Telcom Customer Churn Data analysis project

* **Importin libraries**

*import pandas as pd*

*import matplotlib.pyplot as plt*

*import seaborn as sns*

*import numpy as np*

* **Reading CSV file**

*df=pd.read\_csv(r'D:\Python projects\Customer churn project\Customer Churn.csv')*

* **First we will inspect the dataset**

*print(df.describe())*

*print(df.shape)*

*print(df.head())*

*print(df.info())*

* **Total Charges column type is Object but it should be numeric . so after checking csv data in excel we found that there are some blank values because of the tenure of customer is 0 .**
* **We will simply replace Blank space (' ') with 0 and change the data type, so that data type become numeric .**

*df['TotalCharges'] = df['TotalCharges'].replace(' ','0')*

*df['TotalCharges']=df['TotalCharges'].astype('float')*

*print(df.info())*

* **Now we'll check null values , ofcourse by looking at data info we know that there are no null values . But still we will check for summary purpose.**

*print(df.isnull().sum().sum())*

*print(df.duplicated().sum())*

* **we need to check duplicate values on basis of the unique value's column also.**

*print(df['customerID'].duplicated().sum())*

* **The 'SeniorCitizen' column data type is integer because it has value as 0 and 1 , 0 for False and 1 for True .**
* **We will also convert it in to String .**

*def conve(value):*

*if value == 1:*

*return 'YES'*

*else:*

*return 'NO'*

*df['SeniorCitizen']=df['SeniorCitizen'].apply(conve)*

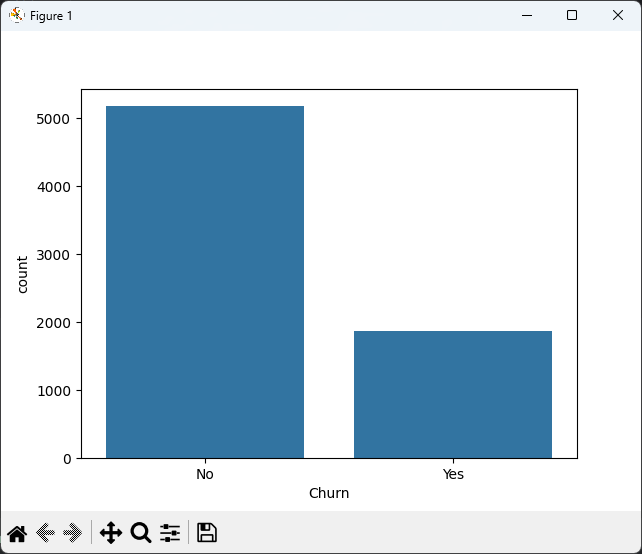
*print(df['SeniorCitizen'])*

*print(df.info())*

* **To check count of the customers who has chruned we use sns.countplot.**

*sns.countplot(x=df['Churn'])*

*plt.show()*

**Plot :-**

* **to show the count number on bar**

*ax = sns.countplot(x='Churn',data= df)*

*for container in ax.containers:*

*ax.bar\_label(container)*

*plt.title(label='Count of Customer by Churn')*

*plt.show()*

**Plot:-**

A screenshot of a computer screen

AI-generated content may be incorrect.

* **To find the percantage of Churn by customer using pie chart**

*plt.figure(figsize=(3,4))* # to change figure properties

*gb = df.groupby('Churn').agg({'Churn':'count'})* # to get % values of yes and no we aregrouping by column 'Churn' and aggregating it to count

*#gb = df['Churn'].value\_counts()*

*print(gb)*

*plt.pie(gb['Churn'], labels=gb.index, autopct='%1.2f%%')* # To get pie chart with lableand % written

*plt.title(label='Percantage of Churned Customer',fontsize=10)*

*plt.show()*

**Plot:-**

A pie chart with numbers and a blue circle

AI-generated content may be incorrect.

* **From the given Pie Chart we can conclude the 26.54% of Customer have churned out.**
* **We will explore the reason behind it .**
* **Lets explore data on basis of Gender.**

*plt.figure(figsize=(4,5))*

*sns.countplot(x='gender',data=df , hue='Churn')*

*plt.show()*

**Plot:-**

A graph with numbers and a chart

AI-generated content may be incorrect.

* **Lets explore data on basis of SeniorCitizen**.

*plt.figure(figsize=(4,5))*

*aa=sns.countplot(x='SeniorCitizen',data=df , hue='Churn')*

*for container in aa.containers:*

*aa.bar\_label(container)*

*plt.show()*

**Plot:-**

A graph with numbers and a bar

AI-generated content may be incorrect.

* **Pie chart by Senior Citizen**

*sc=df.groupby('SeniorCitizen').agg({'SeniorCitizen':'count'})*

*print(sc)*

*plt.pie(sc['SeniorCitizen'],labels=sc.index, autopct='%1.2f%%')*

*plt.title(label='Senior Citizen Pie chart')*

*plt.show()*

**Plot:-**

A screenshot of a computer screen

AI-generated content may be incorrect.

* **Look data by Tenure**

*plt.figure(figsize=(10,5))*

*sns.histplot(x = df['tenure'],data=df,bins=72 , hue='Churn' )*

*for container in ax.containers:*

*ax.bar\_label(container)*

*plt.show()*

**Plot:-**

A screenshot of a computer screen

AI-generated content may be incorrect.

* **Looking at data by Contract**

*plt.figure(figsize=(4,6))*

*ab=sns.countplot(x= 'Contract', data= df , hue='Churn')*

*for container in ab.containers:*

*ab.bar\_label(container)*

*plt.title(label='Count of customer by Contract type')*

*plt.show()*

*aba=df.groupby('Contract').agg({'Contract':'count'})*

*print(aba)*

*plt.pie(aba['Contract'], labels=aba.index , autopct='%1.2f%%')*

*plt.show()*

**Plot:-**

A screenshot of a computer screen

AI-generated content may be incorrect.

* **1. Total customers by contract**

*total\_countbyc = df['Contract'].value\_counts()*

* **2. Only churned customers by contract**

*churned\_customerbyc =df[df['Churn']=='Yes']['Contract'].value\_counts()*

* **3. Plot two pie charts side by side**

*fig, axes = plt.subplots(1,2, figsize=(10,5))*

* **Pie 1: Total customer distribution**

*axes[0].pie(total\_countbyc, labels=total\_countbyc.index, autopct='%1.3f%%')*

*axes[0].set\_title('Total customer by Contract')*

* **Pie 2: Churned customer distribution**

*axes[1].pie(churned\_customerbyc, labels=churned\_customerbyc.index, autopct='%1.3f%%')*

*axes[1].set\_title('Total Churned Customer by contract')*

*plt.tight\_layout()*

*plt.show()*

**Plot:-**

A screenshot of a pie chart

AI-generated content may be incorrect.

* **people who have month to month contract are likely to churn then from those who have 1 or 2 years or contract.**

*print(df.columns.values)*

'''#############\*\*\*\*\*\*\*\*\*\*\*\*\*\*####################'''

*columns = ['PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity',*

*'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies']*

* **Number of columns for the subplot grid**

*n\_cols = 3*

*n\_rows = (len(columns) + n\_cols - 1) // n\_cols # Calculate number of rows needed*

* **Create subplots**

*fig, axes = plt.subplots(n\_rows, n\_cols, figsize=(15, n\_rows \* 4)) # Adjust figsize as needed*

* **Flatten the axes array for easy iteration (handles both 1D and 2D arrays)**

*axes = axes.flatten()*

* **Iterate over columns and plot count plots**

*for i, col in enumerate(columns):*

*sns.countplot(x=col, data=df, ax=axes[i], hue = df["Churn"])*

*axes[i].set\_title(f'Count Plot of {col}')*

*axes[i].set\_xlabel(col)*

*axes[i].set\_ylabel('Count')*

* **Remove empty subplots (if any)**

*for j in range(i + 1, len(axes)):*

*fig.delaxes(axes[j])*

*plt.tight\_layout()*

*plt.show()*

**Plot:-**

A screenshot of a graph

AI-generated content may be incorrect.

* **check data by PaymentMethod**

*plt.figure(figsize=(5,5))*

*ax=sns.countplot(x='PaymentMethod', data=df, hue='Churn')*

*ax.bar\_label(ax.containers[0])*

*ax.bar\_label(ax.containers[1])*

*plt.title('Churned Customer by Payment Method')*

*plt.xticks(rotation=60)*

*plt.show()*

**Plot:-**

A screenshot of a graph

AI-generated content may be incorrect.