Website Traffic Analysis – Phase 3

TEAM MEMBERS :

KAMARAJ	**2021115047**
KARTHIKA	**2021115049**
KARTHIKEYAN	**2021115050**
KAVIYA	**2021115051**
JEEVESH	**2021115312**

DATA PREPROCESSING AND CLEANING:

In this phase, our primary task is to analyze the given dataset and clean it so that we can use it for further project development. The process of data cleaning includes removing null values. So at the end of the process, we must have all non-null values in all fields.

Data Ingest

```
import pandas as pd

FILE_LOCATION = 'daily-website-visitors.csv'

whole_dataset = pd.read_csv(FILE_LOCATION,
                            index_col='Date',
                            thousands=',')
whole_dataset.index = pd.to_datetime(whole_dataset.index)
whole_dataset
```

Date	Row	Day	Day.Of.Week	Page.Loads	Unique.Visits	First.Time.Visits	Re
2014-09-14	1	Sunday	1	2146	1582	1430	
2014-09-15	2	Monday	2	3621	2528	2297	
2014-09-16	3	Tuesday	3	3698	2630	2352	
2014-09-17	4	Wednesday	4	3667	2614	2327	
2014-09-18	5	Thursday	5	3316	2366	2130	
...	...	...	...	...	...	...	
2020-08-15	2163	Saturday	7	2221	1696	1373	
2020-08-16	2164	Sunday	1	2724	2037	1686	

```
whole_dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 2167 entries, 2014-09-14 to 2020-08-19
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Row                2167 non-null   int64
 1   Day                2167 non-null   object
 2   Day.Of.Week        2167 non-null   int64
 3   Page.Loads         2167 non-null   int64
 4   Unique.Visits      2167 non-null   int64
 5   First.Time.Visits  2167 non-null   int64
 6   Returning.Visits   2167 non-null   int64
dtypes: int64(6), object(1)
memory usage: 135.4+ KB
```

From the above result, we can see that our dataset is already clear and contains all non-null

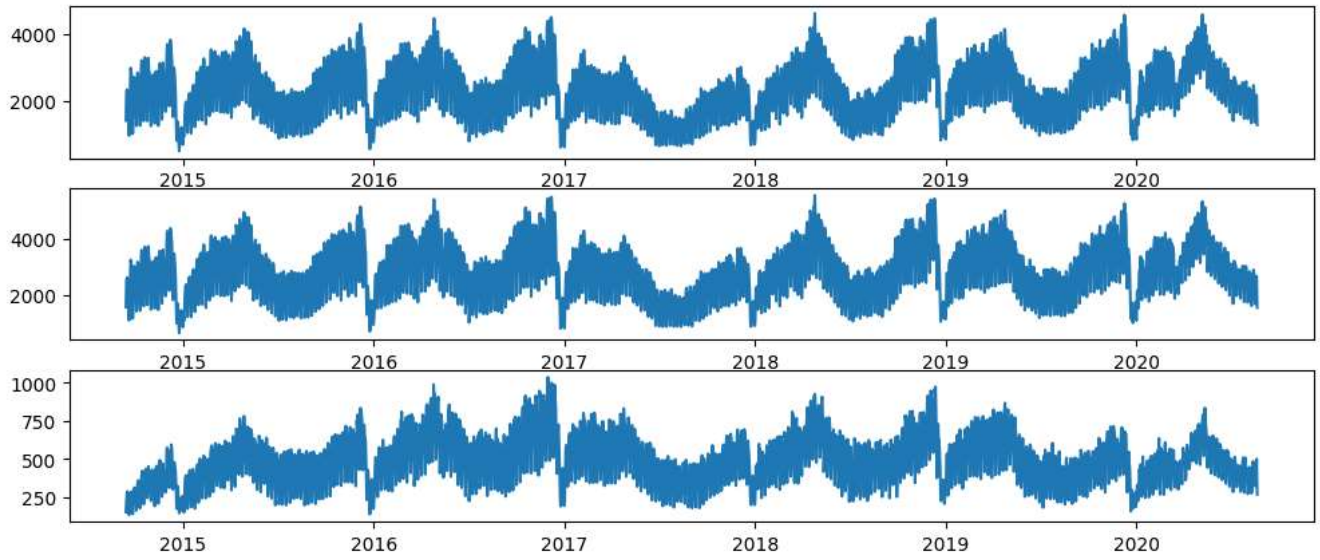
```
whole_dataset.describe()
```

	Row	Day.Of.Week	Page.Loads	Unique.Visits	First.Time.Visits	Returning.Visits
count	2167.000000	2167.000000	2167.000000	2167.000000	2167.000000	2167.000000
mean	1084.000000	3.997231	4116.989386	2943.646516	2431.824181	511.822335
std	625.703338	2.000229	1350.977843	977.886472	828.704688	168.736370
min	1.000000	1.000000	1002.000000	667.000000	522.000000	133.000000
25%	542.500000	2.000000	3114.500000	2226.000000	1830.000000	388.500000
50%	1084.000000	4.000000	4106.000000	2914.000000	2400.000000	509.000000
75%	1625.500000	6.000000	5020.500000	3667.500000	3038.000000	626.500000
max	2167.000000	7.000000	7984.000000	5541.000000	4616.000000	1036.000000

```
import matplotlib.pyplot as plt

fig, axs = plt.subplots(3, figsize=(12, 5))
```

```
axs[0].plot(whole_dataset['First.Time.Visits'])
axs[1].plot(whole_dataset['Unique.Visits'])
axs[2].plot(whole_dataset['Returning.Visits'])
plt.show()
```



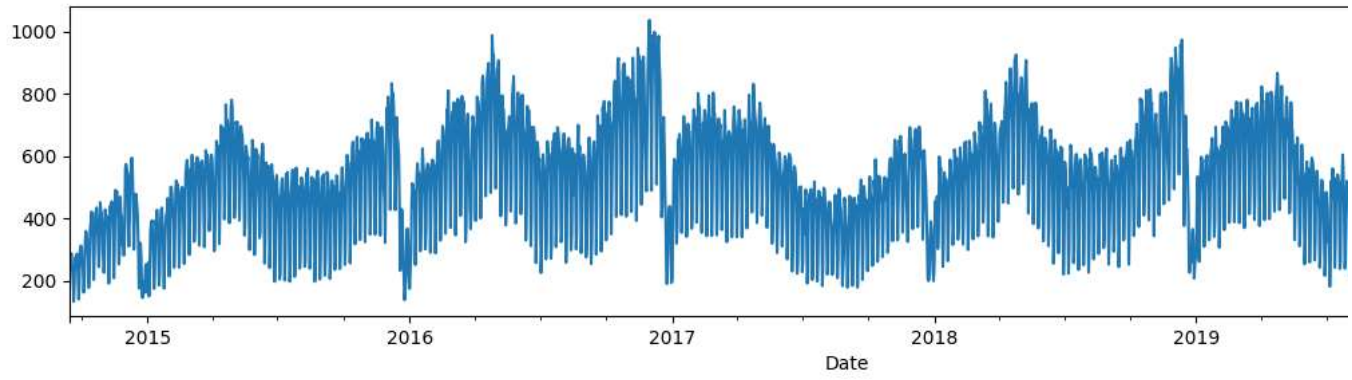
Preprocessing the data

- Target Attribute: **Returning.Visits** We shall predict the **Returning.Visits** given past data.

```
target_column = whole_dataset['Returning.Visits']
target_column
```

```
Date
2014-09-14    152
2014-09-15    231
2014-09-16    278
2014-09-17    287
2014-09-18    236
              ...
2020-08-15    323
2020-08-16    351
2020-08-17    457
2020-08-18    499
2020-08-19    267
Name: Returning.Visits, Length: 2167, dtype: int64
```

```
target_column.plot(figsize=(15, 3))
plt.show()
```



#Data Preprocessing

```
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.tsatools import freq_to_period
from statsmodels.graphics.tsaplots import plot_pacf
from statsmodels.tsa.arima_model import ARIMA
import statsmodels.api as sm

# Corrected file path using double backslashes
data = pd.read_csv("daily-website-visitors.csv")
print(data.head())
```

```
   Row        Day  Day.Of.Week       Date Page.Loads Unique.Visits  \
0    1     Sunday            1  9/14/2014      2,146         1,582
1    2     Monday            2  9/15/2014      3,621         2,528
2    3    Tuesday            3  9/16/2014      3,698         2,630
3    4  Wednesday            4  9/17/2014      3,667         2,614
4    5   Thursday            5  9/18/2014      3,316         2,366

  First.Time.Visits Returning.Visits
0             1,430              152
1             2,297              231
2             2,352              278
3             2,327              287
4             2,130              236
```

```
data['Date'] = pd.to_datetime(data['Date'], format='%m/%d/%Y')

print(data.info())
```

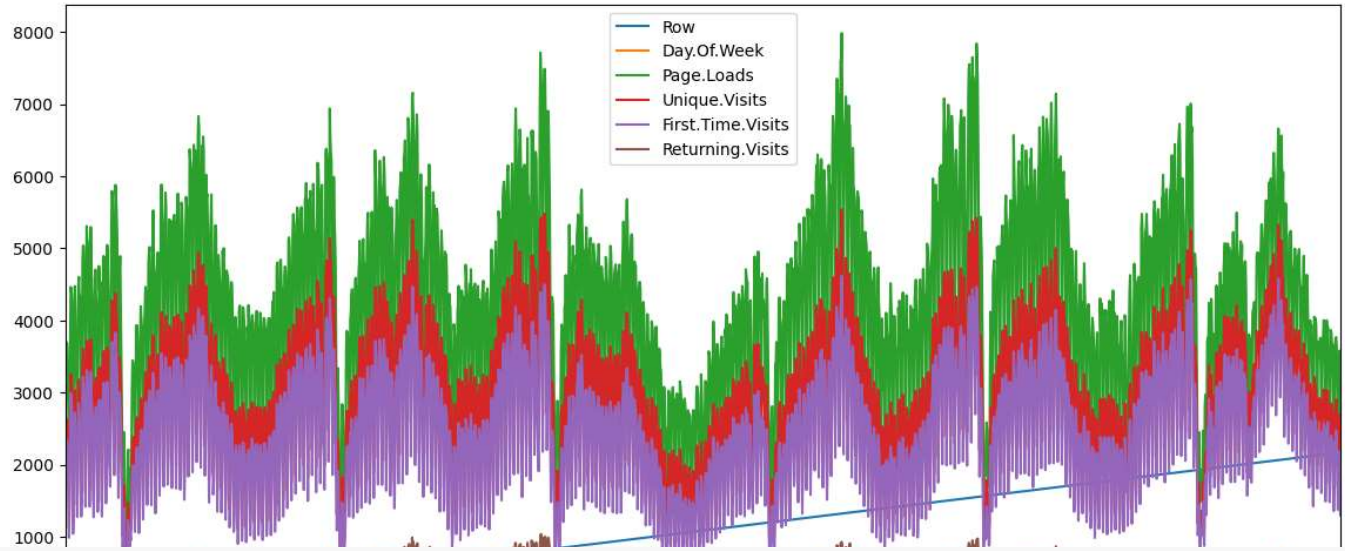
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2167 entries, 0 to 2166
Data columns (total 8 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Row                2167 non-null   int64
 1   Day                2167 non-null   object
 2   Day.Of.Week        2167 non-null   int64
 3   Date               2167 non-null   datetime64[ns]
 4   Page.Loads         2167 non-null   object
 5   Unique.Visits      2167 non-null   object
 6   First.Time.Visits  2167 non-null   object
 7   Returning.Visits   2167 non-null   object
dtypes: datetime64[ns](1), int64(2), object(5)
memory usage: 135.6+ KB
None
```

```
df = pd.read_csv("daily-website-visitors.csv", \
                 index_col = 'Date', thousands = ',', parse_dates=True)
df.head()
```

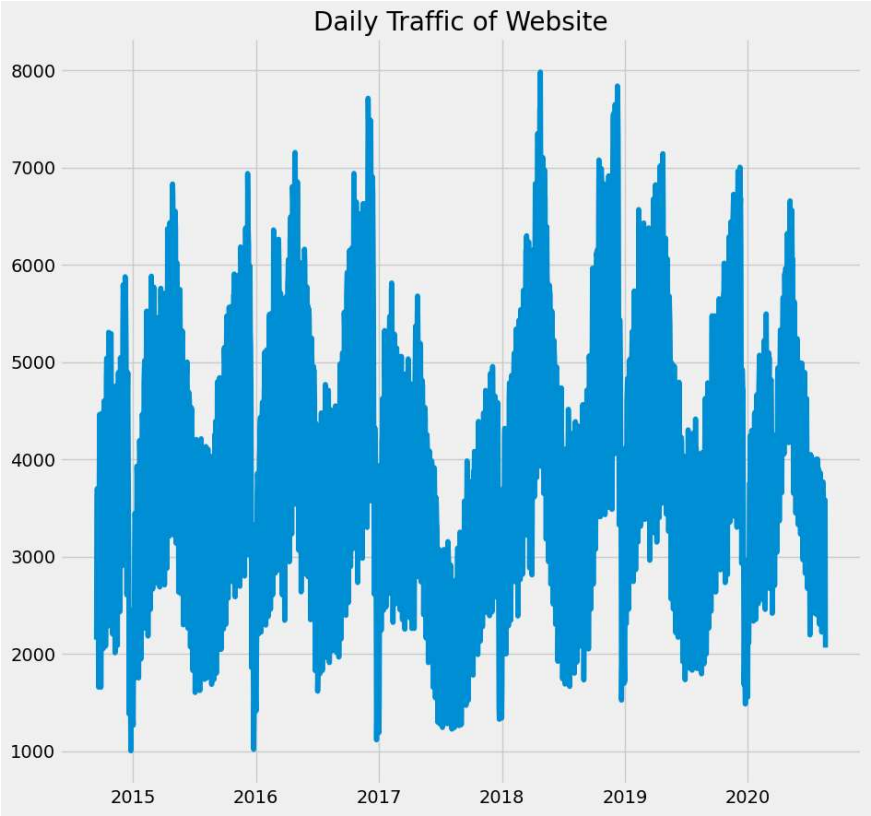
Date	Row	Day	Day.Of.Week	Page.Loads	Unique.Visits	First.Time.Visits	Returning.Visits
2014-09-14	1	Sunday	1	2146	1582	1430	152
2014-09-15	2	Monday	2	3621	2528	2297	231
2014-09-16	3	Tuesday	3	3698	2630	2352	278
2014-09-17	4	Wednesday	4	3667	2614	2327	287
2014-09-18	5	Thursday	5	3316	2366	2130	236

```
df.plot(figsize=(14,7))
```

```
<Axes: xlabel='Date'>
```




```
plt.style.use('fivethirtyeight')
plt.figure(figsize=(10, 10))
plt.plot(data["Date"], data["Page.Loads"])
plt.title("Daily Traffic of Website")
plt.show()
```




```
import matplotlib.pyplot as plt

plt.style.use('fivethirtyeight')
plt.figure(figsize=(15, 10))
```

```
# Convert "First.Time.Visits" to numeric values
data["First.Time.Visits"] = data["First.Time.Visits"].str.replace(',', '').astype(int)

plt.plot(data["Date"], data["First.Time.Visits"])
plt.title("Daily Traffic of Website")

# Adjust the y-axis range to make all values visible
plt.ylim(0, data["First.Time.Visits"].max() + 100)  # Adjust the upper limit as needed

plt.show()
```