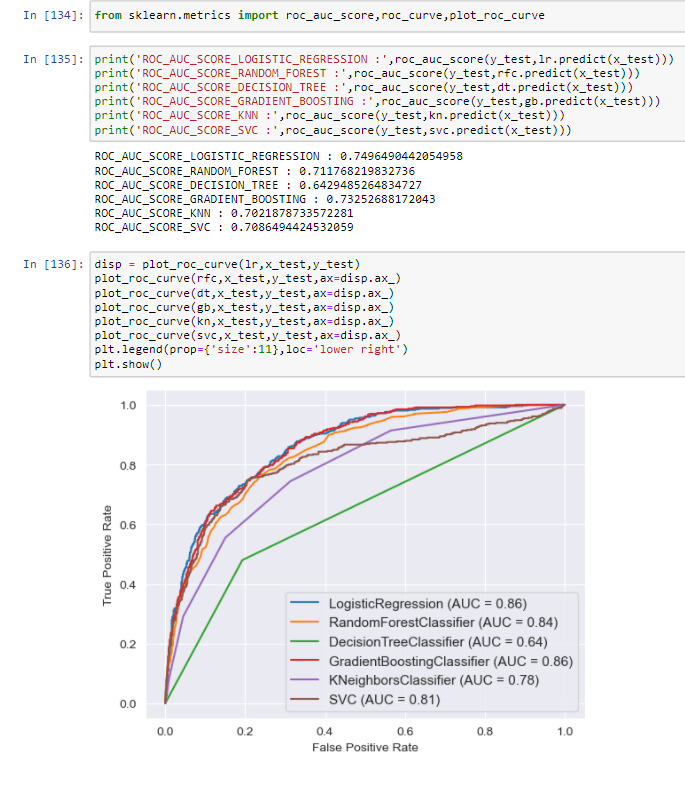
 



From the above model performances we can easily conclude that the model GradientBoostingClassifer performs well with Accuracy Score of 81% as compare to other models. Also, its cross validation score is also near to its accuracy score so Therefore, moving further with **GradientBoostingClassifier**.

# PLOT ROC-AUC CURVE AND DETERMINING SCORES

In Machine Learning, performance measurement is an essential task. So when it comes to a classification problem, we can count on an AUC - ROC Curve. When we need to check or visualize the performance of the multi-class classification problem, we use the AUC (**Area Under The Curve**) ROC (**Receiver Operating Characteristics**) curve. It is one of the most important evaluation checking any classification model’s performance. It is also written as AUROC (**Area Under the** **Receiver Operating Characteristics**).



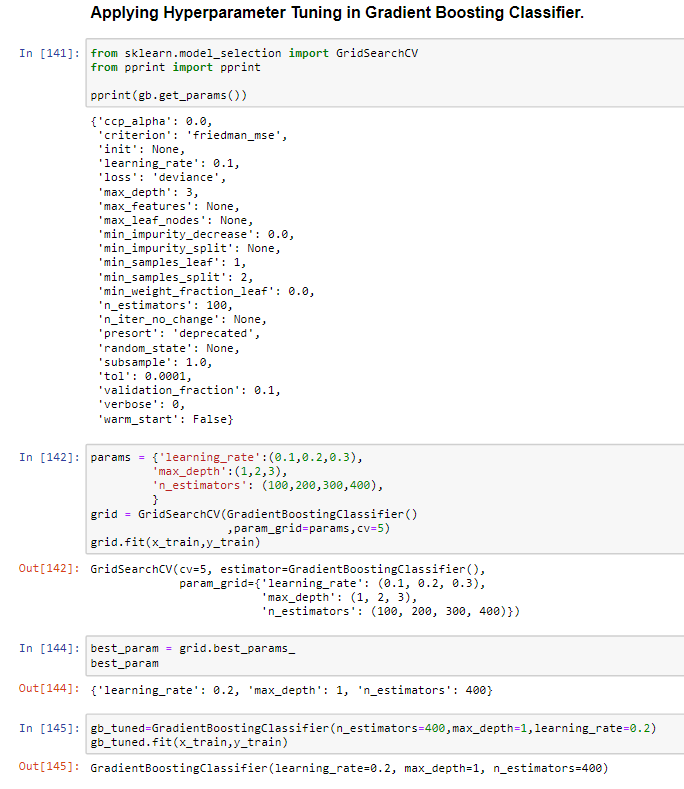
##### *As per ROC AUC curve score Logistic Regression and Gradient Boosting Classifier are best fit models.*[*¶*](http://localhost:8888/notebooks/Customer%20Churn%20Analysis.ipynb#As-per-ROC-AUC-curve-score-Logistic-Regression-and-Gradient-Boosting-Classifier-are-best-fit-models.)

##### *So, we will do Hyperparameter tuning in any one of them.*

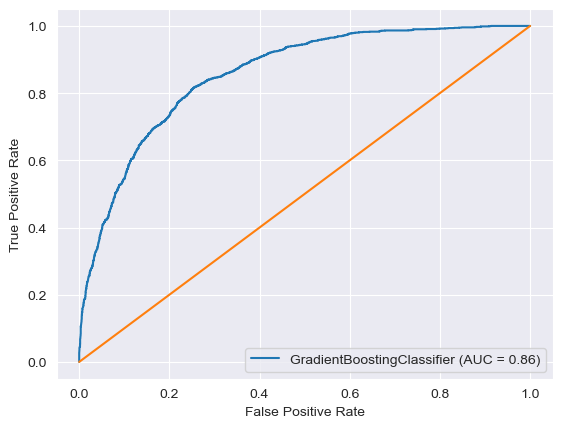
# HYPERPARAMETER TUNING USING GRID SEARCH CV

In [machine learning](https://en.wikipedia.org/wiki/Machine_learning), **hyperparameter optimization** or tuning is the problem of choosing a set of optimal [hyperparameters](https://en.wikipedia.org/wiki/Hyperparameter_(machine_learning)) for a learning algorithm. A hyperparameter is a [parameter](https://en.wikipedia.org/wiki/Parameter) whose value is used to control the learning process. By contrast, the values of other parameters (typically node weights) are learned.

The same kind of machine learning model can require different constraints, weights or learning rates to generalize different data patterns. These measures are called hyperparameters, and have to be tuned so that the model can optimally solve the machine learning problem. Hyperparameter optimization finds a tuple of hyperparameters that yields an optimal model which minimizes a predefined [loss function](https://en.wikipedia.org/wiki/Loss_function) on given independent data.The objective function takes a tuple of hyperparameters and returns the associated loss. [Cross-validation](https://en.wikipedia.org/wiki/Cross-validation_(statistics)) is often used to estimate this generalization performance.[[](https://en.wikipedia.org/wiki/Hyperparameter_optimization#cite_note-bergstra-2)

Hyperparameters are **crucial as they control the overall behaviour of a machine learning model**. The ultimate goal is to find an optimal combination of hyperparameters that minimizes a predefined loss function to give better results.

# 



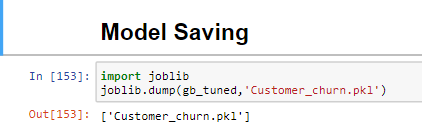
# EFFICIENCY ACHIEVED

* After hyperparameter tuning we are able to increase roc\_auc\_score.
* Model accuracy is increased.
* True Positive is increased in confusion matrix

# Model Selection: The Final Model

In this step, we will save or serialize the final model which gives the highest performance into an object or pickle file.

**Importing Libraries and Saving Model:**



## Conclusion

The final model performance is good with **Accuracy 82.39%**

# Concluding Remarks

To expand on this project to get a better final model, I would like to see if there is similar data available for further analysis. This way I can test my model further and perhaps find some more insight into what will work best in a real-world situation. My model performed perfectly on my dataset, so it’s worth questioning if the dataset was perhaps compromised and therefore, testing my model on another dataset could provide further validation.