NAAN MUDHALVAN PROJECT

PHASE 3 : Development Part 1

CUSTOMER CHURN PREDICTION

Team Members :

ANANYA K A – 2021115013

ANULATHA S K - 2021115014

ARADHYA M - 2021115015

SAKTHIVEL K – 2021115088

BALAJI J - 2021115323

Development part :

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
df = pd.read_csv('customer_churn.csv')
df.head(5)
```

Importing Libraries:

- import numpy as np: Imports the NumPy library, which is used for numerical and mathematical operations in Python. It's commonly used for working with arrays and matrices.
- import pandas as pd: Imports the Pandas library, which is widely used for data manipulation and analysis. It provides data structures like DataFrames and Series, making it ideal for working with structured data.
- import matplotlib.pyplot as plt: Imports the Matplotlib library, a popular library for data visualization in Python.

Loading the Dataset:

- Reads a CSV file named 'customer_churn.csv' and stores the data in a Pandas DataFrame named df. This line loads the dataset for analysis. The .read_csv function is used to read data from a CSV file and create a DataFrame.

Displaying the First 5 Rows of the Dataset:

- The .head(5) method is used to display the first 5 rows of the dataset stored in the DataFrame df. This provides an initial glimpse of the data to understand its structure and content. The number 5 in .head(5) specifies how many rows to display

```
df.drop('customerID',axis='columns',inplace=True)
df.sample(5)
```

Dropping a Column:

- This code is removing a specific column from the DataFrame df. The column being removed is specified by its label, in this case, 'customerID'. The drop method is used for this purpose. The method can be used to drop rows or columns from the DataFrame.
- axis='columns': The axis parameter is set to 'columns' to indicate that we want to drop a column. inplace=True: The inplace parameter is set to True, which means that the change is made directly to the DataFrame df, and it doesn't return a new DataFrame. If inplace is set to False (the default), a new DataFrame with the column dropped would be returned, but in this case, it's modifying the existing DataFrame.

Displaying a Random Sample:

- After dropping the 'customerID' column, this code displays a random sample of 5 rows from the DataFrame df. The .sample(5) method is used for this purpose.

OUTPUT:

```
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
```

```
df_tenure_no = df1[df1.Churn == 'No'].tenure
df_tenure_yes = df1[df1.Churn == 'Yes'].tenure
plt.hist([df_tenure_yes,df_tenure_no],color=['green','red'],label=['Churn=Yes','Churn=No'])
```

```
plt.legend()
plt.xlabel('Tenure')
plt.ylabel('Number of customers')
plt.title('Histogram based on Tenure of customers')
```

Filtering Data:

- The code starts by filtering the data into two separate Pandas Series based on the value of the 'Churn' column:
  - df_tenure_no: Contains the 'tenure' values for customers who have not churned ('Churn=No').
  - df_tenure_yes: Contains the 'tenure' values for customers who have churned ('Churn=Yes').

Creating a Histogram:

- The plt.hist function is used to create a histogram. It takes two lists (in this case, Pandas Series) as input for the data to be plotted. In this case, it's creating a histogram for two groups: customers who have churned ('Churn=Yes') and customers who have not churned ('Churn=No').

Color and Labels:

- The color parameter is used to specify the colors for the histogram bars. In this code, it assigns 'green' to the bars for customers who have churned ('Churn=Yes') and 'red' to the bars for customers who have not churned ('Churn=No').
- The label parameter is used to provide labels for the legend. It labels the two sets of data as 'Churn=Yes' and 'Churn=No' to distinguish them in the legend.

Legend:

- The plt.legend() function is used to display a legend that helps identify the groups. In this case, it will show 'Churn=Yes' and 'Churn=No' with the corresponding colors.

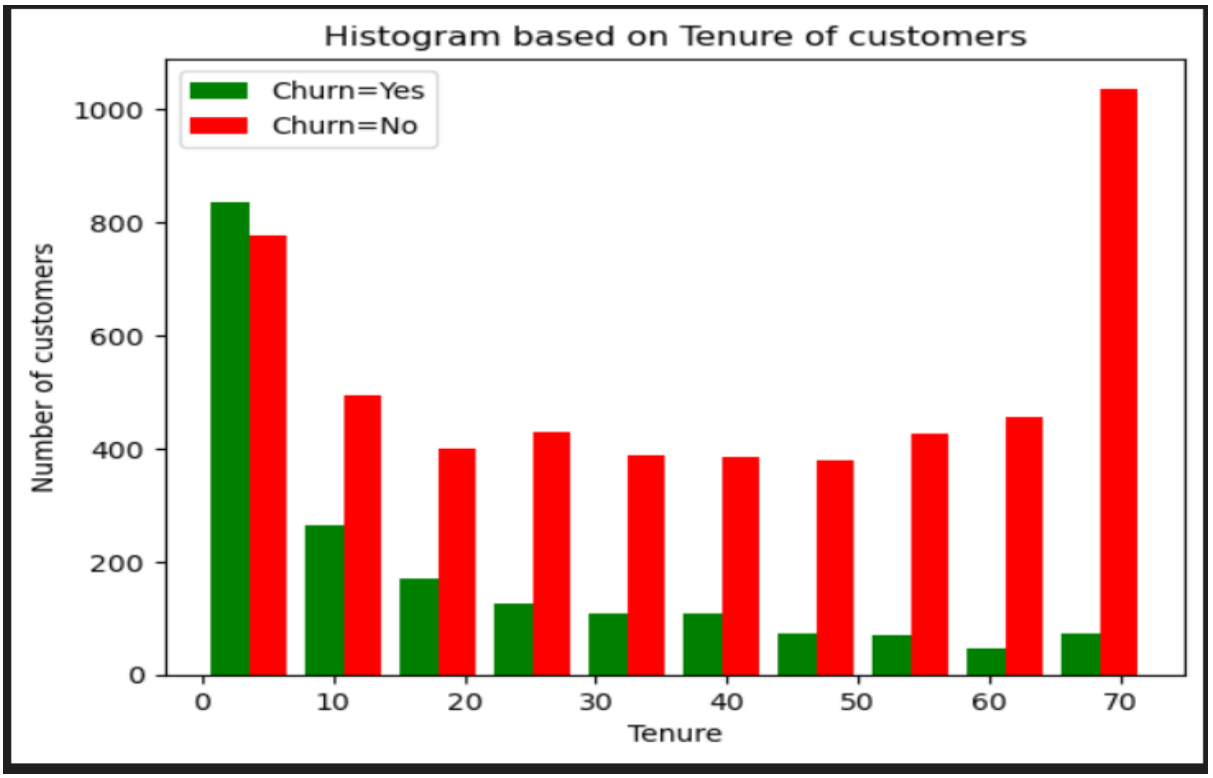
X and Y Labels:

- plt.xlabel('Tenure') sets the label for the x-axis, indicating that the histogram represents the 'Tenure' of customers.
- plt.ylabel('Number of customers') sets the label for the y-axis, indicating that the values on the y-axis represent the number of customers.

Title:

- plt.title('Histogram based on Tenure of customers') sets the title of the histogram to provide context to the viewer.
-

The result is a histogram that visually represents the distribution of customer tenures for two groups.



```
df1.replace('No phone service','No',inplace=True)
df1.replace('No internet service','No',inplace=True)
print_unique_values(df1)
```

Replacing Values:

- df1.replace('No phone service', 'No', inplace=True): This line of code is replacing all occurrences of 'No phone service' in the DataFrame df1 with 'No'. The inplace=True parameter means that the changes are made directly to the DataFrame, and it doesn't return a new DataFrame.
- df1.replace('No internet service', 'No', inplace=True): This line is similarly replacing all occurrences of 'No internet service' with 'No'.

These replacements are often done to standardize the data by replacing different but equivalent values with a common representation ('No' in this case).

Calling a Function:

- print_unique_values(df1): The code is calling a function named print_unique_values(df1) to print the unique values in the DataFrame df1. However, the function print_unique_values() is not defined in the provided code, so the code as it stands would result in an error.

```
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' 'Yes']
InternetService ['DSL' 'Fiber optic' 'No']
OnlineSecurity ['No' 'Yes']
OnlineBackup ['Yes' 'No']
DeviceProtection ['No' 'Yes']
TechSupport ['No' 'Yes']
StreamingTV ['No' 'Yes']
StreamingMovies ['No' 'Yes']
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']
```

Similarly

```
df_yes_no_cols =
['Partner','Dependents','PhoneService','MultipleLines','OnlineSecurity'
,'OnlineBackup','DeviceProtection','TechSupport','StreamingTV','Streami
ngMovies','PaperlessBilling','Churn']
for cols in df_yes_no_cols:
    df2[cols].replace({'Yes':1,'No':0},inplace=True)
print_unique_values(df2)
for cols in df_true_false_cols:
    df2[cols].replace({True:1,False:0},inplace=True)
```

```
gender ['Female' 'Male']
SeniorCitizen [0 1]
Partner [1 0]
Dependents [0 1]
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 [0 1]
MultipleLines [0 1]
OnlineSecurity [0 1]
OnlineBackup [1 0]
DeviceProtection [0 1]
TechSupport [0 1]
StreamingTV [0 1]
StreamingMovies [0 1]
PaperlessBilling [1 0]
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 [0 1]
InternetService_DSL [1 0]
InternetService_Fiber optic [0 1]
InternetService_No [0 1]
Contract_Month-to-month [1 0]
Contract_One year [0 1]
...
PaymentMethod_Bank transfer (automatic) [0 1]
PaymentMethod_Credit card (automatic) [0 1]
PaymentMethod_Electronic check [1 0]
PaymentMethod_Mailed check [0 1]
```

```
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' 'Yes']
OnlineSecurity ['No' 'Yes']
OnlineBackup ['Yes' 'No']
DeviceProtection ['No' 'Yes']
TechSupport ['No' 'Yes']
StreamingTV ['No' 'Yes']
StreamingMovies ['No' 'Yes']
PaperlessBilling ['Yes' 'No']
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']
InternetService_DSL [1 0]
InternetService_Fiber optic [0 1]
InternetService_No [0 1]
Contract_Month-to-month [1 0]
Contract_One year [0 1]
...
PaymentMethod_Bank transfer (automatic) [0 1]
PaymentMethod_Credit card (automatic) [0 1]
PaymentMethod_Electronic check [1 0]
PaymentMethod_Mailed check [0 1]
```

```
df2=pd.get_dummies(data=df1,columns=['InternetService','Contract','Paym
entMethod'])
Df2.shape
print_unique_values(df2)
```

```
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' 'Yes']
OnlineSecurity ['No' 'Yes']
OnlineBackup ['Yes' 'No']
DeviceProtection ['No' 'Yes']
TechSupport ['No' 'Yes']
StreamingTV ['No' 'Yes']
StreamingMovies ['No' 'Yes']
PaperlessBilling ['Yes' 'No']
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']
InternetService_DSL [1 0]
InternetService_Fiber optic [0 1]
InternetService_No [0 1]
Contract_Month-to-month [1 0]
Contract_One year [0 1]
...
PaymentMethod_Bank transfer (automatic) [0 1]
PaymentMethod_Credit card (automatic) [0 1]
PaymentMethod_Electronic check [1 0]
PaymentMethod_Mailed check [0 1]
```