```
In [2]:
        # Import Libraries
            import numpy as np
            import pandas as pd
            import seaborn as sns
            import warnings
            warnings.filterwarnings('ignore')
            import os
            from sklearn import preprocessing
            from sklearn.preprocessing import LabelEncoder
            from sklearn.model selection import train test split
            from sklearn.ensemble import RandomForestClassifier
            from sklearn.metrics import confusion matrix
            from sklearn.metrics import roc curve
            from sklearn.linear model import LogisticRegression
            import matplotlib
            import matplotlib.ticker as mtick
            import matplotlib.pyplot as plt
            %matplotlib inline
            from mpl_toolkits.mplot3d import Axes3D
            !pip install chart-studio
            import chart_studio.plotly as py
            from plotly import __version__
            from IPython.display import display, HTML
```

```
Requirement already satisfied: chart-studio in c:\users\mumitul\anaconda3\l
ib\site-packages (1.1.0)
Requirement already satisfied: six in c:\users\mumitul\anaconda3\lib\site-p
ackages (from chart-studio) (1.15.0)
Requirement already satisfied: retrying>=1.3.3 in c:\users\mumitul\anaconda
3\lib\site-packages (from chart-studio) (1.3.3)
Requirement already satisfied: requests in c:\users\mumitul\anaconda3\lib\s
ite-packages (from chart-studio) (2.24.0)
Requirement already satisfied: plotly in c:\users\mumitul\anaconda3\lib\sit
e-packages (from chart-studio) (4.14.1)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\mumitul\anaco
nda3\lib\site-packages (from requests->chart-studio) (2020.6.20)
Requirement already satisfied: idna<3,>=2.5 in c:\users\mumitul\anaconda3\l
ib\site-packages (from requests->chart-studio) (2.10)
Requirement already satisfied: chardet<4,>=3.0.2 in c:\users\mumitul\anacon
da3\lib\site-packages (from requests->chart-studio) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in
c:\users\mumitul\anaconda3\lib\site-packages (from requests->chart-studio)
(1.25.9)
```

```
In [3]:  M df = pd.read_csv('Churn_Data.csv')
```

## In [4]: ► df.head()

### Out[4]:

MultipleLin€	PhoneService	tenure	Dependents	Partner	SeniorCitizen	gender	customerID	
No phor servic	No	1	No	Yes	0	Female	7590- VHVEG	0
Ν	Yes	34	No	No	0	Male	5575- GNVDE	1
Ν	Yes	2	No	No	0	Male	3668- QPYBK	2
No phor servic	No	45	No	No	0	Male	7795- CFOCW	3
٨	Yes	2	No	No	0	Female	9237- HQITU	4

5 rows × 21 columns

# In [5]: ► df.dtypes

Out[5]: customerID object gender object SeniorCitizen int64 Partner object Dependents object tenure int64 PhoneService object MultipleLines object InternetService object OnlineSecurity object OnlineBackup object DeviceProtection object TechSupport object StreamingTV object StreamingMovies object Contract object PaperlessBilling object PaymentMethod object MonthlyCharges float64 TotalCharges object Churn object dtype: object

### In [6]: ▶ df.shape

Out[6]: (7043, 21)

```
In [7]:
         ▶ | for item in df.columns:
                print(item)
                print (df[item].unique())
            customerID
            ['7590-VHVEG' '5575-GNVDE' '3668-QPYBK' ... '4801-JZAZL' '8361-LTMKD'
             '3186-AJIEK']
            gender
            ['Female' 'Male']
            SeniorCitizen
            [0 1]
            Partner
            ['Yes' 'No']
            Dependents
            ['No' 'Yes']
            tenure
            [ 1 34  2 45  8 22 10 28 62 13 16 58 49 25 69 52 71 21 12 30 47 72 17 27
              5 46 11 70 63 43 15 60 18 66 9 3 31 50 64 56 7 42 35 48 29 65 38 68
             32 55 37 36 41 6 4 33 67 23 57 61 14 20 53 40 59 24 44 19 54 51 26 0
             39]
            PhoneService
            ['No' 'Yes']
            MultipleLines
            ['No phone service' 'No' 'Yes']
            InternetService
            ['DSL' 'Fiber optic' 'No']
            OnlineSecurity
            ['No' 'Yes' 'No internet service']
            OnlineBackup
            ['Yes' 'No' 'No internet service']
            DeviceProtection
            ['No' 'Yes' 'No internet service']
            TechSupport
            ['No' 'Yes' 'No internet service']
            StreamingTV
            ['No' 'Yes' 'No internet service']
            StreamingMovies
            ['No' 'Yes' 'No internet service']
            Contract
            ['Month-to-month' 'One year' 'Two year']
            PaperlessBilling
            ['Yes' 'No']
            PaymentMethod
            ['Electronic check' 'Mailed check' 'Bank transfer (automatic)'
             'Credit card (automatic)']
            MonthlyCharges
            [29.85 56.95 53.85 ... 63.1 44.2 78.7]
            TotalCharges
            ['29.85' '1889.5' '108.15' ... '346.45' '306.6' '6844.5']
```

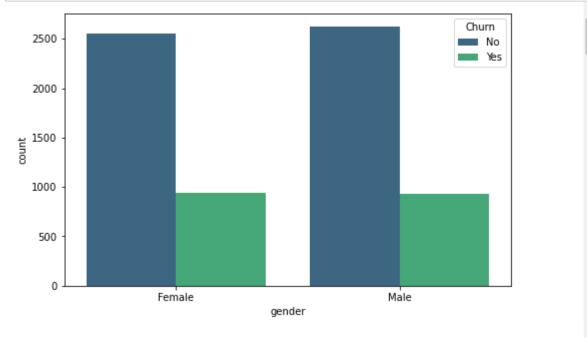
Churn

['No' 'Yes']

# In [8]: ► df.describe()

### Out[8]:

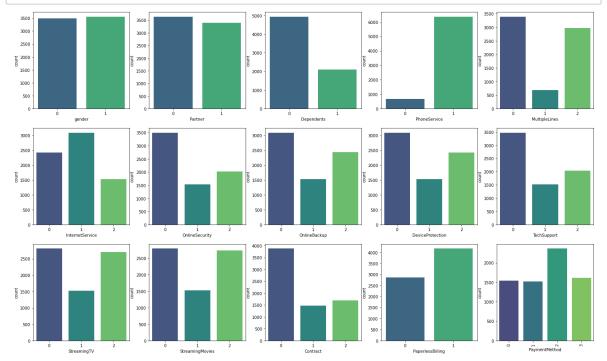
	SeniorCitizen	tenure	MonthlyCharges
count	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692
std	0.368612	24.559481	30.090047
min	0.000000	0.000000	18.250000
25%	0.000000	9.000000	35.500000
50%	0.000000	29.000000	70.350000
75%	0.000000	55.000000	89.850000
max	1.000000	72.000000	118.750000

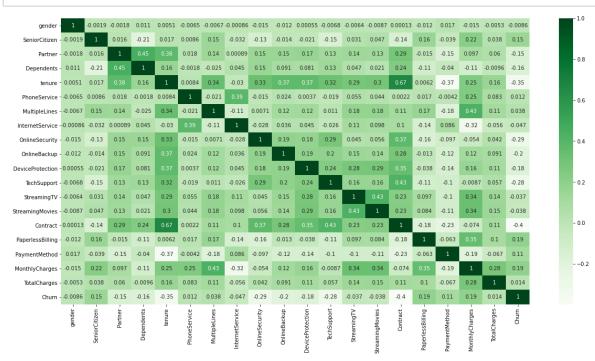


Out[10]: 18

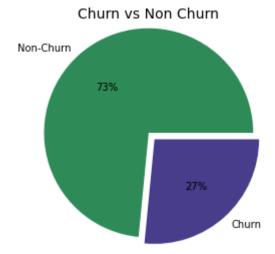
```
In [13]:
          df.info()
             <class 'pandas.core.frame.DataFrame'>
             RangeIndex: 7043 entries, 0 to 7042
             Data columns (total 21 columns):
                                    Non-Null Count Dtype
              #
                  Column
             - - -
                  -----
                                    -----
                                                    _ _ _ _
              0
                                    7043 non-null
                                                    object
                  customerID
              1
                  gender
                                    7043 non-null
                                                    int32
              2
                  SeniorCitizen
                                    7043 non-null
                                                    int64
              3
                  Partner
                                    7043 non-null
                                                    int32
              4
                                    7043 non-null
                  Dependents
                                                    int32
              5
                  tenure
                                    7043 non-null
                                                    int64
              6
                                    7043 non-null
                  PhoneService
                                                    int32
              7
                  MultipleLines
                                    7043 non-null
                                                    int32
              8
                  InternetService
                                    7043 non-null
                                                    int32
              9
                  OnlineSecurity
                                    7043 non-null
                                                    int32
              10
                                    7043 non-null
                                                    int32
                 OnlineBackup
              11
                 DeviceProtection 7043 non-null
                                                    int32
              12
                 TechSupport
                                    7043 non-null
                                                    int32
              13
                                    7043 non-null
                 StreamingTV
                                                    int32
              14
                 StreamingMovies
                                    7043 non-null
                                                    int32
              15
                 Contract
                                    7043 non-null
                                                    int32
                 PaperlessBilling 7043 non-null
                                                    int32
              17
                 PaymentMethod
                                    7043 non-null
                                                    int32
              18
                 MonthlyCharges
                                    7043 non-null
                                                    float64
              19
                                                    int32
                 TotalCharges
                                    7043 non-null
              20
                 Churn
                                    7043 non-null
                                                    int32
             dtypes: float64(1), int32(17), int64(2), object(1)
             memory usage: 687.9+ KB
             df['TotalCharges'] = pd.to numeric(df['TotalCharges'])
In [14]:
             def uni(columnname):
                 print(columnname,"--" ,df[columnname].unique())
In [15]:
          uni(dfobject.columns[i])
             gender -- [0 1]
             Partner -- [1 0]
             Dependents -- [0 1]
             PhoneService -- [0 1]
             MultipleLines -- [1 0 2]
             InternetService -- [0 1 2]
             OnlineSecurity -- [0 2 1]
             OnlineBackup -- [2 0 1]
             DeviceProtection -- [0 2 1]
             TechSupport -- [0 2 1]
             StreamingTV -- [0 2 1]
             StreamingMovies -- [0 2 1]
             Contract -- [0 1 2]
             PaperlessBilling -- [1 0]
             PaymentMethod -- [2 3 0 1]
             TotalCharges -- [2505 1466 157 ... 2994 2660 5407]
             Churn -- [0 1]
```

fig, axes = plt.subplots(nrows = 3,ncols = 5,figsize = (25,15)) In [16]: sns.countplot(x = "gender", data = df,palette="viridis", ax=axes[0][0]) sns.countplot(x = "Partner", data = df,palette="viridis", ax=axes[0][1]) sns.countplot(x = "Dependents", data = df,palette="viridis", ax=axes[0][2]) sns.countplot(x = "PhoneService", data = df,palette="viridis", ax=axes[0][3]) sns.countplot(x = "MultipleLines", data = df,palette="viridis", ax=axes[0][4] sns.countplot(x = "InternetService", data = df,palette="viridis", ax=axes[1][ sns.countplot(x = "OnlineSecurity", data = df,palette="viridis", ax=axes[1][1 sns.countplot(x = "OnlineBackup", data = df,palette="viridis", ax=axes[1][2]) sns.countplot(x = "DeviceProtection", data = df,palette="viridis", ax=axes[1] sns.countplot(x = "TechSupport", data = df,palette="viridis", ax=axes[1][4]) sns.countplot(x = "StreamingTV", data = df,palette="viridis", ax=axes[2][0]) sns.countplot(x = "StreamingMovies", data = df,palette="viridis", ax=axes[2][ sns.countplot(x = "Contract", data = df,palette="viridis", ax=axes[2][2]) sns.countplot(x = "PaperlessBilling",palette="viridis", data = df, ax=axes[2] ax = sns.countplot(x = "PaymentMethod", data = df,palette="viridis", ax=axes[ ax.set xticklabels(ax.get xticklabels(),rotation=90) plt.show(fig)





Out[18]: Text(0.5, 1.0, 'Churn vs Non Churn')



In [21]: N X = df1.drop('Churn', 1)
y = df1['Churn']

```
In [19]:
          # Drop Customer Id for modelling
             df1 = df.drop(['customerID'], axis = 1)
          ▶ | from sklearn.tree import DecisionTreeClassifier
In [20]:
             from sklearn.ensemble import RandomForestClassifier
             from sklearn.naive_bayes import GaussianNB
             from sklearn.neighbors import KNeighborsClassifier
             from sklearn.svm import SVC
             from sklearn.neural_network import MLPClassifier
             from sklearn.ensemble import AdaBoostClassifier
             from sklearn.ensemble import GradientBoostingClassifier
             from sklearn.ensemble import ExtraTreesClassifier
             from sklearn.linear model import LogisticRegression
             from sklearn.model_selection import train_test_split
             from sklearn.metrics import accuracy_score
             from sklearn.preprocessing import LabelEncoder
```

### In [22]: ▶ pip install xgboost

Requirement already satisfied: xgboost in c:\users\mumitul\anaconda3\lib\si te-packages (1.3.0.post0)

Requirement already satisfied: scipy in c:\users\mumitul\anaconda3\lib\site -packages (from xgboost) (1.5.0)

Requirement already satisfied: numpy in c:\users\mumitul\anaconda3\lib\site -packages (from xgboost) (1.18.5)

Note: you may need to restart the kernel to use updated packages.

## In [23]: ▶ pip install --upgrade xgboost

Requirement already up-to-date: xgboost in c:\users\mumitul\anaconda3\lib\s ite-packages (1.3.0.post0)

Requirement already satisfied, skipping upgrade: scipy in c:\users\mumitul \anaconda3\lib\site-packages (from xgboost) (1.5.0)

Requirement already satisfied, skipping upgrade: numpy in c:\users\mumitul \anaconda3\lib\site-packages (from xgboost) (1.18.5)

Note: you may need to restart the kernel to use updated packages.

### In [24]: ▶ from xgboost import XGBClassifier

```
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = 0.30, ran
In [25]:
             classifiers = [['DecisionTree :',DecisionTreeClassifier()],
                             ['RandomForest :',RandomForestClassifier()],
['Naive Bayes :', GaussianNB()],
                             ['KNeighbours:', KNeighborsClassifier()],
                             ['SVM :', SVC()],
                             ['LogisticRegression:', LogisticRegression(max iter=500)],
                             ['Neural Network :', MLPClassifier()],
                             ['ExtraTreesClassifier:', ExtraTreesClassifier()],
                             ['AdaBoostClassifier:', AdaBoostClassifier()],
                             ['XGBoost :', XGBClassifier(use_label_encoder=False, disable_d
                             ['GradientBoostingClassifier: ', GradientBoostingClassifier()]
             predictions df = pd.DataFrame()
             predictions_df['actual_labels'] = y_test
             for name, classifier in classifiers:
                 classifier = classifier
                  classifier.fit(X train, y train)
                 predictions = classifier.predict(X test)
                  predictions_df[name.strip(" :")] = predictions
                  print(name, accuracy_score(y_test, predictions))
```

DecisionTree : 0.7198296261239944 RandomForest : 0.792238523426408 Naive Bayes : 0.7340274491244676 KNeighbours : 0.7458589682915286

SVM : 0.7321344060577378

LogisticRegression: 0.7893989588263133
Neural Network: 0.6715570279223853
ExtraTreesClassifier: 0.7737813535257927
AdaBoostClassifier: 0.779460482725982

XGBoost: 0.7808802650260294

GradientBoostingClassifier: 0.7884524372929484