Task_1_Data_Preparation

November 21, 2025

1 1. Loading the Dataset

1. Data From GitHub
2. Display First 5 Rows
3. Shape of the Dataset

[142]:
```
import pandas as pd

# GitHub raw URL
url = 'https://raw.githubusercontent.com/abuthahir17/Dataset/main/
 ↪Telco_Customer_Churn_Dataset.csv'

# Read CSV file
data = pd.read_csv(url)

print("Dataset Loaded Successfully!")
```

```
Dataset Loaded Successfully!
```

[143]:
```
# Display First 5 rows
print("First Five Rows in the Dataset: \n")
data.head()
```

```
First Five Rows in the Dataset:
```

[143]:

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

	MultipleLines	InternetService	OnlineSecurity	…	DeviceProtection	\
0	No phone service	DSL	No	…	No	
1	No	DSL	Yes	…	Yes	
2	No	DSL	Yes	…	No	
3	No phone service	DSL	Yes	…	Yes	
4	No	Fiber optic	No	…	No	

```
  TechSupport StreamingTV StreamingMovies        Contract PaperlessBilling  \
0          No          No              No  Month-to-month              Yes
1          No          No              No        One year               No
2          No          No              No  Month-to-month              Yes
3         Yes          No              No        One year               No
4          No          No              No  Month-to-month              Yes

               PaymentMethod MonthlyCharges  TotalCharges Churn
0           Electronic check          29.85         29.85    No
1               Mailed check          56.95        1889.5    No
2               Mailed check          53.85        108.15   Yes
3  Bank transfer (automatic)          42.30       1840.75    No
4           Electronic check          70.70        151.65   Yes

[5 rows x 21 columns]
```

[144]:
```
print("Shape of dataset:", data.shape)
```

```
Shape of dataset: (7043, 21)
```

2 2. Initial Data Exploration

1. Dataset Information
2. Statistical Summary of the Dataset
3. Data type of the Dataset
4. Number of Missing Values (Count)

[126]:
```
print("\nDataset Info:\n")
print(data.info())
```

```
Dataset 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)
memory usage: 1.1+ MB
None
```

[127]:
```
print("\nStatistical Summary:\n")
print(data.describe(include='all'))
```

```
Statistical Summary:

        customerID gender  SeniorCitizen Partner Dependents       tenure  \
count         7043   7043    7043.000000    7043       7043  7043.000000
unique        7043      2            NaN       2          2          NaN
top     3186-AJIEK   Male            NaN      No         No          NaN
freq             1   3555            NaN    3641       4933          NaN
mean           NaN    NaN       0.162147     NaN        NaN    32.371149
std            NaN    NaN       0.368612     NaN        NaN    24.559481
min            NaN    NaN       0.000000     NaN        NaN     0.000000
25%            NaN    NaN       0.000000     NaN        NaN     9.000000
50%            NaN    NaN       0.000000     NaN        NaN    29.000000
75%            NaN    NaN       0.000000     NaN        NaN    55.000000
max            NaN    NaN       1.000000     NaN        NaN    72.000000

       PhoneService MultipleLines InternetService OnlineSecurity  …  \
count          7043          7043            7043           7043  …
unique            2             3               3              3  …
top             Yes            No     Fiber optic             No  …
freq           6361          3390            3096           3498  …
mean            NaN           NaN             NaN            NaN  …
std             NaN           NaN             NaN            NaN  …
min             NaN           NaN             NaN            NaN  …
25%             NaN           NaN             NaN            NaN  …
50%             NaN           NaN             NaN            NaN  …
75%             NaN           NaN             NaN            NaN  …
max             NaN           NaN             NaN            NaN  …
```

```
       DeviceProtection TechSupport StreamingTV StreamingMovies  \
count              7043        7043        7043            7043
unique                3           3           3               3
top                  No          No          No              No
freq               3095        3473        2810            2785
mean                NaN         NaN         NaN             NaN
std                 NaN         NaN         NaN             NaN
min                 NaN         NaN         NaN             NaN
25%                 NaN         NaN         NaN             NaN
50%                 NaN         NaN         NaN             NaN
75%                 NaN         NaN         NaN             NaN
max                 NaN         NaN         NaN             NaN

              Contract PaperlessBilling     PaymentMethod MonthlyCharges  \
count             7043             7043              7043    7043.000000
unique               3                2                 4            NaN
top     Month-to-month              Yes  Electronic check            NaN
freq              3875             4171              2365            NaN
mean               NaN              NaN               NaN      64.761692
std                NaN              NaN               NaN      30.090047
min                NaN              NaN               NaN      18.250000
25%                NaN              NaN               NaN      35.500000
50%                NaN              NaN               NaN      70.350000
75%                NaN              NaN               NaN      89.850000
max                NaN              NaN               NaN     118.750000

       TotalCharges Churn
count          7043  7043
unique         6531     2
top                   No
freq             11  5174
mean            NaN   NaN
std             NaN   NaN
min             NaN   NaN
25%             NaN   NaN
50%             NaN   NaN
75%             NaN   NaN
max             NaN   NaN

[11 rows x 21 columns]
```

[128]:
```
# Print Column data types
print("Column Data Types:\n", data.dtypes, "\n")
```

```
Column Data Types:
 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
```

[129]:
```
print("\nNumber of the missing values in each column:\n")
print(data.isnull().sum())
```

```
Number of the missing values in each column:

customerID          0
gender              0
SeniorCitizen       0
Partner             0
Dependents          0
tenure              0
PhoneService        0
MultipleLines       0
InternetService     0
OnlineSecurity      0
OnlineBackup        0
DeviceProtection    0
TechSupport         0
StreamingTV         0
StreamingMovies     0
Contract            0
PaperlessBilling    0
PaymentMethod       0
MonthlyCharges      0
TotalCharges        0
Churn               0
```

```
dtype: int64
```

3 3. Handling Missing Values

1. Check also " " value (Null)
2. TotalCharges -> Numeric (Convert)
3. Fill with Median
4. Recheck Null values

[130]:
```
# Also Check the " " value

data.replace(" ", None, inplace=True)
print("Missing Value: \n", data.isnull().sum())
```

```
Missing Value:
 customerID          0
gender               0
SeniorCitizen        0
Partner              0
Dependents           0
tenure               0
PhoneService         0
MultipleLines        0
InternetService      0
OnlineSecurity       0
OnlineBackup         0
DeviceProtection     0
TechSupport          0
StreamingTV          0
StreamingMovies      0
Contract             0
PaperlessBilling     0
PaymentMethod        0
MonthlyCharges       0
TotalCharges        11
Churn                0
dtype: int64
```

[131]:
```
# Convert TotalCharges to Numeric
data["TotalCharges"] = pd.to_numeric(data["TotalCharges"], errors="coerce")
```

[132]:
```
# Fill the missing value with median
data["TotalCharges"] = data["TotalCharges"].fillna(data["TotalCharges"].
 ↪median())
```

[133]:
```
# Also Check the " " value
data.replace(" ", None, inplace=True)
print("Missing Value after Cleaning: \n", data.isnull().sum())
```

```
Missing Value after Cleaning:
 customerID          0
gender               0
SeniorCitizen        0
Partner              0
Dependents           0
tenure               0
PhoneService         0
MultipleLines        0
InternetService      0
OnlineSecurity       0
OnlineBackup         0
DeviceProtection     0
TechSupport          0
StreamingTV          0
StreamingMovies      0
Contract             0
PaperlessBilling     0
PaymentMethod        0
MonthlyCharges       0
TotalCharges         0
Churn                0
dtype: int64
```

4 4. Encoding Categorical Variables

1. Identify Numerical and Categorical Columns
2. Before Encoding
3. During Encoding
4. After Encoding

[134]:
```
# Identify numerical columns
numerical_cols = data.select_dtypes(include=['int64', 'float64']).columns
print("Total number of Numerical Columns:", len(numerical_cols))
print("All Numerical Columns:" ,list(numerical_cols), "\n")

# Identify categorical columns
categorical_cols = data.select_dtypes(include=['object']).columns
print("Total number of Categorical Columns:", len(categorical_cols))
print("All Categorical Columns:" , list(categorical_cols))
```

```
Total number of Numerical Columns: 4
All Numerical Columns: ['SeniorCitizen', 'tenure', 'MonthlyCharges',
'TotalCharges']

Total number of Categorical Columns: 17
All Categorical Columns: ['customerID', 'gender', 'Partner', 'Dependents',
'PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity',
```

```
'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV',
'StreamingMovies', 'Contract', 'PaperlessBilling', 'PaymentMethod', 'Churn'] /n
```

[135]:
```
print("Before encoding:\n", data.head())

print("Shape before encoding:", data.shape)
```

```
Before encoding:
    customerID  gender  SeniorCitizen Partner Dependents  tenure PhoneService  \
0  7590-VHVEG  Female              0     Yes         No       1           No
1  5575-GNVDE    Male              0      No         No      34          Yes
2  3668-QPYBK    Male              0      No         No       2          Yes
3  7795-CFOCW    Male              0      No         No      45           No
4  9237-HQITU  Female              0      No         No       2          Yes

      MultipleLines InternetService OnlineSecurity  … DeviceProtection  \
0  No phone service             DSL             No  …               No
1                No             DSL            Yes  …              Yes
2                No             DSL            Yes  …               No
3  No phone service             DSL            Yes  …              Yes
4                No     Fiber optic             No  …               No

  TechSupport StreamingTV StreamingMovies        Contract PaperlessBilling  \
0          No          No              No  Month-to-month              Yes
1          No          No              No        One year               No
2          No          No              No  Month-to-month              Yes
3         Yes          No              No        One year               No
4          No          No              No  Month-to-month              Yes

               PaymentMethod MonthlyCharges  TotalCharges  Churn
0           Electronic check          29.85         29.85     No
1               Mailed check          56.95       1889.50     No
2               Mailed check          53.85        108.15    Yes
3  Bank transfer (automatic)          42.30       1840.75     No
4           Electronic check          70.70        151.65    Yes

[5 rows x 21 columns]
Shape before encoding: (7043, 21)
```

[136]:
```
from sklearn.preprocessing import LabelEncoder

# Binary columns (2 unique values)
binary_cols = [col for col in categorical_cols if data[col].nunique() == 2]
print("Binary Columns (LabelEncode):", binary_cols, "\n")

# Multi-category columns (>2 unique values)
multi_cat_cols = [col for col in categorical_cols if data[col].nunique() > 2␣
 ↪and col not in binary_cols]
```

```
print("Multi-category Columns (One-hot Encode):", multi_cat_cols, "\n")

label_encoder = LabelEncoder()

# Binary
for col in binary_cols:
    data[col] = label_encoder.fit_transform(data[col])

# Multi-Category (One-Hot Encoding)
data = pd.get_dummies(data, columns=multi_cat_cols, drop_first=True)

print("Categorical Encoding Completed.")
```

```
Binary Columns (LabelEncode): ['gender', 'Partner', 'Dependents',
'PhoneService', 'PaperlessBilling', 'Churn']

Multi-category Columns (One-hot Encode): ['customerID', 'MultipleLines',
'InternetService', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract', 'PaymentMethod']

Categorical Encoding Completed.
```

[137]:
```
print("After encoding:\n", data.head())
print("New Shape After Encoding:", data.shape)
```

```
After encoding:
    gender  SeniorCitizen  Partner  Dependents  tenure  PhoneService  \
0       0              0        1           0       1             0
1       1              0        0           0      34             1
2       1              0        0           0       2             1
3       1              0        0           0      45             0
4       0              0        0           0       2             1

   PaperlessBilling  MonthlyCharges  TotalCharges  Churn  …  \
0                 1           29.85         29.85      0  …
1                 0           56.95       1889.50      0  …
2                 1           53.85        108.15      1  …
3                 0           42.30       1840.75      0  …
4                 1           70.70        151.65      1  …

   TechSupport_Yes  StreamingTV_No internet service  StreamingTV_Yes  \
0            False                            False            False
1            False                            False            False
2            False                            False            False
3             True                            False            False
4            False                            False            False

   StreamingMovies_No internet service  StreamingMovies_Yes  \
```

```
0                                False                False
1                                False                False
2                                False                False
3                                False                False
4                                False                False

   Contract_One year  Contract_Two year  \
0              False              False
1               True              False
2              False              False
3               True              False
4              False              False

   PaymentMethod_Credit card (automatic)  PaymentMethod_Electronic check  \
0                                  False                            True
1                                  False                           False
2                                  False                           False
3                                  False                           False
4                                  False                            True

   PaymentMethod_Mailed check
0                       False
1                        True
2                        True
3                       False
4                       False

[5 rows x 7073 columns]
New Shape After Encoding: (7043, 7073)
```

5 5. Dataset Splitting (Train/Test)

1. Fix the target variable
2. Split the Dataset

[140]:
```python
from sklearn.model_selection import train_test_split

# Target variable
y = data["Churn"]
X = data.drop("Churn", axis=1)
```

[141]:
```python
# Split the dataset into 2 set (80% train, 20% test)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

print("Training set shape:", X_train.shape)
```

```
print("Testing set shape:", X_test.shape)
```

```
Training set shape: (5634, 7072)
Testing set shape: (1409, 7072)
```