

# Phase 3

## Customer Churn Prediction

To load and preprocess a Customer Churn dataset for analysis, you can follow these general steps using Python and Pandas. Make sure you have a Customer Churn dataset in a suitable format available.

1. **Import Libraries:** Start by importing the necessary Python libraries, including Pandas, to load and preprocess the dataset.

### > Import pandas as pd

2. **Load the Customer Churn Dataset:** Load the COVID-19 dataset into a Pandas DataFrame. You can use `pd.read_csv()` for CSV files, but the method may vary depending on the file format.

```
In [ ]: import pandas as pd
```

```
In [9]: import pandas as pd
df = pd.read_csv("K:\Downloads\customer-churn.csv")
df
```

Out[9]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechS
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	
...	...	...	...	...	...	...	...	...	...	...	...	...	...
7038	6840-RESVB	Male	0	Yes	Yes	24	Yes	Yes	DSL	Yes	...	Yes	
7039	2234-XADUH	Female	0	Yes	Yes	72	Yes	Yes	Fiber optic	No	...	Yes	
7040	4801-JZAZL	Female	0	Yes	Yes	11	No	No phone service	DSL	Yes	...	No	
7041	8361-LTMKD	Male	1	Yes	No	4	Yes	Yes	Fiber optic	No	...	No	
7042	3186-AJIEK	Male	0	No	No	66	Yes	No	Fiber optic	Yes	...	Yes	

7043 rows × 14 columns

3. **Data Inspection:** Before preprocessing, inspect the data to understand its structure and identify any potential issues.

```
In [10]: print(df.head())
```

	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]

```
In [11]: print(df.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
```

```
In [13]: print(df.dtypes)
```

```
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
```

## 4. Data Preprocessing:

### a. Data Cleaning:

Handle missing values by either imputing them or removing rows with missing data

```
In [14]: df.fillna(0,inplace=True)
df.dropna(inplace=True)
df
```

Out[14]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechS
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	
...	...	...	...	...	...	...	...	...	...	...	...	...	...
7038	6840-RESVB	Male	0	Yes	Yes	24	Yes	Yes	DSL	Yes	...	Yes	
7039	2234-XADUH	Female	0	Yes	Yes	72	Yes	Yes	Fiber optic	No	...	Yes	
7040	4801-JAZL	Female	0	Yes	Yes	11	No	No phone service	DSL	Yes	...	No	
7041	8361-LTMKD	Male	1	Yes	No	4	Yes	Yes	Fiber optic	No	...	No	
7042	3186-AJIEK	Male	0	No	No	66	Yes	No	Fiber optic	Yes	...	Yes	

7043 rows x 21 columns

**b. Data Transformation:** If necessary, transform the data to suit your analysis objectives. For instance, you may want to aggregate data by date or region.

```
In [17]: df=df.groupby('customerID').agg({'MonthlyCharges':'sum','TotalCharges':'sum'}).reset_index()
df
```

Out[17]:

	customerID	MonthlyCharges	TotalCharges
0	0002-ORFBO	65.60	593.3
1	0003-MKNFE	59.90	542.4
2	0004-TLHLJ	73.90	260.85
3	0011-IGKFF	98.00	1237.85
4	0013-EXCHZ	83.90	267.4
...	...	...	...
7038	9987-LUTYD	55.15	742.9
7039	9992-RRAMN	85.10	1873.7
7040	9992-UJOEL	50.30	92.75
7041	9993-LHIEB	67.85	4627.65
7042	9995-HOTOH	59.00	3707.6

7043 rows x 3 columns

**c. Data Filtering:**

- Filter the data to focus on a specific time frame or specific regions of interest.

```
In [22]: start_id = '0013-EXCHZ'
end_id = '9987-LUTYD'
df=df[(df['customerID']>=start_id)&(df['customerID']<=end_id)]
df
```

```
Out[22]:
```

	customerID	MonthlyCharges	TotalCharges
4	0013-EXCHZ	83.90	267.4
5	0013-MHZWF	69.40	571.45
6	0013-SMEOE	109.70	7904.25
7	0014-BMAQU	84.65	5377.8
8	0015-UOCOJ	48.20	340.35
...	...	...	...
7034	9978-HYCIN	84.95	4018.05
7035	9979-RGMZT	94.05	633.45
7036	9985-MWVIX	70.15	70.15
7037	9986-BONCE	20.95	85.5
7038	9987-LUTYD	55.15	742.9

7035 rows x 3 columns

**5. Save the Preprocessed Data (Optional):** If you want to save the preprocessed data for future analysis, you can use Pandas to save it to a new CSV file.

```
In [23]: df.to_csv('preprocessed_customer_churn.csv',index=False)
```

These are the general steps to load and preprocess a Customer Churn dataset using Python and Pandas. Remember that the specific preprocessing steps and operations may vary depending on the structure of your dataset and your analysis objectives