DATA ANALYTICS WITH COGNOS- GROUP3

CUSTOMER CHURN PREDICTION

PHASE3 | DEVELOPMENT PART 1

DATA PREPROCESSING:

STEP 1: Loading Dataset

We import the necessary libraries and read the dataset using pandas and then we use the head() function to view the sample of how the dataset is.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
data= pd.read_csv("/content/churn_dataset.csv")
data.head()
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No

5 rows × 21 columns

STEP2: Checking for null values

We try to find the columns equipping null values as it might be a drawback while performing operations on the dataset.

```
[ ]  data.isnull().sum()

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
```

It has been analyzed that the dataset doesn’t hold any null values by checking the sum of null values in each column using the isnull().sum() function.

STEP3: Insights of the dataset

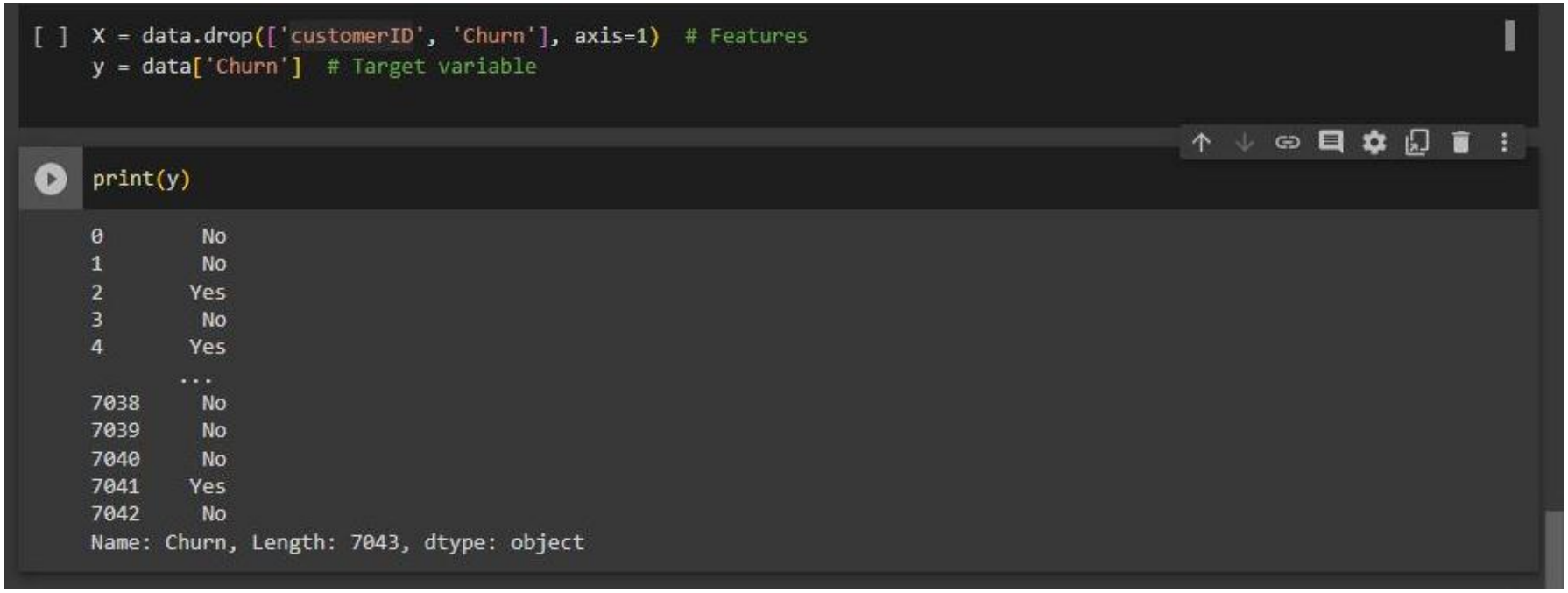
We derive the information about the dataset using the info() function, the count of the churners and non-churners and also about the relativity of churners and the payment method which is one of the attribute using the inbuilt functions.

```
[ ]  data.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
```

```
[ ]  data['Churn'].value_counts()

     No     5174
     Yes    1869
     Name: Churn, dtype: int64


[ ]  data.columns

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


[ ]  print(data.groupby('Churn')['PaymentMethod'].value_counts())

     Churn  PaymentMethod
     No     Mailed check                1304
            Electronic check            1294
            Credit card (automatic)     1290
            Bank transfer (automatic)   1286
     Yes    Electronic check            1071
            Mailed check                 308
            Bank transfer (automatic)    258
            Credit card (automatic)      232
     Name: PaymentMethod, dtype: int64
```

STEP4: Splitting feature and target variable

For splitting feature variable(x), we drop the customerID(as it does not contribute to the prediction) and churn(as it is the target variable) and for target variable(y), we store the churn column.



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

```
print(y)
```

```
0        No
1        No
2       Yes
3        No
4       Yes
       ...
7038     No
7039     No
7040     No
7041    Yes
7042     No
Name: Churn, Length: 7043, dtype: object
```

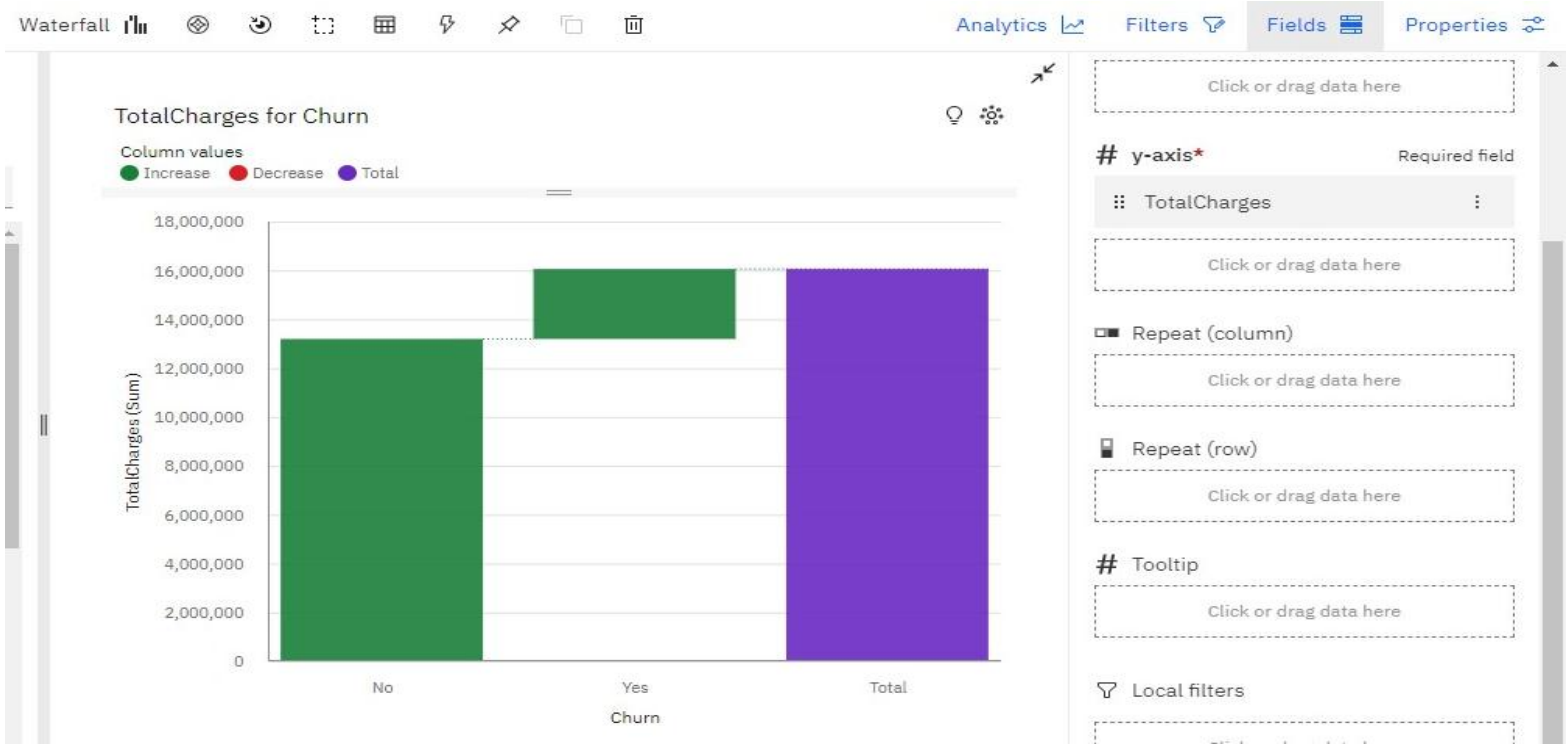
We print the y which is the target variable to make sure it has been split in the appropriate manner as it is the important aspect that the model has to learn for the prediction.

DATA VISUALIZATION:

We use **Cognos** to visualize the data in the efficient way by uploading our dataset and performing distinct visualization forms which will be efficient for our analytics.

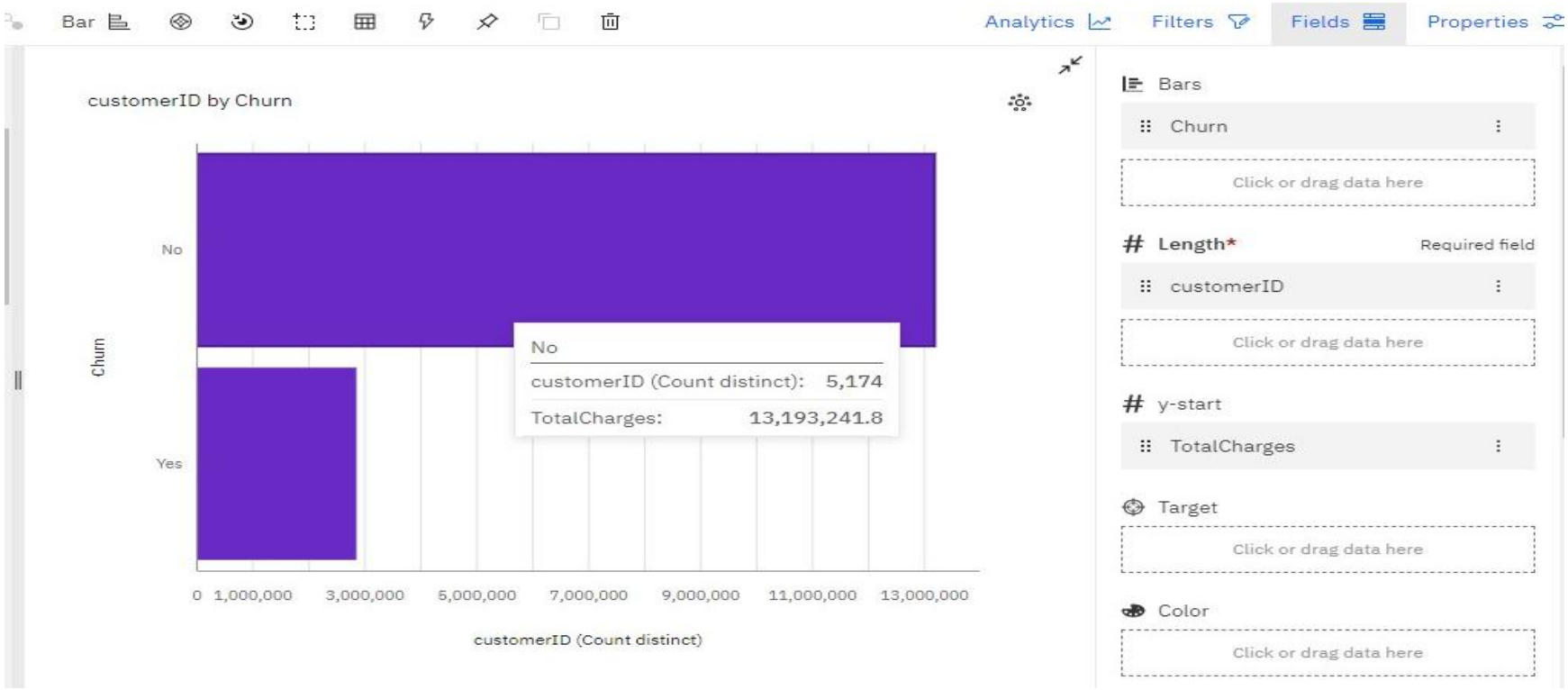
1) WATERFALL

We use the waterfall visualization method to represent the relation between the total charges of churners and non-churners and also the sum of them and their relativity.



2) BAR PLOT

We use bar plot to represent the Total charges and CustomerID distinct count with respect to churn.



Here it is observed that the distinct count of Non-churners is 5,174 and the sum of their total charges is around 13,193,241.

3) LINE AND COLUMN CHART

Using this line and column chart we represent the line by senior citizen sum total and the columns by the payment method with respect to churn and this chart exhibits their relationship with one another.

CONCLUSION:

In this phase, The Preprocessing and Visualization of Data had been implemented using python and cognos.

In the state of Data Preprocessing we have made sure the dataset doesn’t contain any null values, derived some valuable insights about the data by analyzing them and had split the dataset as features and target variable.

In the state of Data visualization we have used cognos to visualize specific features from the dataset with respect to the target variable.