Customer Churn Prediction

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

**2.Data Analysis:**

To work on any machine learning project,we need to import basic libraries such as pandas, numpy, some visualization libraries such as seaborn and matplotlib,while working on data,it give some warnings, such warnings can be ignored,

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import warnings

warnings.filterwarnings('ignore')

Pandas and numpy(numerical python) are libraries used for data analysis, and doing machine learning task, by using this libraries we can do data cleaning,data merging,operations on rows and columns,it means operations of dataframe, in dataframe operation we need some common operation such deletion or insertion of any row or columns, coversion of one data type to another data types.Numpy is used for mathematical operations on array which consumes less memory. Matplotlib is 2D data visualization library.its coding is similar to Matlab.Data can be easily analyzed by Matplotlib using python libraries such as pandas and numpy.Seaborn has less syntax and default themes,seaborn graphics is more beautiful than Matplotlib and avoid overlapping of plots with the help of themes.

pd.set\_option('display.max\_columns',None)

This command displays all the columns in dataset, to view all rows, we will replace word “columns” by rows. After displaying all columns we can actually judge the data type and contents of column

customer=pd.read\_csv("E:\\Data Science\\Data Trained Evaluation Project\\Telecom\_customer\_churn.csv")

customer.head(10)

Here we import the dataset which in csv format, the content in the bracket shows the path of dataset.we can read other data format also like exel and JSON.

customer.shape

for the customer dataset, we get 7043 rows and 21 columns.

**3. EDA:**

customer=customer.drop(columns=['customerID'])

Total customer in the data is 7043,and everyone has own unique ID, we can use unique ID as index also,so it will not take part in data analysis, here we drop this feature,as it not so important, now we will use default index.

Data columns (total 20 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 gender 7043 non-null object

1 SeniorCitizen 7043 non-null int64

2 Partner 7043 non-null object

3 Dependents 7043 non-null object

4 tenure 7043 non-null int64

5 PhoneService 7043 non-null object

6 MultipleLines 7043 non-null object

7 InternetService 7043 non-null object

8 OnlineSecurity 7043 non-null object

9 OnlineBackup 7043 non-null object

10 DeviceProtection 7043 non-null object

11 TechSupport 7043 non-null object

12 StreamingTV 7043 non-null object

13 StreamingMovies 7043 non-null object

14 Contract 7043 non-null object

15 PaperlessBilling 7043 non-null object

16 PaymentMethod 7043 non-null object

17 MonthlyCharges 7043 non-null float64

18 TotalCharges 7043 non-null object

19 Churn 7043 non-null object

dtypes: float64(1), int64(2), object(17)

memory usage: 1.1+ MB

The info(). Method is used to find datatype of each column as well as we get non null count for each column. Out of 20 columns only one column has data type float64, two columns has data type integer, remaining 16 columns has object data types.

gender 0

SeniorCitizen 0

Partner 0

Dependents 0

tenure 0

PhoneService 0

MultipleLines 0

InternetService 0

OnlineSecurity 0

OnlineBackup 0

DeviceProtection 0

TechSupport 0

StreamingTV 0

StreamingMovies 0

Contract 0

PaperlessBilling 0

PaymentMethod 0

MonthlyCharges 0

TotalCharges 0

Churn 0

dtype: int64

By using customer.isnull().sum() command, we get total null values for each column. .in this dataset we don’t have any null values , so no need of imputation techniques for data.

customer.describe()

Describe method give statistical description of numerical columns only, there are 3 numerical columns in our dataset ,first column 'SeniorCitizen', has min value 0 and max value 1,so it is categorical feature. "tenure" and 'MonthlyCharges'is continous data feature with its mean and standard deviation are given, along with this first,second and third quartile range is also given.

**3.1 Univariate Analysis**

In univariate analysis we deal with categorical values and continous values. we build count plot for each column, also get value counts and non unique values for each column.

1. customer['gender'].value\_counts()

Male 3555

Female 3488

Name: gender, dtype: int64

Out of 7043 customers ,3555 are male customer and 3488 are female.There are only two class, here we implement the count plot for feature “gender”



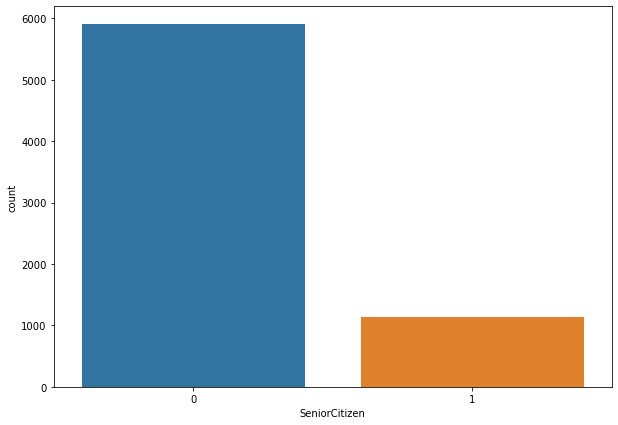
1. customer['SeniorCitizen'].value\_counts()

0 5901

1 1142

Name: SeniorCitizen, dtype: int64

The feature “SeniorCitizen” has two class 0 and 1, 0 class has count 5901 and class 1 has count 1142.lets see its count plot.



We can implement count plot for below features as well, just by changing the name of feature in countplot command

1. customer['Partner'].value\_counts()

No 3641

Yes 3402

Name: Partner, dtype: int64

Customer 3641 has no partner,3402 has partner

1. customer['Dependents'].value\_counts()

No 4933

Yes 2110

Name: Dependents, dtype: int64

Total customer 4933 has no dependant, and 2110 has dependant

1. customer['PhoneService'].value\_counts()

Yes 6361

No 682

Name: PhoneService, dtype: int64

The number of customer has phoneService 6361 and 682 has no phoneservice

1. customer['MultipleLines'].value\_counts()

No 3390

Yes 2971

No phone service 682

Total 3390 customer has no multiplelines and 2971 customer has taken multiple lines

1. customer['InternetService'].value\_counts()

Fiber optic 3096

DSL 2421

No 1526

Name: InternetService, dtype: int64

This is type of connectivity for internet, total 3096 customer selected fibre optic network,2421 customer selected Digital Subscriber line(DSL) and 1526 customer has not taken internet service

1. customer['OnlineSecurity'].value\_counts()

No 3498

Yes 2019

No internet service 1526

The service of “onlineSecurity” is selected by 2019 customer, and 3498 has not taken this service

1. customer['OnlineBackup'].value\_counts()

No 3088

Yes 2429

No internet service 1526

The service of “OnlineBackup” is selected by 2429 customer, and 3088 has not taken this service

1. customer['DeviceProtection'].value\_counts()

No 3095

Yes 2422

No internet service 1526

The service of “DeviceProtection” is selected by 2422 customer, and 3095 has not taken this service, and 1526 are not internet user

11)customer['TechSupport'].value\_counts()

No 3473

Yes 2044

No internet service 1526

The service of “TechSupport” is selected by 2044 customer, and 3473 has not taken this service, and 1526 are not internet user

12 ) customer['StreamingTV'].value\_counts()

No 2810

Yes 2707

No internet service 1526

The service of “StreamingTV” is selected by 2707 customer, and 2810 has not taken this service, and 1526 are not internet user

13) customer['StreamingMovies'].value\_counts()

No 2785

Yes 2732

No internet service 1526

The service of “StreamingMovies” is selected by 2732 customer, and 2785 has not taken this service, and 1526 are not internet user

14) customer['Contract'].value\_counts()

Month-to-month 3875

Two year 1695

One year 1473

The service of “Contract” is month to month is for 3875,then 1695 customer has made contract of two year, and 1473 customer made contract of one year

15) customer['PaperlessBilling'].value\_counts()

Yes 4171

No 2872

“PaperlessBiling” is a payment option , total 4171 customer select this, and 2872 has no selected “PaperlessBiling”

16)customer['PaymentMethod'].value\_counts()

Electronic check 2365

Mailed check 1612

Bank transfer (automatic) 1544

Credit card (automatic) 1522

“PaymentMethod' is categorized into 4 mode, total 2365 people make payment by Electronic Check, 1612 customer make payment by Mailed Check, 1544 makes payment by Bank transfer and 1522 customer make payment by credit card.

17) customer['Churn'].value\_counts()

No 5174

Yes 1869

Total 5174 customer has not stopped business with company and 1869 has stopped business with company

**3.2 Change of Data type**

The feature “TotalCharges” has datatype object , but contents are float number, so we used value\_counts() method, which shows that 11 space are present in the feature, we found this using following command.we get idex location of empty space.

customer.loc[customer['TotalCharges']==' ']

To remove this space, we use replace method which space are replaced by NaN ,so we get 11 NaN, now the datatype of this column is float, so to replace NaN values, we use mean() method, in which NaN location are replaced by mean.

customer['TotalCharges']=customer['TotalCharges'].replace(np.nan,customer['TotalCharges'].mean())

The Label “Churn” is categorical data type, so our problem is Classification Type.

customer\_num=customer.drop(columns=['gender','SeniorCitizen','Partner','Dependents','PhoneService','MultipleLines','InternetService','OnlineSecurity','OnlineBackup','DeviceProtection','TechSupport','StreamingTV','StreamingMovies','Contract','PaperlessBilling','PaymentMethod','Churn'],axis=1)

customer\_num.head()

In above command, we make separate dataframe for those features which has continous data type,Customer\_num is continous data type dataframe, so we drop categorical feature.

customer\_num.shape

(7043, 3)

Here shape method shows 3 features are continous data type, the features are “**tenure”,” MonthlyCharges” and “TotalCharges” .**

customer\_cat=customer.drop(columns=['tenure','MonthlyCharges','TotalCharges'],axis=1)

customer\_cat.head()

Now , we make separate dataframe only for categorical features, so we drop continous features from original dataframe “customer”

customer\_cat.shape

(7043, 17)

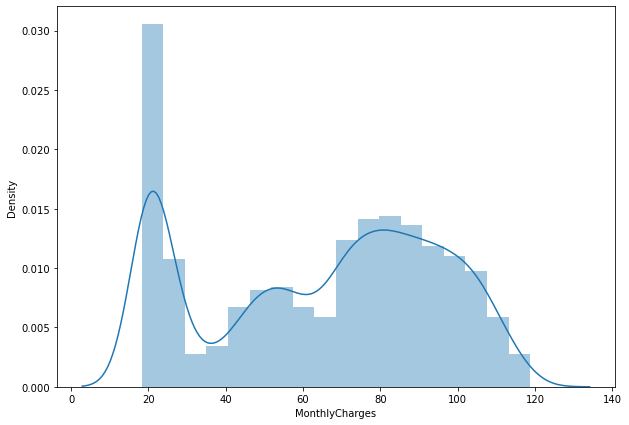
The categorical dataframe contains 17 categorical features.

**3.3 Distplot for Continous features:**

For continous feature, it must posses normal distribution, lets check it,

plt.figure(figsize=(10,7))

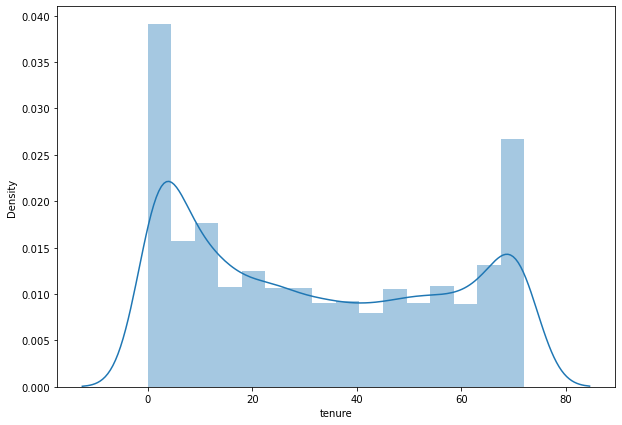
sns.distplot(customer\_num['MonthlyCharges'])



The distplot for feature 'MonthlyCharges' is shown above, no left or right skewness is present in this plot, the top portion is non flat,lets check for next feature.

plt.figure(figsize=(10,7))

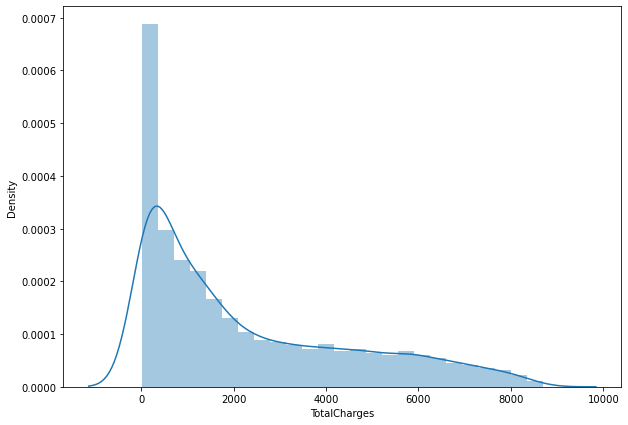
sns.distplot(customer\_num['tenure'])



The distplot for feature “tenure”has no skewness present in the distplot, so we check distplot for next feature,

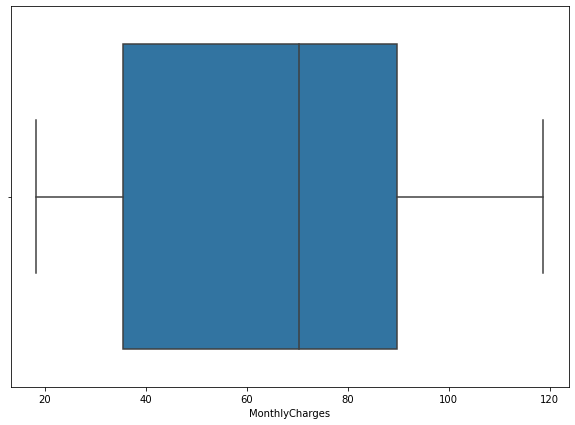
plt.figure(figsize=(10,7))

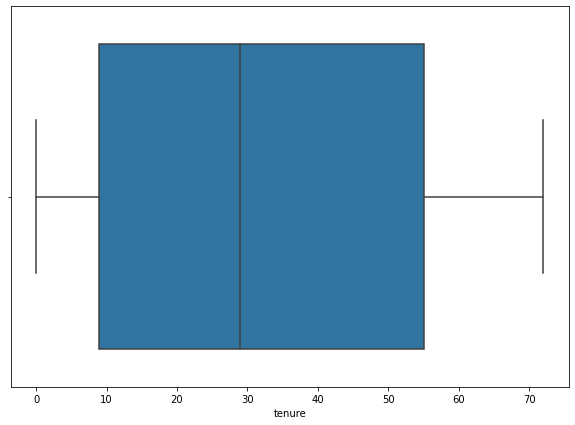
sns.distplot(customer\_num['TotalCharges'])

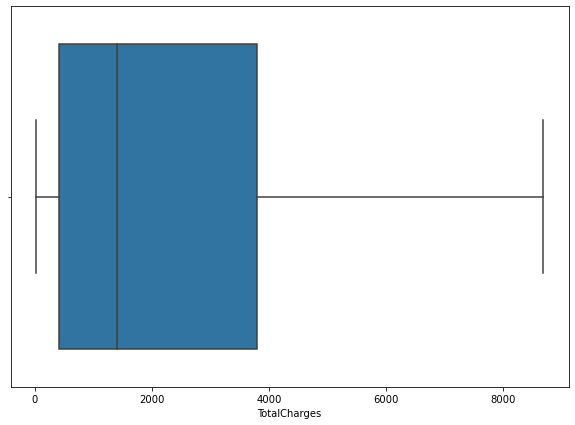


The distplot for feature “TotalCharges” shows long tail toward right direction , it means right skewness is present in this feature. We need to apply skewness removal techniques, like power transformation.

**3.4 Outlier Detection**

****

****



The boxplot is univariate analysis used for outlier detection, it consist of first , second and third quartile, the describe().method is very usefull for finding the values of first, second and third quartile,t he outliers can be present below first quartile or above third quartile,In above three boxplot , no one shows any outlier,this is very good for data analysis as well as model building. Due to absence of any outlier, model shows high confidence. There is no need of z score method or IQR method for removing outliers. We can directly move to skewness removal techniques.

# 3.5 Skewness Removal

customer\_num.skew()

tenure 0.239540

MonthlyCharges -0.220524

TotalCharges 0.962394

Before applying skewness removal, we checked the skewness score of all three features, the skewness range is -0.5 to +0.5 , the feature “tenure” and “MonthlyCharges” shows skewness in between desired range.the feature “TotalCharges” shows skewness more 0.5 ,so it need further processing.

from sklearn.preprocessing import PowerTransformer

scaler=PowerTransformer(method='yeo-johnson')

customer\_num['tenure']=scaler.fit\_transform(customer\_num['tenure'].values.reshape(-1,1))

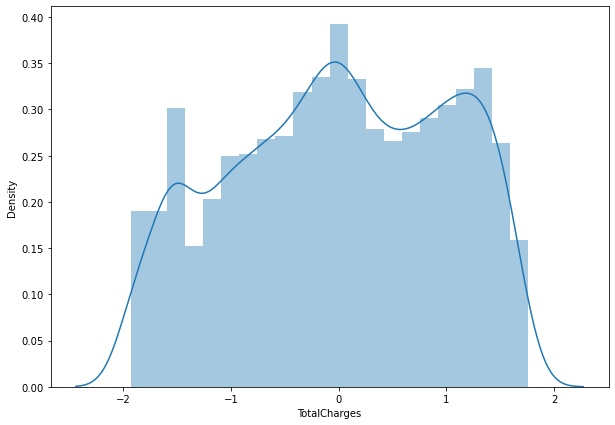
customer\_num['MonthlyCharges']=scaler.fit\_transform(customer\_num['MonthlyCharges'].values.reshape(-1,1))

customer\_num['TotalCharges']=scaler.fit\_transform(customer\_num['TotalCharges'].values.reshape(-1,1))

here powerTransformer is skewness removal technique, we can use sqrt,cbrt and log transformation also, here after applying this transformation skewness is reduced ,now we checked skewness of “totalCharges”

plt.figure(figsize=(10,7))

sns.distplot(customer\_num['TotalCharges'])



We get normal distribution for “TotalCharges”, skewness is removed now, lets again check skewness.

customer\_num.skew()

tenure -0.243325

MonthlyCharges -0.259035

TotalCharges -0.144899

# 3.6 Encoding

0 gender 7043 non-null object

1 SeniorCitizen 7043 non-null int64

2 Partner 7043 non-null object

3 Dependents 7043 non-null object

4 tenure 7043 non-null int64

5 PhoneService 7043 non-null object

6 MultipleLines 7043 non-null object

7 InternetService 7043 non-null object

8 OnlineSecurity 7043 non-null object

9 OnlineBackup 7043 non-null object

10 DeviceProtection 7043 non-null object

11 TechSupport 7043 non-null object

12 StreamingTV 7043 non-null object

13 StreamingMovies 7043 non-null object

14 Contract 7043 non-null object

15 PaperlessBilling 7043 non-null object

16 PaymentMethod 7043 non-null object

17 MonthlyCharges 7043 non-null float64

18 TotalCharges 7043 non-null float64

19 Churn 7043 non-null object

Upto to this point we deal with continous data, now we will do oeration on categorical data,the most important operation is Encoding, we will not do operation such as distribution, outlier removal and skewness removal on categorical features.

In this section. We use label encoding and one hot encoding,

customer\_cat=pd.get\_dummies(customer\_cat,columns=['MultipleLines','InternetService','OnlineSecurity','OnlineBackup','DeviceProtection','TechSupport','StreamingTV','StreamingMovies','Contract','PaymentMethod'])

from sklearn import preprocessing

le = preprocessing.LabelEncoder()

customer\_cat['gender']=le.fit\_transform(customer\_cat['gender'])

customer\_cat['Partner']=le.fit\_transform(customer\_cat['Partner'])

customer\_cat['Dependents']=le.fit\_transform(customer\_cat['Dependents'])

customer\_cat['PhoneService']=le.fit\_transform(customer\_cat['PhoneService'])

customer\_cat['PaperlessBilling']=le.fit\_transform(customer\_cat['PaperlessBilling'])

customer\_cat['Churn']=le.fit\_transform(customer\_cat['Churn'])

for one hot encoding, the feature which contains more than 2 class are passed, and for label encoding the feature which contains two class only, are passed.we can all feature to one hot encoding also, but it will create more columns and we can pass all feature to label encoder, but it will not create binary values for multiclass values. So I used this approach.

customer\_cat.shape

(7043, 38)

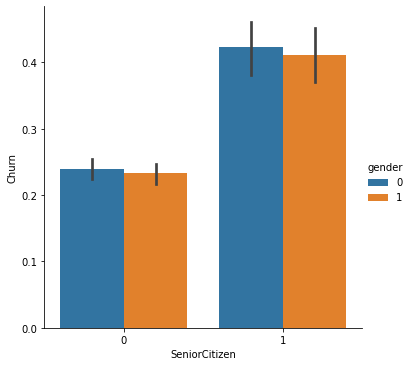
So after encoding number of columns increased upto 38,this is due to one hot encoding, in one hot encoding, if features contains 4 class then it create 4 new columns, in label encoding it will assign 0 to n-1 values to n classes. So now all features of dataframe contains numerical data, the feature which is one hot encoded got datatype as uint8 and for label encoded feature data type is int32.

After every operation we must check that is any null values is created or not and shape also.

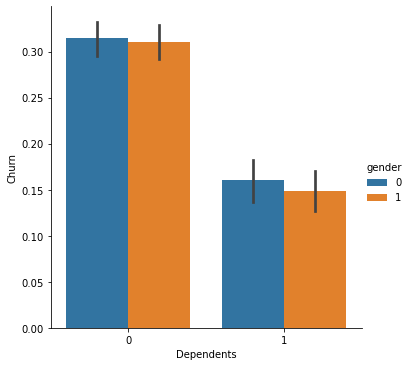
**3.7 Bivariate Analysis**

For categorical feature and categorical label, we preferred catplot for bivariate analysis.it need x and y and data as a compulsory parameter,x is input or feature and y is label and data is customer\_cat. We plot catplot for every categorical feature against y. we use hue =gender, it will classify the data according to gender.and kind=’bar’ so we get bar graph.. here we will print 2 catplot.by understanding these 2 plot, we can easily understand remaining plots.

sns.catplot(x="SeniorCitizen", y="Churn", hue="gender", kind="bar", data=customer\_cat)



sns.catplot(x="Dependents", y="Churn", hue="gender", kind="bar", data=customer\_cat)



Due to use of hue=’gender’, two bargraph is created for class 0 and class 1, for feature “seniorcitizen” class 0 has low churn and for feature “ Dependant” class 0 has high churn values.we can do this procedure for remaining feature also and implement catplot for them.

**3.8 EDA concluding Remark**

In EDA, first we performed univariate analysis of categorical feature ,where we get count of each class in every feature, we did missing value operation on feature “TotalCharges” and changed its datatypes. Next we performed operation on continous data, where we displayed the distplot to see normal distribution of each feature, one feature “TotalCharges” shows right skewness, which is removed using power transformer.we plot boxplot for all feature ,but no outlier detected so no need of zscore and IQR method, then we used one hot encoding and label encoding for categorical feature, in last bivariate analysis is performed,

**4.Pre-processing Pipeline**

**4.1 Scaling**

Standard scaler removes the mean and scales each features to unit variance ,we got all transformed values,generally it is useful for those feature which has normal distribution, here we pass customer\_num to standard scaler.

from sklearn.preprocessing import StandardScaler

scaler=StandardScaler()

x\_scaled=scaler.fit\_transform(customer\_num)

customer\_num1=pd.DataFrame(x\_scaled,columns = customer\_num.columns)

customer\_num1.head()

for standard scaler we pass only continous feature,dataframe “customer\_num” contains only continous feature, we can not perform scaling operation on categorical feature, we get x\_scaled array, this array is then converted to dataframe with name “customer\_num1”.

In next operation we concatenated dataframe ‘customer\_num1’ and ‘customer\_cat’

customer\_new= pd.concat([customer\_num1,customer\_cat],axis=1)

customer\_new is anew dataframe created after all operations and combination of both dataframes.

customer\_new.shape

(7043, 41)

Now total rows are 7043 and columns are 41 now.

**4.2 Oversampling using SMOTE**

customer['Churn'].value\_counts()

No 5174

Yes 1869

Lets see the count of class No and Yes, if ratio between two class is 50: 50 or 40:60 then it is considered as balanced data, if ratio is more than this,then there is class imbalance, here ratio is imbalanced.

x\_1=customer\_new.drop(columns=['Churn'],axis=1)

y\_1=customer\_new['Churn']

from imblearn.over\_sampling import SMOTE

SM=SMOTE()

x\_1,y\_1=SM.fit\_resample(x\_1,y\_1)

1 5174

0 5174

We create x\_1 feature and y\_1 has labels,SMOTE is oversampling technique,which works on low class value and increase the class value , make it equal to high class, both class value are now equal.now due to oversampling the rows will increase. Lets check shape of data.

customer\_new.shape

(10348, 41)

Now we have total 10348 rows,

x=customer\_new.drop(columns=['Churn'],axis=1)

y=customer\_new['Churn']

for train test split we need to separate feature from labels, so x contains only features and y contains only label.

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import classification\_report,accuracy\_score

classification score is used to check performance of classification model,classification score give accuracy which is ratio of true predictions divided by total predictions.classification report contains precision,f1 score and accuracy. Train test split is used to evaluate the performance of algorithm when we have large data,It divide the data into two set,training and testing phase, training data is used for fitting machine learning model and testing data is used for evaluating model performance

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.25,random\_state=MaxRs)

in train test split we pass x and y with test size=0.25 and best random state.

**5.Building Machine Learning Models**

from sklearn.linear\_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.neighbors import KNeighborsClassifier

from sklearn.svm import SVC

from sklearn.ensemble import RandomForestClassifier

from sklearn.ensemble import GradientBoostingClassifier

from xgboost import XGBClassifier

1. Logistics Regression is a algorithm used to predict data value by analysing one or more the independant features or variables
2. Decision Tree builds model in the form of tree structure,it divide the dataset into smaller and smaller subset, we determine the root node,then split the subsets,repeat the process till all instance have same class.
3. KNN store all cases and classify new model based on distance(Euclidean method),classification done by majority of votes neighbors
4. SVC is effective when number of dimensions are greater than number of samples,it uses subset of training points(support vector) and hyperplane to give classification output.
5. RFC is ensemble algorithm which combines more than one algorithm of same or different kind.Random Forest gives outcome based on predictions of number of models by taking mean of output from various trees.
6. Gradient boosting is ensemble algorithm it involves building a strong model by using collection of weaker models.It combine many weak models and make strong predictive model.
7. Extreme Gradient Boosting(XGB) is powerfull algorithm in entire machine learning.All the trees are build parallel in XG boosting.

Lets check the accuracy for each model

1. Log\_reg\_accuracy 84.0742172400464
2. DTC\_accuracy 80.13142636258213
3. KNC\_accuracy 82.75995361422497
4. svc\_accuracy 83.91959798994975
5. RFC\_accuracy 85.7750289911094
6. GBC\_accuracy 83.22381136451487
7. XBC\_accuracy 85.15655199072285

Here we get all model accuracy score , we get highest accuracy for Random forest Classifier and XBC (Extreme Boosting Classifier) , now we will do cross validation for all models.

**Cross validation-**

CV is a technique used to check whether model is overfitted or not, if model accuracy is high, then there is chances of model is overfitted.We use CV score for every model,for this we require model,features and labels, cv is cross validation spliting strategy.finally it take mean of all iteration and return cv score

1. Log\_reg\_cv\_score=82.36465608020865
2. DTC\_CV\_score= 78.32505142627656
3. KNC\_CV\_score= 82.44138571925573
4. svc\_CV\_score= 82.77043917222959
5. RFC\_CV\_score= 85.42775221057106
6. GBC\_CV\_score = 81.00164143802114
7. XBC\_CV\_score= 83.52445462462903

**6.     Concluding Remarks**

Here we get cross validation score for each model. Now we check difference between each model accuracy score with its cv score

Logistic\_regression\_diff 1.7095611598377474

DTC\_diff 1.8063749363055734

KNC\_diff 0.31856789496923454

svc\_diff 1.1491588177201635

RFC\_diff 0.347276780538337

GBC\_diff 2.2221699264937342

XBC\_diff 1.6320973660938165

By observing the difference between accuracy score and cv score for each model , we get less difference for Random Forest Classifier, We get highest Accuracy and CV score for Random forest Classifier.therefore we select Random Forest classifier as a best model, now we will do hyperparameter tuning for Random Forest Classifier.

For Hyperparameter tuning we use GridsearchCV, GridsearchCV is parameter tuning approach which build and evaluate the model performance based on best parameters, we pass parameters of only those model which has less difference between accuracy and cv score. after passing the best parameter we can get best and authenticated accuracy which is not overfitted also.

params={'n\_estimators':[10,15], 'criterion':['entropy'], 'max\_depth':[10,15],

'min\_samples\_leaf':[5,6], 'min\_samples\_split':[10,11]}

Lets understand each parameter one by one.

1. n\_estimators= Random Forest classifier is group multiple decision tree,n\_estimator decides the number of decision tree in classifier, here we give 10 and 15 , it means it will build either 10 or 15 decision tree .
2. criterion=there are two criterion ,’entropy’ and “gini indexing”,used to decide root node.
3. max\_depth= it give the maximum depth upto which tree inside forest grows.high depth increase model accuracy but upto certain limits only, if max\_depth is more than limit, then it causes overfitting.
4. min\_samples\_leaf'= it give the minimum number of sample a node posses after getting split.its default value is 1.
5. Min\_sample\_split= it gives the minimum number of sample as internal node hold to split into further nodes.

We pass this parameter to hyperparameter tuning, we get best value from it, and again we get final accuracy score.

final\_score=83.76497873985312

we get final score in between 83 to 85

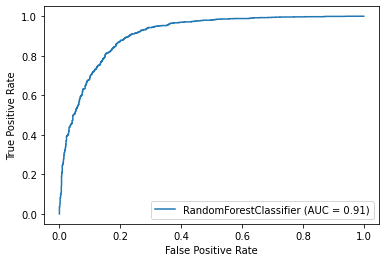
Now we will implement confusion matrix for best model(RFC)

array([[1044, 248],

[ 172, 1123]], dtype=int64)

Here we get array of TP ,TN ,FP and FN, it is used to describe classification model performance, we get parameter such as accuracy ,precision and f1 score from confusion matrix.

Now we implement ROC AUC curve for RFC model



This curve helps us to choose the best model amongst the models for which we have plotted ROC curve.The best model is one which cover the maximum area under it.our Random Forest classifier ROC curve cover almost more than 90 % area under it.

import joblib

joblib.dump(Final\_RFC,"Customer Churn Prediction.pkl")

joblib is replacement to pickle because it is more efficient and carry large numpy array. We saved our model as “Customer Churn Prediction”. This is the final step of our project.