**Customer Churn Prediction Project Documentation**

### Step 1: Importing Necessary Libraries

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

!pip install numba

import plotly.offline as po

import plotly.graph\_objs as go

%matplotlib inline

conda install -c conda-forge numba pandas-profiling

import pandas\_profiling

**Explanation:**

* The code begins by importing essential Python libraries for data manipulation, analysis, and visualization, including NumPy, Pandas, Matplotlib, Seaborn, Plotly, and Pandas Profiling.

### **Step 2: Loading the Dataset**

file\_path = "C:\\Users\\surus\\Downloads\\archive (10)\\WA\_Fn-UseC\_-Telco-Customer-Churn.csv"

df = pd.read\_csv(file\_path)

churn\_dataset = df

**Explanation:**

* The dataset is loaded from the specified file path using Pandas' **read\_csv’** function.
* The DataFrame ‘**churn\_dataset’** is created to hold the dataset for further processing.

### **Step 3: Data Cleaning and Preprocessing**

churn\_dataset.loc[churn\_dataset.Churn=='No','Churn']=0

churn\_dataset.loc[churn\_dataset.Churn=='yes','Churn']=1

**Explanation:**

* The 'Churn' column values are converted from 'Yes' and 'No' to binary values, 1 and 0, respectively.

cols = ['OnlineBackup', 'StreamingMovies', 'DeviceProtection', 'TechSupport', 'OnlineSecurity', 'StreamingTV']

for i in cols:

churn\_dataset[i] = churn\_dataset[i].replace({'No internet service': 'No'})

**Explanation:**

* Certain columns with 'No internet service' are replaced with 'No' in the specified list of columns.

churn\_dataset['TotalCharges'] = churn\_dataset["TotalCharges"].replace(" ", np.nan)

churn\_dataset = churn\_dataset[churn\_dataset["TotalCharges"].notnull()]

churn\_dataset = churn\_dataset.reset\_index()[churn\_dataset.columns]

churn\_dataset["TotalCharges"] = churn\_dataset["TotalCharges"].astype(float)

**Explanation:**

* Missing values in the 'TotalCharges' column are replaced with ‘**NaN’**, and rows with missing values are dropped.
* The 'TotalCharges' column is then converted to a float type.

### **Step 4: Exploratory Data Analysis (EDA) with Plotly**

#### **Overall Churn Distribution - Pie Chart**

plot\_by\_churn\_labels = churn\_dataset["Churn"].value\_counts().keys().tolist()

plot\_by\_churn\_values = churn\_dataset["Churn"].value\_counts().values.tolist()

plot\_data = [go.Pie(labels=plot\_by\_churn\_labels,

values=plot\_by\_churn\_values,

marker=dict(colors=['Teal', 'Grey'],

line=dict(color="white",

width=1.5)),

rotation=90,

hoverinfo="label+value+text",

hole= .6)]

plot\_layout = go.Layout(dict(title="Customer Churn",

plot\_bgcolor="rgb(243,243,243)",

paper\_bgcolor="rgb(243,243,243)"))

fig = go.Figure(data=plot\_data, layout=plot\_layout)

po.plot(fig)

**Explanation:**

* A pie chart is created to visualize the overall distribution of churn.
* Plotly is used for interactive plotting, allowing users to hover over segments for additional information.

#### **Churn Rate by Gender - Bar Chart**

plot\_by\_gender = churn\_dataset.groupby('gender').Churn.mean().reset\_index()

plot\_data = [

go.Bar(

x=plot\_by\_gender['gender'],

y=plot\_by\_gender['Churn'],

width=[0.3, 0.3],

marker=dict(color=['orange', 'green'])

)

]

plot\_layout = go.Layout(

xaxis={"type": "category"},

yaxis={"title": "Churn Rate"},

title='Churn Rate by Gender',

plot\_bgcolor='rgb(243,243,243)',

paper\_bgcolor='rgb(243,243,243)',

)

fig = go.Figure(data=plot\_data, layout=plot\_layout)

po.iplot(fig)

**Explanation:**

* A bar chart is created to depict the churn rate based on gender.
* Different colors represent different genders for better visualization.

#### **Churn Rate by Contract Duration - Bar Chart**

plot\_by\_contract = churn\_dataset.groupby('Contract').Churn.mean().reset\_index()

plot\_data = [

go.Bar(

x=plot\_by\_contract['Contract'],

y=plot\_by\_contract['Churn'],

width=[0.3, 0.3, 0.3],

marker=dict(

color=['orange', 'green', 'teal'])

)

]

plot\_layout = go.Layout(

xaxis={"type": "category"},

yaxis={"title": "Churn Rate"},

title='Churn Rate by Contract Duration',

plot\_bgcolor='rgb(243,243,243)',

paper\_bgcolor='rgb(243,243,243)',

)

fig = go.Figure(data=plot\_data, layout=plot\_layout)

po.iplot(fig)

**Explanation:**

* A bar chart is created to illustrate the churn rate based on different contract durations.
* Distinct colors are used for better visual separation.

#### **Relationship Between Tenure & Churn Rate - Scatter Plot**

plot\_by\_tenure = churn\_dataset.groupby('tenure').Churn.mean().reset\_index()

plot\_data = [

go.Scatter(

x=plot\_by\_tenure['tenure'],

y=plot\_by\_tenure['Churn'],

mode='markers',

name='Low',

marker=dict(size=5,

line=dict(width=0.8),

color='green')

)

]

plot\_layout = go.Layout(

yaxis={"title": "Churn Rate"},

xaxis={"title": "Tenure"},

title='Churn Rate by Tenure & Churn Rate',

plot\_bgcolor='rgb(243,243,243)',

paper\_bgcolor='rgb(243,243,243)',

)

fig = go.Figure(data=plot\_data, layout=plot\_layout)

po.iplot(fig)

**Explanation:**

* A scatter plot is created to visualize the relationship between tenure and churn rate.
* Different marker sizes and colors are used to emphasize specific data points.

### Overall Comments:

* The code integrates data cleaning, preprocessing, and exploratory data analysis using Plotly for interactive visualizations.
* Visualizations include a pie chart for overall churn distribution and bar charts for churn rates based on gender and contract duration.
* The scatter plot explores the relationship between tenure and churn rate.
* Each visualization is designed to provide insights into different aspects of customer churn.

**Note:**

* Ensure that the file path is correct, and all required libraries are installed for the code to run successfully.
* Adjustments may be needed based on specific data and analysis requirements.