# Customer Churn Analysis for Telecommunications Sector Using Machine Learning

Sudhansu Mandal Aug 24,2021

When a company's customer stop using product & services with that company then its call Customer Churn. Today Era where many players available in market who offers their product and services as a result market becomes competitive for all players So Customer Churn Analysis play very vital role for success of any company/Business.

## ➢ PROBLEM STATEMENT

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.

## ➢ DATA ANALYSIS

- **Data Input for our analysis.**

  For our analysis we have data set from IBM Sample Data Sets with the aim of building and comparing several customer churn prediction models.

  The dataset contains 7043 sample records for our analysis and 21 features columns.

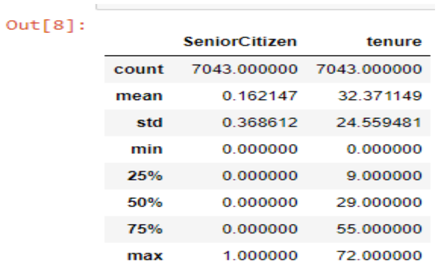
- **Data Set's Features data type.**

  The Dataset' columns/features having float, integer and object type data. Which are the predictor variables or independent variables for our model and also having target variable or dependent variable for our analysis.

- **Statistic Summary of Data set.**

  Since our dataset having both type integer/float and object type columns. So, we will have statistic Summary analysis for our numeric feature and categorical features. Pandas describe () method is very useful to view some basic statistical details like percentile, mean, std etc of a data frame and also provide sense of data distribution for our numeric columns. (As shown in Fig 1.)

**Fig: 1**

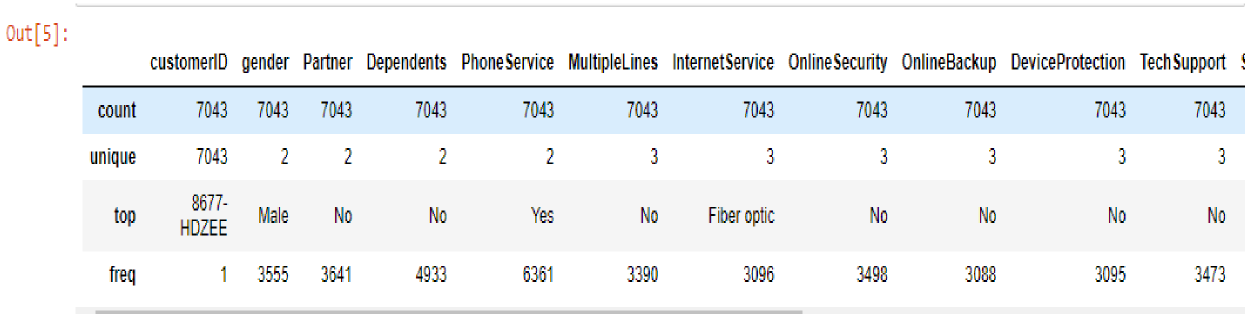


Out[8]:

|  | SeniorCitizen | tenure |
|---|---|---|
| **count** | 7043.000000 | 7043.000000 |
| **mean** | 0.162147 | 32.371149 |
| **std** | 0.368612 | 24.559481 |
| **min** | 0.000000 | 0.000000 |
| **25%** | 0.000000 | 9.000000 |
| **50%** | 0.000000 | 29.000000 |
| **75%** | 0.000000 | 55.000000 |
| **max** | 1.000000 | 72.000000 |

While applying describe () method in categorical columns it provides some basis statistical details like count, unique, top and frequency of the dataset columns. (As shown in Fig 2.)

**Fig: 2**



Out[5]:

|  | customerID | gender | Partner | Dependents | PhoneService | MultipleLines | InternetService | OnlineSecurity | OnlineBackup | DeviceProtection | TechSupport |
|---|---|---|---|---|---|---|---|---|---|---|---|
| count | 7043 | 7043 | 7043 | 7043 | 7043 | 7043 | 7043 | 7043 | 7043 | 7043 | 7043 |
| unique | 7043 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | 3 |
| top | 8677-HDZEE | Male | No | No | Yes | No | Fiber optic | No | No | No | No |
| freq | 1 | 3555 | 3641 | 4933 | 6361 | 3390 | 3096 | 3498 | 3088 | 3095 | 3473 |

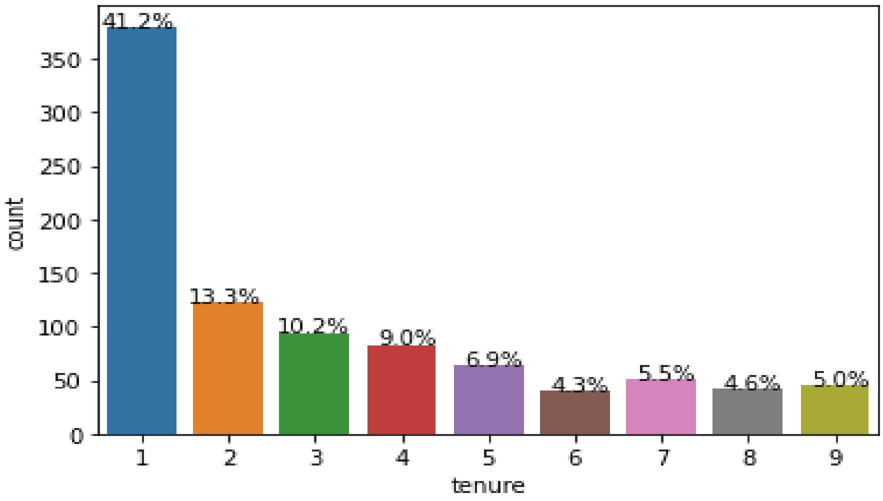
## ➢ <u>EXPLOTERY DATA ANALYSIS(EDA)</u>

Here we will try to visualize those patterns in data which are responsible for Customer Churn. The Purpose of this analysis is to identify the potential factor which are reason for Customer Churn.

- **Tenure**

  As we know that Tenure features in data set represent the customer lifespan in months. We have found very interesting insight that 41% of Customers Churns the services in the first months itself, as shown in Fig 3.

**Fig:3**

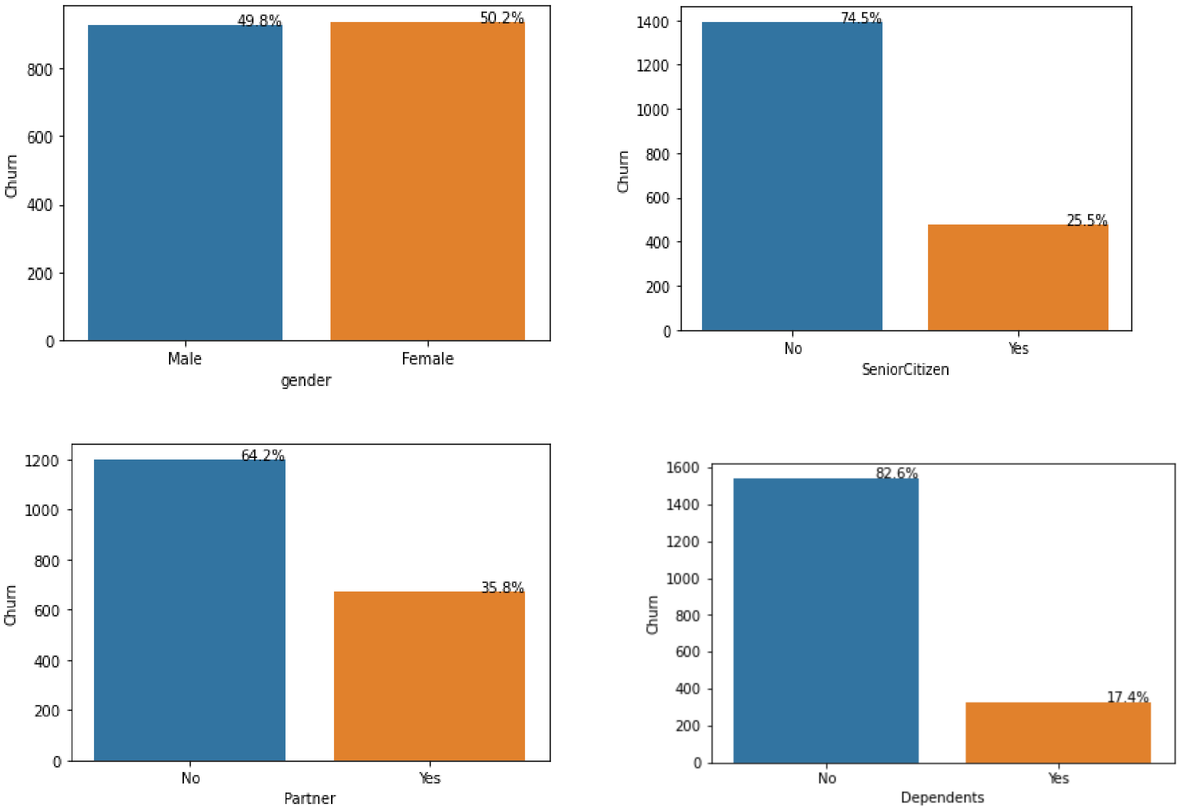


Here we can see that maximum customer churn is happening in the first month itself after starting of service by customer, which may represent that the first-time bad experience, experiencing trail period or prepaid account got expire automatically if no top up recharge is done within the stipulated time.

- **Personal Attribute**

In the data set there are personal attribute features like Gender, Senior Citizen, Partner and Dependents.

**Fig:4**



As shown in Fig 4, we have got valueable insight as mentioned below:

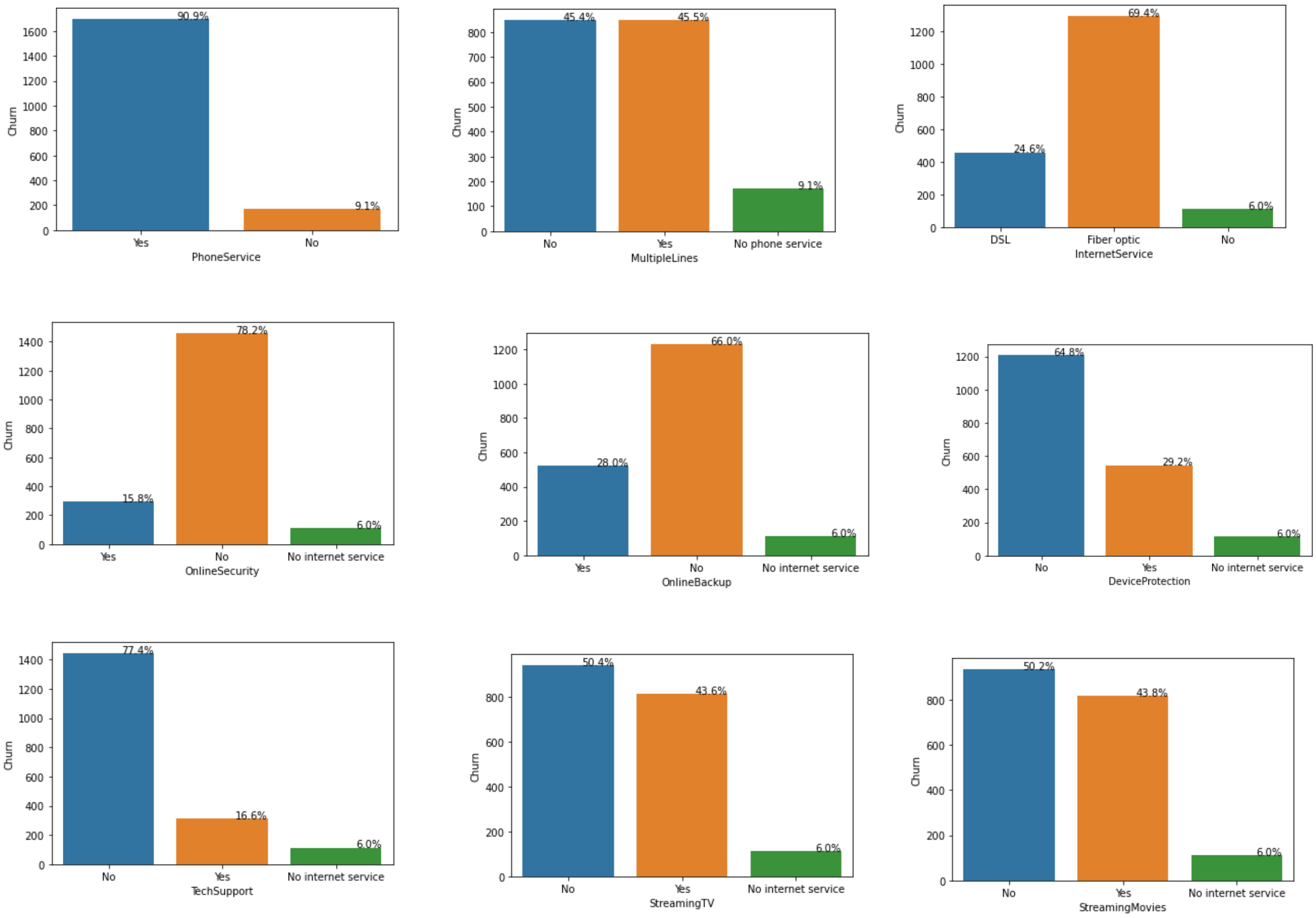
- Senior Citizen is 3 times less Customer Churn.
- Customer Churn is 2 times less for Customer who are partners.
- Customer without dependents 4 times more likely to Churn.

- **Service Attribute**

The Service Attribute describe the services that Customer would like to subscribe like Phone Service,Internet Service,Online Service,Online Backup,Device protection,Tech Support .We can infer from the below graph **(Fig: 5)** that customer churn more likely to happen during which kind of services.

This would help the company to improve their services or to find the reason of Churn while customer using these services.

**Fig: 5**



As shown in Fig 5, we have below insight from service attribute:

- Customer churn is more likely to happen when customer has enabled phone services.
- Churn is more likely to happen fiber optic internet service than DSL.

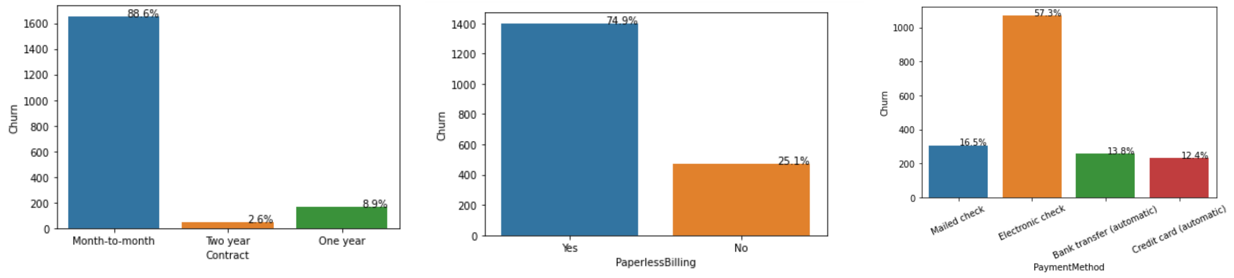
- The Customer who has not enabled online Security ,online Backup and Device protection Services likely to be more churn.

- The Customer who has not enabled tech support services more likely to churn.

- **Contract Attribute**

  Contract attribute indicates contract aspects and gather attributes like Contract,Paperless Billing and Payment method.We will trying to figure out the impact of these attribute in churn .

  **Fig: 6**



  As we can see above Fig: 6, Below are the observation:

  - Month to Month contract more likely to churn, where long term contract having less churn

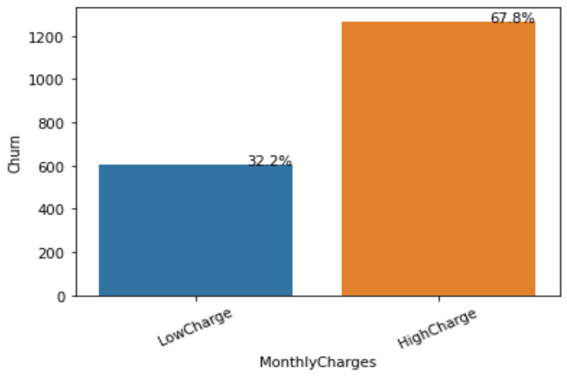
  - The Customer having paperless billing more likely to churn.

  - We can infer from Fig 6, that the customer doing payment through electronic check more likely to churn

- **Charges Attribute:**

  Charges Attribute represent the Charges of services, we have coerced monthly charges to a 2 level low charge and high charge, 0 to 70 considered as low charge and more than 70 has considered as High charge.

  **Fig: 7**

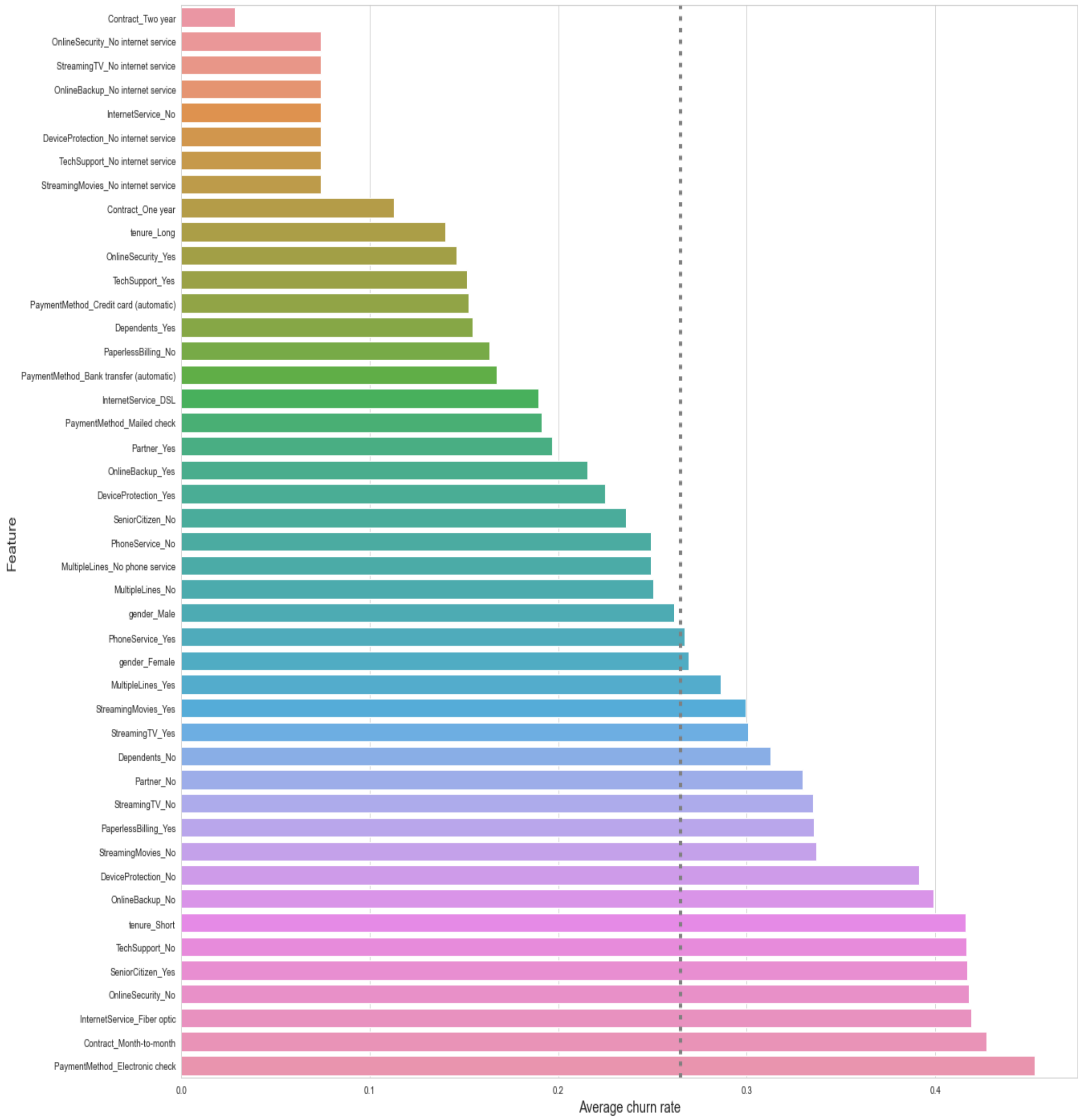


As we can infer from Fig 7 that High charge (more than 70) more likely to churn than low charge(less than 70)

- ## Factor Analysis:

Here we will try to find out the churn rate of all categorical feature variable which will indicate the features with their variable more than and less than the average churn rate.

**Fig: 8**



As shown in Fig: 8, We have observed below mentioned points.

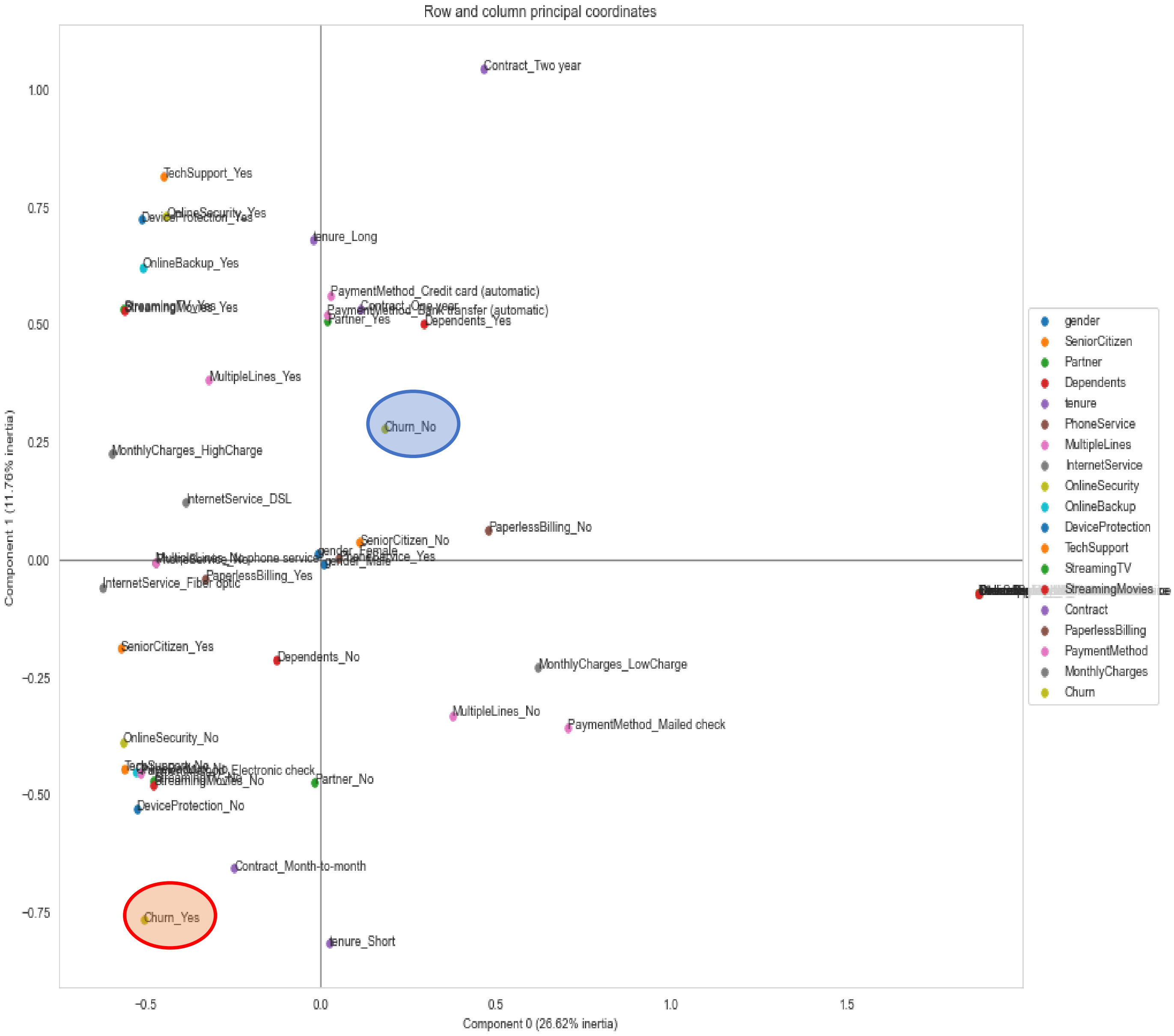
- we can infer from plot that payment method by electronic check, Contract Month to month, Internet Service Fibre optic, tenure less than 24 months, Monthly charges more than 70 having much higher than the average rates of churn, means more likely to churn

- On the other contract one year and 2-year, tenure more than 24 months, payment method credit card/online transfer, monthly charge with low rate having much lower than the average rates of churn.

## • Multiple Correspondence Analysis

Here we will try with another technique to know the features which impact on our dependent variable more likely to Churn and less likely to Churn. Multiple Correspondence Analysis (MCA) is an extension of corre- spondence analysis (CA) which allows one to analyse the pattern of relationships of several categorical dependent variables.

**Fig: 9**



As shown in Fig: 9, We can infer that those factors which are nearby of churn yes, causes for more churn in telecom choices. Like Contract monthly, payment method electronic check, senior citizen etc. Factors like payment method credit card, monthly high charges more than 70, paperless billing, internet service DSL type having less churn.

## ➢ DATA PRE-PROCESSING

The Machine Learning Model Performance is not only depending on models and hyperparameters but also how we process and provide input of different types of variables to the model. Pre-processing of categorical variable is necessary steps as machine learning only accept numerical variables.

There are two types of Categorical data:

Ordinal Data:

Ordinal Data are those category where categorical variables has an inherent order.

Nominal Data:

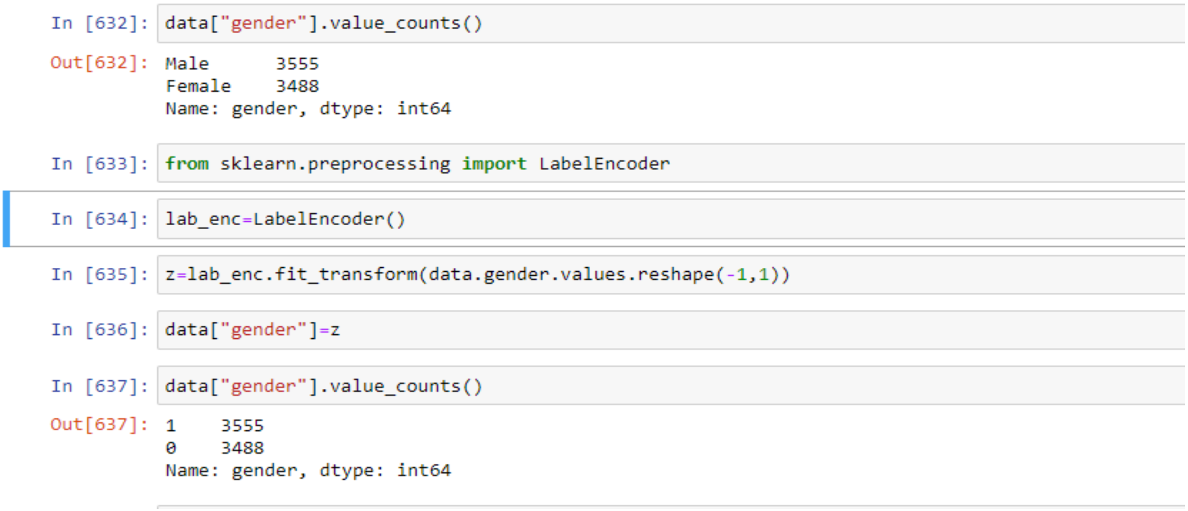
Nominal Data are referring to category where categorical variable don't have an inherent order.

Mentioned Below are the types of Encoding technique which we have used in this data:

- Label Encoder:

we have encoded Gender, Senior Citizen, Partner, Dependents, Phone Service, Paperless Billing categorical features to numeric variable by using Label Encoder. These Categorical Features variables are responded either Yes or No. As example shown in Fig 10.

**Fig: 10**

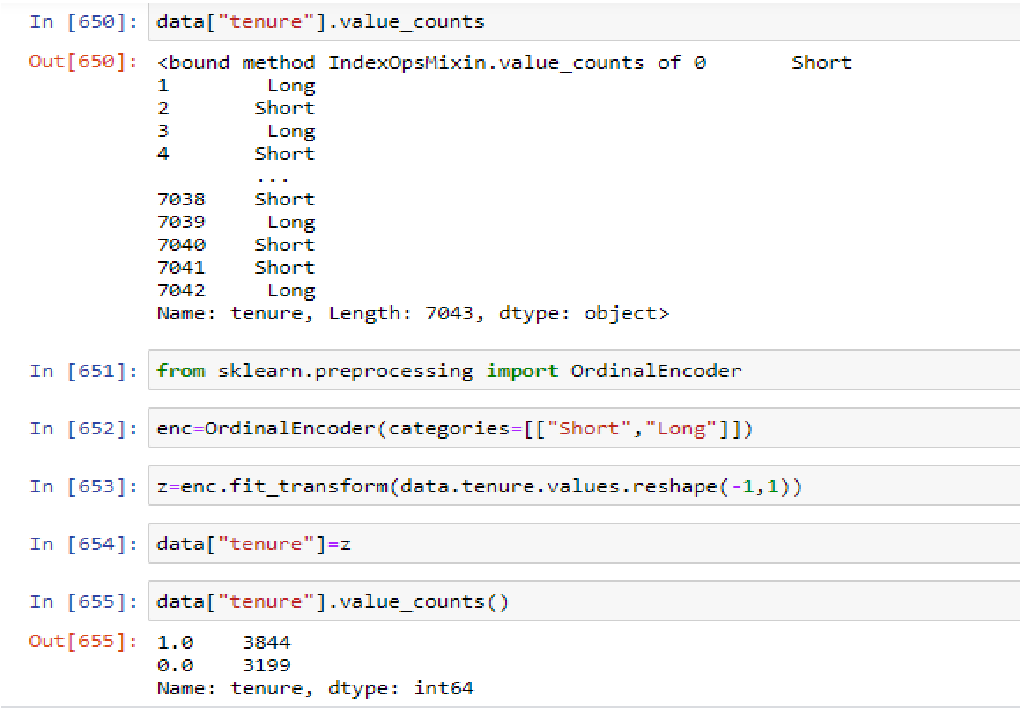


```
In [632]: data["gender"].value_counts()

Out[632]: Male      3555
          Female    3488
          Name: gender, dtype: int64

In [633]: from sklearn.preprocessing import LabelEncoder

In [634]: lab_enc=LabelEncoder()

In [635]: z=lab_enc.fit_transform(data.gender.values.reshape(-1,1))

In [636]: data["gender"]=z

In [637]: data["gender"].value_counts()

Out[637]: 1    3555
          0    3488
          Name: gender, dtype: int64
```

- Ordinal Encoder:

We have encoded tenure, Internet Service and monthly charges categorical features to numerical features by using Ordinal Encoder. Since, these categories variable are in order. As example shown in Fig 11.

**Fig: 11**

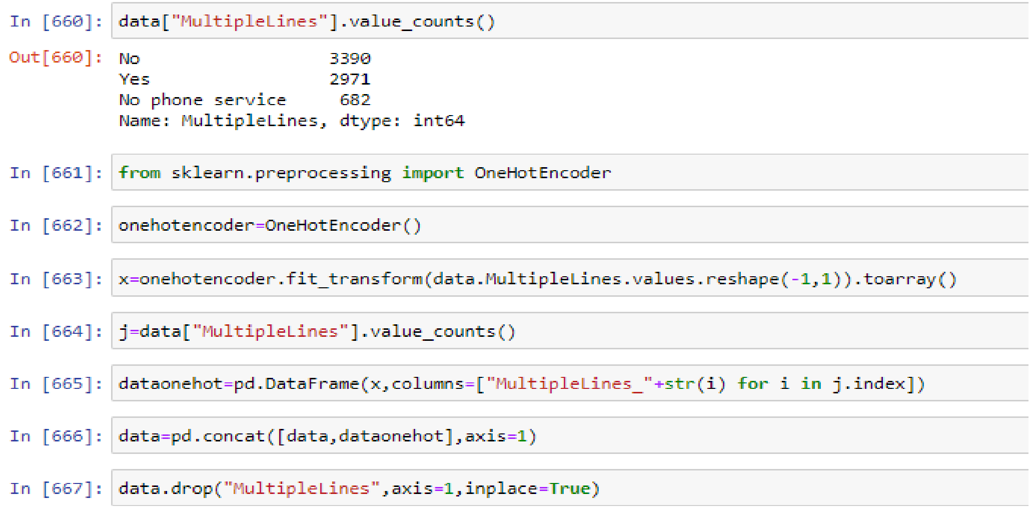


```
In [650]: data["tenure"].value_counts
Out[650]: <bound method IndexOpsMixin.value_counts of 0       Short
          1        Long
          2       Short
          3        Long
          4       Short
                  ...
          7038    Short
          7039     Long
          7040    Short
          7041    Short
          7042     Long
          Name: tenure, Length: 7043, dtype: object>

In [651]: from sklearn.preprocessing import OrdinalEncoder

In [652]: enc=OrdinalEncoder(categories=[["Short","Long"]])

In [653]: z=enc.fit_transform(data.tenure.values.reshape(-1,1))

In [654]: data["tenure"]=z

In [655]: data["tenure"].value_counts()
Out[655]: 1.0    3844
          0.0    3199
          Name: tenure, dtype: int64
```

- One Hot Encoding:

We have encoded Multiple lines, Internet Service, Online Security, Online Backup, Device Protection, Tech Support, Streaming TV, Streaming Movies, Contract, Payment Method categorical feature to numerical features with One Hot Encoding, since there are multiple variable with no order. As example shown in Fig 12.

**Fig: 12**



```
In [660]: data["MultipleLines"].value_counts()
Out[660]: No                  3390
          Yes                 2971
          No phone service     682
          Name: MultipleLines, dtype: int64

In [661]: from sklearn.preprocessing import OneHotEncoder

In [662]: onehotencoder=OneHotEncoder()

In [663]: x=onehotencoder.fit_transform(data.MultipleLines.values.reshape(-1,1)).toarray()

In [664]: j=data["MultipleLines"].value_counts()

In [665]: dataonehot=pd.DataFrame(x,columns=["MultipleLines_"+str(i) for i in j.index])

In [666]: data=pd.concat([data,dataonehot],axis=1)

In [667]: data.drop("MultipleLines",axis=1,inplace=True)
```

- ## Dealing Imbalanced Target Variable

After analysing our target variable, we have seen that our Target Variable is not balanced. As we know that one of the biggest constraints is humongous data and its distribution. Customer Churn in this data set is significantly lesser than Customer Retention i.e. its around 26% of the total sample observation (as shown in Fig 13). So, we have to improve identification of the rare minority class as opposed to achieving higher accuracy level.

Machine Learning algorithms produce unsatisfactory classifiers when dataset target variable is imbalanced.

Usually there are mainly two technique to handle imbalanced data set as mentioned below:

- ✓ Data label Technique/Resampling Technique
  - Random Under Sampling
  - Random Over Sampling
  - Cluster Based Over Sampling
- ✓ Algorithms Ensemble Technique
  - Bagging Based
  - Boosting -Based

Here in this project we will used Random Over Sampling data label Technique to deal with imbalanced data set to avoid any loss of data. As shown in Fig 14.

**Fig: 13**

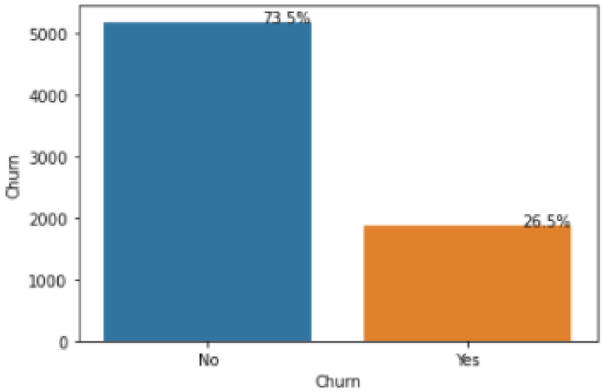
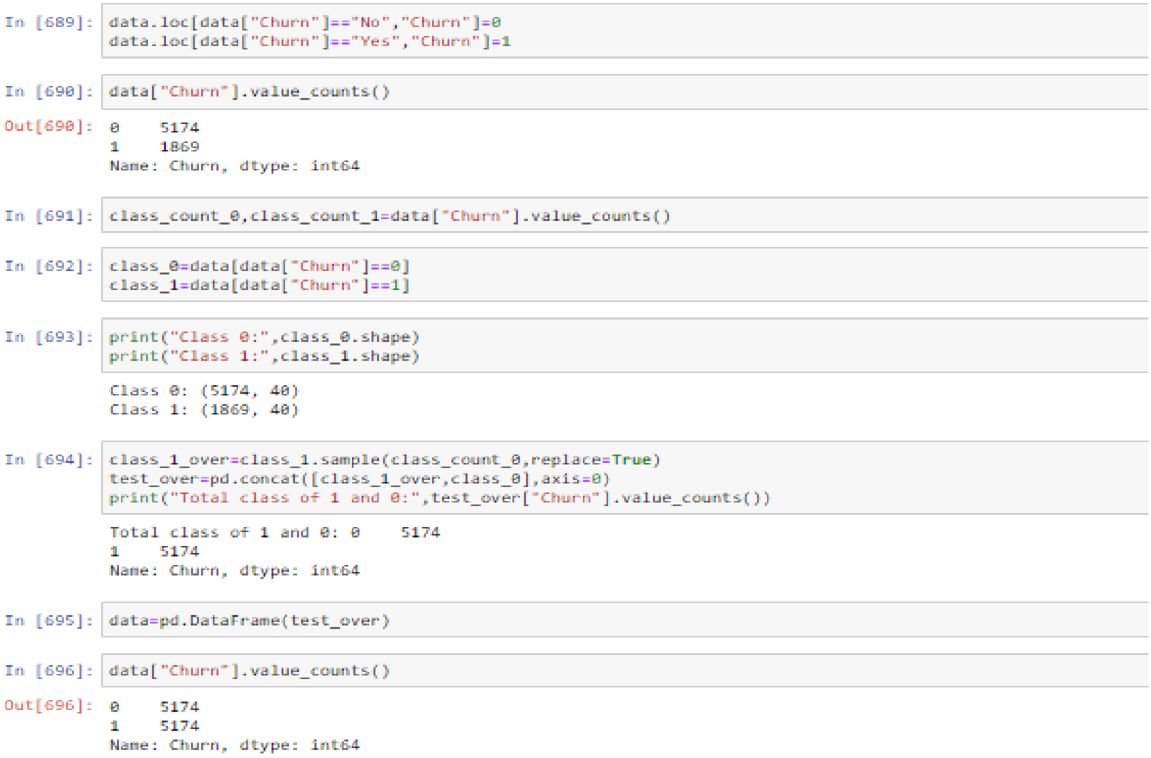


**Fig: 14**

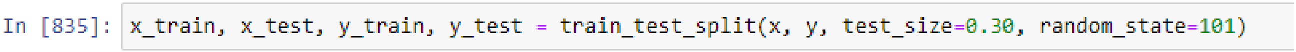


```
In [689]: data.loc[data["Churn"]=="No","Churn"]=0
          data.loc[data["Churn"]=="Yes","Churn"]=1

In [690]: data["Churn"].value_counts()

Out[690]: 0    5174
          1    1869
          Name: Churn, dtype: int64

In [691]: class_count_0,class_count_1=data["Churn"].value_counts()

In [692]: class_0=data[data["Churn"]==0]
          class_1=data[data["Churn"]==1]

In [693]: print("Class 0:",class_0.shape)
          print("Class 1:",class_1.shape)

          Class 0: (5174, 40)
          Class 1: (1869, 40)

In [694]: class_1_over=class_1.sample(class_count_0,replace=True)
          test_over=pd.concat([class_1_over,class_0],axis=0)
          print("Total class of 1 and 0:",test_over["Churn"].value_counts())

          Total class of 1 and 0: 0    5174
          1    5174
          Name: Churn, dtype: int64

In [695]: data=pd.DataFrame(test_over)

In [696]: data["Churn"].value_counts()

Out[696]: 0    5174
          1    5174
          Name: Churn, dtype: int64
```

- ## Splitting Train and Test Dataset

The dataset has been split into 70% for training and 30% for testing (shown in fig 15). The training dataset would be using to generate the model the chosen algorithms will use when exposed to new unseen data. The test data set is the final dataset which would be using to measure model performance based on some metrics.

**Fig:15**



```
In [835]: x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.30, random_state=101)
```

## ➢ <u>BUILDING MACHINE LEARNING MODEL</u>

- ## Model:

For this analysis, have applied different Classifier model to analyse and compare their accuracy score. Those models are listed below:

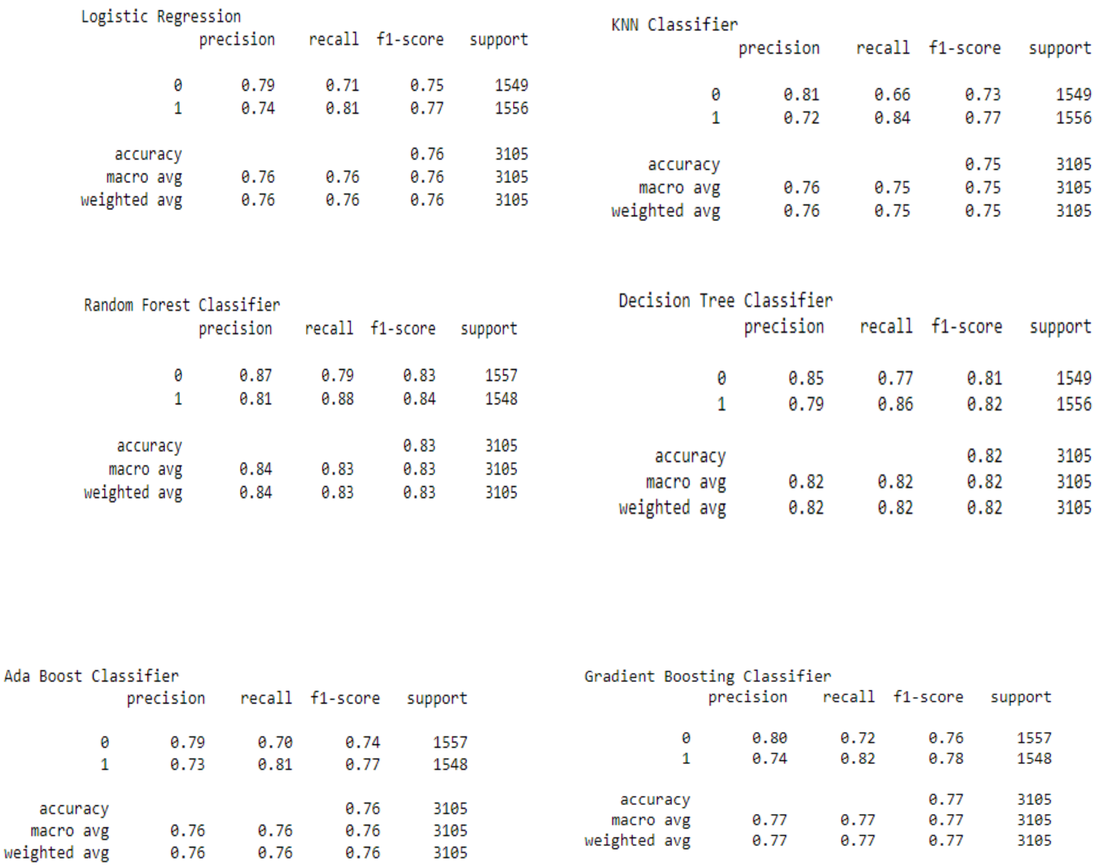
- ✓ Logistic Regression
- ✓ KNN Classifier
- ✓ Random Forest Classifier
- ✓ Decision Tree Classifier
- ✓ Ada Booster Classifier
- ✓ Gradient Boosting Classifier

After applying mentioned above model to our training and test dataset, we are getting accuracy score for Logistic Regression is 75.78%, KNN Classifier 75.49%, Random Forest Classifier is 83.41, Decision Tree Classification is 81.15%, Ada Booster Classifier is 75.65%, Gradient Boosting Classifier is 76.77%.3.

- ## Evaluation Matrix:

As mentions in Fig 16, The classification report for each model which indicate the Precision, Recall, F1 Score.

**Fig: 16**



```
Logistic Regression
              precision    recall  f1-score   support

           0       0.79      0.71      0.75      1549
           1       0.74      0.81      0.77      1556

    accuracy                           0.76      3105
   macro avg       0.76      0.76      0.76      3105
weighted avg       0.76      0.76      0.76      3105
```

```
KNN Classifier
              precision    recall  f1-score   support

           0       0.81      0.66      0.73      1549
           1       0.72      0.84      0.77      1556

    accuracy                           0.75      3105
   macro avg       0.76      0.75      0.75      3105
weighted avg       0.76      0.75      0.75      3105
```

```
Random Forest Classifier
              precision    recall  f1-score   support

           0       0.87      0.79      0.83      1557
           1       0.81      0.88      0.84      1548

    accuracy                           0.83      3105
   macro avg       0.84      0.83      0.83      3105
weighted avg       0.84      0.83      0.83      3105
```

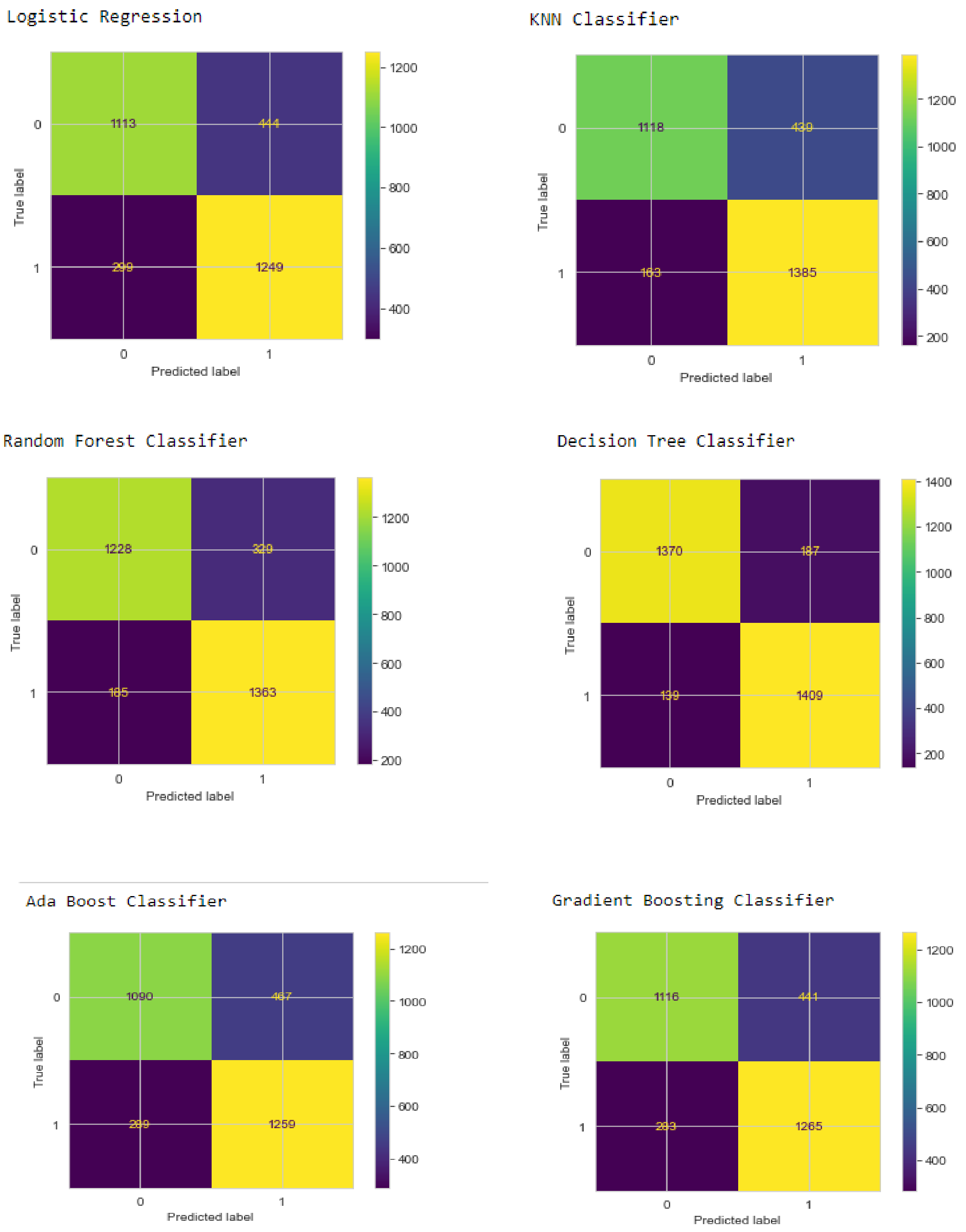
```
Decision Tree Classifier
              precision    recall  f1-score   support

           0       0.85      0.77      0.81      1549
           1       0.79      0.86      0.82      1556

    accuracy                           0.82      3105
   macro avg       0.82      0.82      0.82      3105
weighted avg       0.82      0.82      0.82      3105
```

```
Ada Boost Classifier
              precision    recall  f1-score   support

           0       0.79      0.70      0.74      1557
           1       0.73      0.81      0.77      1548

    accuracy                           0.76      3105
   macro avg       0.76      0.76      0.76      3105
weighted avg       0.76      0.76      0.76      3105
```

```
Gradient Boosting Classifier
              precision    recall  f1-score   support

           0       0.80      0.72      0.76      1557
           1       0.74      0.82      0.78      1548

    accuracy                           0.77      3105
   macro avg       0.77      0.77      0.77      3105
weighted avg       0.77      0.77      0.77      3105
```

As we can see in the below Fig 17, the confusion matrix for each model, which indicate the False Positive, True Positive, False Negative, True Negative with True label and Predicted label for each class.

**Fig: 17**

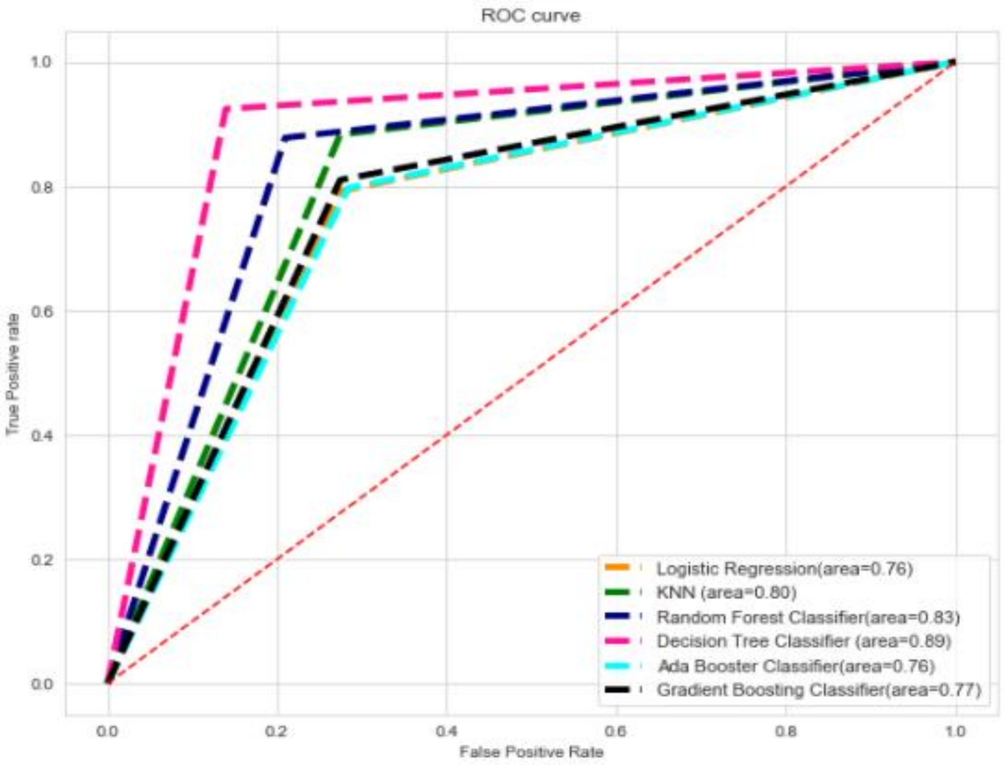


- ## Choosing Best Model Using ROC AUC Curve:

In this Section, we will evaluate and validate how good is our machine learning model. So, I can decide which model can choose for further hyper tuning to get best result out of it.

As shown in Fig 18, ROC AUC Curve helps us to identify or visualize how good machine learning classifier is performing.

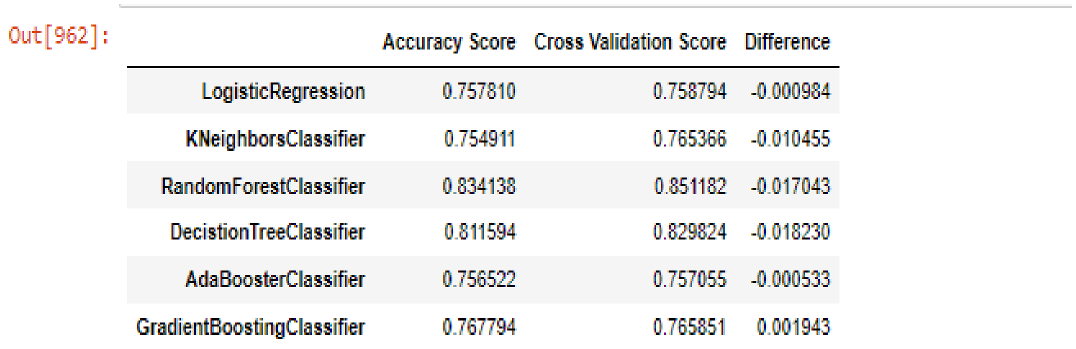
**Fig: 18**



## • Choosing Best Model by Comparing Cross Validation Score:

We have seen that Fig 18, According to ROC AUC Curve Decision Tree Classifier model is performing well in this dataset. We can also select the final model based on the difference between Validation Score and Accuracy Score to ensure that model is not overfitting and has been checked at difference cross validation as shown in Fig 19.

**Fig: 19**



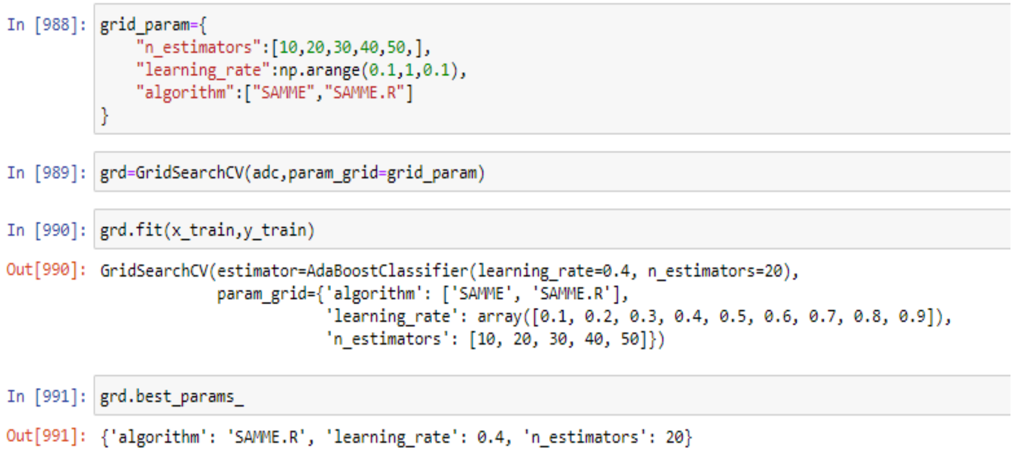
Out[962]:

| | Accuracy Score | Cross Validation Score | Difference |
|---|---|---|---|
| LogisticRegression | 0.757810 | 0.758794 | -0.000984 |
| KNeighborsClassifier | 0.754911 | 0.765366 | -0.010455 |
| RandomForestClassifier | 0.834138 | 0.851182 | -0.017043 |
| DecistionTreeClassifier | 0.811594 | 0.829824 | -0.018230 |
| AdaBoosterClassifier | 0.756522 | 0.757055 | -0.000533 |
| GradientBoostingClassifier | 0.767794 | 0.765851 | 0.001943 |

## • Hyper tuning the best model:

After Checking Cross Validation Score and ROC AUC Curve the AdaBooster Classifier model is performing well, so we will try to improve model accuracy with hyper tuning as shown in Fig 20.

**Fig: 20**



```
In [988]: grid_param={
              "n_estimators":[10,20,30,40,50,],
              "learning_rate":np.arange(0.1,1,0.1),
              "algorithm":["SAMME","SAMME.R"]
          }

In [989]: grd=GridSearchCV(adc,param_grid=grid_param)

In [990]: grd.fit(x_train,y_train)

Out[990]: GridSearchCV(estimator=AdaBoostClassifier(learning_rate=0.4, n_estimators=20),
                       param_grid={'algorithm': ['SAMME', 'SAMME.R'],
                                   'learning_rate': array([0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]),
                                   'n_estimators': [10, 20, 30, 40, 50]})

In [991]: grd.best_params_

Out[991]: {'algorithm': 'SAMME.R', 'learning_rate': 0.4, 'n_estimators': 20}
```

After doing hyper tuning of Ada Booster Classifier model, the accuracy score has increased marginally from 75.65% to 75.68%.

## ➢ CONCLUSION

We have observed from Factor Analysis and Multiple Correspondence Analysis that the features variable like Contract monthly, Payment method through electronic check, Senior Citizen, Internet service Fibre optic, tenure less than 24 months, Monthly Charges more than 70 are more likely to Churn.

On the other side, features like Contract one year and two-year, tenure more than 24 months, payment method credit card/online transfer, Monthly Charge with low rate are less likely to Churn.

After training various model like Logistic Regression, KNN Classifier, Random Forest Classifier, Decision Tree Classifier, Ada Booster Classifier and Gradient Boosting Classifier, have seen that based on AUC ROC Curve, Cross Validation Score Ada Booster Classifier is performing better than another model.