PROJECT TITLE: CUSTOMER CHURN PREDICTION

PHASE 3: DATA PREPROCESSING

IN THIS PHASE WE ARE GOING TO LOAD AND PREPROCESS THE DATA.

OBJECTIVE:

The objective of this phase is to load the given dataset and identify the pattern and factors that contribute to customer churn, enabling proactive measures to retain customers.

STEPS:

STEP 1:DATA COLLECTION AND LOAD THE DATA

Collecting the data from the source shared and loading the dataset.

Using the library function to load the dataset. For ex. pandas in python

STEP 2: HANDLING MISSING VALUES

After loading the dataset, we have to identify the missing values and to handle the missing values.

STEP 3: FEATURE ENGINEERING

Create new features from the existing dataset that could potentially provide better insights for churn prediction. This could involve creating variables like customer tenure, usage frequency, or any other relevant metrics.

STEP 4: DATA TRANSFORMATION

Normalize or standardize the data if necessary to bring all features to a similar scale, enabling a fair comparison between different features.

STEP 5: DATA SPLITTING

Split the dataset into training and testing sets to evaluate the model's performance accurately. Ensure that the split is random and maintains the distribution of the target variable.

STEP 6: DATA VALIDATION

Perform a thorough validation check to ensure that the dataset is error-free and has been processed accurately.

By following these steps we ensure that the dataset is cleaned and preprocessed.

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px
pd.set_option('display.max_columns', None)
```

In [3]:
```
df = pd.read_csv("Telco-Customer-Churn.csv")
```

In [4]:
```
df.head()
```

Out[4]:

nes	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	StreamingTV	StreamingMovies	Contract	PaperlessBilling	PaymentMethod	M
one vice	DSL	No	Yes	No	No	No	No	Month-to-month	Yes	Electronic check	
No	DSL	Yes	No	Yes	No	No	No	One year	No	Mailed check	
No	DSL	Yes	Yes	No	No	No	No	Month-to-month	Yes	Mailed check	
one vice	DSL	Yes	No	Yes	Yes	No	No	One year	No	Bank transfer (automatic)	
No	Fiber optic	No	No	No	No	No	No	Month-to-month	Yes	Electronic check	

In [5]:
```
df.shape
```

Out[5]:
```
(7043, 21)
```

In [6]:
```
df.columns
```

Out[6]:
```
Index(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',
       'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport',
       'StreamingTV', 'StreamingMovies', 'Contract', 'PaperlessBilling',
       'PaymentMethod', 'MonthlyCharges', 'TotalCharges', 'Churn'],
      dtype='object')
```

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   customerID        7043 non-null   object 
 1   gender            7043 non-null   object 
 2   SeniorCitizen     7043 non-null   int64  
 3   Partner           7043 non-null   object 
 4   Dependents        7043 non-null   object 
 5   tenure            7043 non-null   int64  
 6   PhoneService      7043 non-null   object 
 7   MultipleLines     7043 non-null   object 
 8   InternetService   7043 non-null   object 
 9   OnlineSecurity    7043 non-null   object 
 10  OnlineBackup      7043 non-null   object 
 11  DeviceProtection  7043 non-null   object 
 12  TechSupport       7043 non-null   object 
 13  StreamingTV       7043 non-null   object 
 14  StreamingMovies   7043 non-null   object 
 15  Contract          7043 non-null   object 
 16  PaperlessBilling  7043 non-null   object 
 17  PaymentMethod     7043 non-null   object 
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   object 
 20  Churn             7043 non-null   object 
dtypes: float64(1), int64(2), object(18)
```

In [8]:
```
df.nunique()
```

Out[8]:
```
customerID          7043
gender                 2
SeniorCitizen          2
Partner                2
Dependents             2
tenure                73
PhoneService           2
MultipleLines          3
InternetService        3
OnlineSecurity         3
OnlineBackup           3
DeviceProtection       3
TechSupport            3
StreamingTV            3
StreamingMovies        3
Contract               3
PaperlessBilling       2
PaymentMethod          4
MonthlyCharges      1585
TotalCharges        6531
Churn                  2
dtype: int64
```

In [71]:
```
def missing_values(n):
    df_m=pd.DataFrame()
    df_m["missing_values, %"]=df.isnull().sum()*100/len(df.isnull())
    df_m["missing_values, sum"]=df.isnull().sum()
    return df_m.sort_values(by="missing_values, %", ascending=False)
missing_values(df)
```

	missing_values, %	missing_values, sum
gender	0.0	0
SeniorCitizen	0.0	0
TotalCharges	0.0	0
MonthlyCharges	0.0	0
PaymentMethod	0.0	0
PaperlessBilling	0.0	0
Contract	0.0	0
StreamingMovies	0.0	0
StreamingTV	0.0	0
TechSupport	0.0	0
DeviceProtection	0.0	0
OnlineBackup	0.0	0
OnlineSecurity	0.0	0
InternetService	0.0	0
MultipleLines	0.0	0
PhoneService	0.0	0
tenure	0.0	0
Dependents	0.0	0
Partner	0.0	0
Churn	0.0	0

In [9]:
```
df.dtypes
```

Out[9]:
```
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 [10]:
```
for col in df.columns:
    print(f"{col} : {df[col].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 [11]:
```
df.describe()
```

Out[11]:

	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

min total charge == 0 and min monthly charges == 18 that is impossible at least min are the same in both

In [12]:
```
Services = ['PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity' , 'OnlineBackup',
            'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies']
account_information = ['Contract', 'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges', 'TotalCharges', 'tenure']

Demographic = ['gender', 'SeniorCitizen', 'Partner', 'Dependents']
```

In [13]:
```
df["TotalCharges"] = df["TotalCharges"].replace(" " , "0")
df["TotalCharges"] = df["TotalCharges"].astype(float)
```

In [14]:
```
df.drop(columns=['customerID'],inplace=True)
```

In [15]:
```
df.dtypes
```

Out[15]:
```
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       float64
Churn               object
dtype: object
```

In [16]:
```
df["Churn"] = df["Churn"].replace({'Yes' : '1' , "No" : '0'}).astype("int")
```

In [17]:
```
df.head()
```

Out[17]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSuppo
0	Female	0	Yes	No	1	No	No phone service	DSL	No	Yes	No	N
1	Male	0	No	No	34	Yes	No	DSL	Yes	No	Yes	N
2	Male	0	No	No	2	Yes	No	DSL	Yes	Yes	No	N
3	Male	0	No	No	45	No	No phone service	DSL	Yes	No	Yes	Y
4	Female	0	No	No	2	Yes	No	Fiber optic	No	No	No	N

In [18]:
```
df[df["TotalCharges"] == 0]
```

Out[18]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSu
488	Female	0	Yes	Yes	0	No	No phone service	DSL	Yes	No	Yes	
753	Male	0	No	Yes	0	Yes	No	No	No internet service	No internet service	No internet service	No in s
936	Female	0	Yes	Yes	0	Yes	No	DSL	Yes	Yes	Yes	
1082	Male	0	Yes	Yes	0	Yes	Yes	No	No internet service	No internet service	No internet service	No in s
1340	Female	0	Yes	Yes	0	No	No phone service	DSL	Yes	Yes	Yes	
3331	Male	0	Yes	Yes	0	Yes	No	No	No internet service	No internet service	No internet service	No in s
3826	Male	0	Yes	Yes	0	Yes	Yes	No	No internet service	No internet service	No internet service	No in s
4380	Female	0	Yes	Yes	0	Yes	No	No	No internet service	No internet service	No internet service	No in s
5218	Male	0	Yes	Yes	0	Yes	No	No	No internet service	No internet service	No internet service	No in s
6670	Female	0	Yes	Yes	0	Yes	Yes	DSL	No	Yes	Yes	
6754	Male	0	No	Yes	0	Yes	Yes	DSL	Yes	Yes	No	

In [19]:
```
df.loc[df['TotalCharges'] == 0, 'TotalCharges'] = df['MonthlyCharges']
```

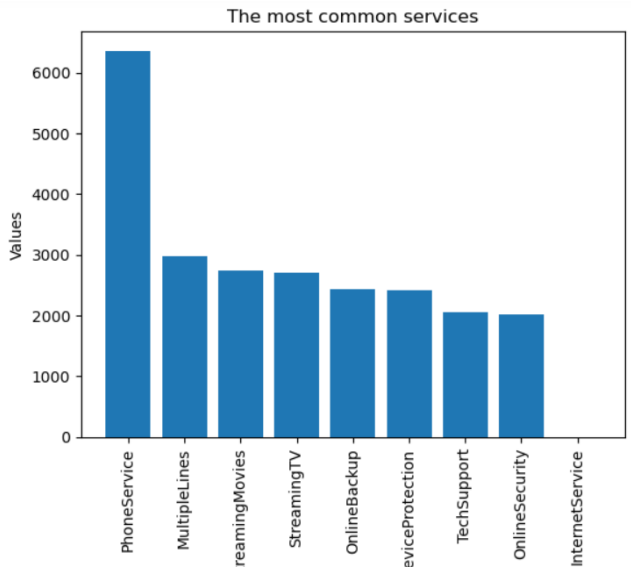
In [20]:
```
for col in Services:
    df[col] = df[col].replace({'No phone service' : 'No' , 'No internet service' : 'No'})
```

In [21]:
```
df.describe()
```

Out[21]:

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges	Churn
count	7043.000000	7043.000000	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692	2279.798992	0.265370
std	0.368612	24.559481	30.090047	2266.730170	0.441561
min	0.000000	0.000000	18.250000	18.800000	0.000000
25%	0.000000	9.000000	35.500000	398.550000	0.000000
50%	0.000000	29.000000	70.350000	1394.550000	0.000000
75%	0.000000	55.000000	89.850000	3786.600000	1.000000
max	1.000000	72.000000	118.750000	8684.800000	1.000000

In [25]:
```
data = pd.DataFrame({'Categories': name, 'Values': corr})
df_sorted = data.sort_values(by='Values', ascending=False)
plt.bar(df_sorted['Categories'], df_sorted['Values'])
plt.xlabel('Categories')
plt.ylabel('Values')
plt.title('The most common services')
plt.xticks(rotation='vertical');
```



In [27]:
```
Services
```

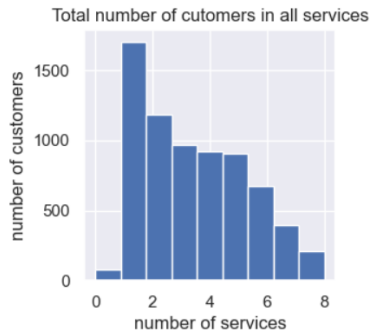
Out[27]:
```
['PhoneService',
 'MultipleLines',
 'InternetService',
 'OnlineSecurity',
 'OnlineBackup',
 'DeviceProtection',
 'TechSupport',
 'StreamingTV',
 'StreamingMovies']
```

In [28]:
```
all_services = [0] * 7043
df['allservices'] = all_services
df.head()
```

Out[28]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSuppo
0	Female	0	Yes	No	1	No	No	DSL	No	Yes	No	N
1	Male	0	No	No	34	Yes	No	DSL	Yes	No	Yes	N
2	Male	0	No	No	2	Yes	No	DSL	Yes	Yes	No	N
3	Male	0	No	No	45	No	No	DSL	Yes	No	Yes	Y
4	Female	0	No	No	2	Yes	No	Fiber optic	No	No	No	N

In [30]:
```
plt.hist(df['allservices'] , 9)
plt.xlabel("number of services")
plt.ylabel("number of customers")
plt.title('Total number of cutomers in all services');
```



In [35]:
```
df.describe()
```

Out[35]:

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges	Churn	allservices
count	7043.000000	7043.000000	7043.000000	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692	2279.798992	0.265370	3.362914
std	0.368612	24.559481	30.090047	2266.730170	0.441561	2.062031
min	0.000000	0.000000	18.250000	18.800000	0.000000	0.000000
25%	0.000000	9.000000	35.500000	398.550000	0.000000	1.000000
50%	0.000000	29.000000	70.350000	1394.550000	0.000000	3.000000
75%	0.000000	55.000000	89.850000	3786.600000	1.000000	5.000000
max	1.000000	72.000000	118.750000	8684.800000	1.000000	8.000000

In [36]:
```
df[df['allservices'] == 0]
```

Out[36]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSu
105	Male	0	No	No	5	No	No	DSL	No	No	No	
185	Female	0	Yes	No	1	No	No	DSL	No	No	No	
211	Female	0	No	No	1	No	No	DSL	No	No	No	
272	Male	0	No	No	1	No	No	DSL	No	No	No	
376	Male	0	No	No	1	No	No	DSL	No	No	No	
...	...	...	...	...	...	...	...	...	...	...	...	
6536	Male	0	No	No	1	No	No	DSL	No	No	No	
6607	Male	0	No	Yes	1	No	No	DSL	No	No	No	
6864	Female	1	No	No	3	No	No	DSL	No	No	No	
6979	Male	0	No	Yes	1	No	No	DSL	No	No	No	
6984	Male	0	No	Yes	31	No	No	DSL	No	No	No	

80 rows × 21 columns

In [37]:
```
df[df['TotalCharges'] < 100]
```

Out[37]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSu
0	Female	0	Yes	No	1	No	No	DSL	No	Yes	No	
20	Male	1	No	No	1	No	No	DSL	No	No	Yes	
22	Male	0	No	No	1	Yes	No	No	No	No	No	
27	Male	0	Yes	Yes	1	No	No	DSL	No	Yes	No	
33	Male	0	No	No	1	Yes	No	No	No	No	No	
...	...	...	...	...	...	...	...	...	...	...	...	
7010	Female	1	Yes	No	1	Yes	Yes	Fiber optic	No	No	No	
7016	Female	0	No	No	1	Yes	No	DSL	No	Yes	No	
7018	Male	0	Yes	Yes	1	Yes	No	Fiber optic	No	No	No	
7030	Female	0	No	No	2	Yes	No	No	No	No	No	
7032	Male	1	No	No	1	Yes	Yes	Fiber optic	No	No	No	

In [38]:
```
plt.hist(df['tenure'] , 10);
```