

Website Traffic Analysis

Date	25-10-2023
Team ID	1295
Project Name	Website Traffic Analysis

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

```
df =pd.read_csv("/content/daily-website-visitors.csv")
```

```
df.head()
```

	Row	Day	Day.Of.Week	Date	Page.Loads	Unique.Visits	\
0	1	Sunday	1	9/14/2014	NaN	897.384615	
1	2	Monday	2	9/15/2014	NaN	897.384615	
2	3	Tuesday	3	9/16/2014	NaN	897.384615	
3	4	Wednesday	4	9/17/2014	NaN	897.384615	
4	5	Thursday	5	9/18/2014	NaN	897.384615	

	First.Time.Visits	Returning.Visits
0	841.558442	152.0
1	841.558442	231.0
2	841.558442	278.0
3	841.558442	287.0
4	841.558442	236.0

```
df
```

	Row	Day	Day.Of.Week	Date	Page.Loads	Unique.Visits	\
0	1	Sunday	1	9/14/2014	NaN	897.384615	
1	2	Monday	2	9/15/2014	NaN	897.384615	
2	3	Tuesday	3	9/16/2014	NaN	897.384615	
3	4	Wednesday	4	9/17/2014	NaN	897.384615	
4	5	Thursday	5	9/18/2014	NaN	897.384615	
...
2080	2163	Saturday	7	8/15/2020	NaN	897.384615	
2081	2164	Sunday	1	8/16/2020	NaN	897.384615	
2082	2165	Monday	2	8/17/2020	NaN	897.384615	
2083	2166	Tuesday	3	8/18/2020	NaN	897.384615	
2084	2167	Wednesday	4	8/19/2020	NaN	897.384615	

	First.Time.Visits	Returning.Visits
0	841.558442	152.0
1	841.558442	231.0
2	841.558442	278.0
3	841.558442	287.0
4	841.558442	236.0
...
2080	841.558442	323.0
2081	841.558442	351.0
2082	841.558442	457.0
2083	841.558442	499.0
2084	841.558442	267.0

[2085 rows x 8 columns]

```
import pandas as pd
import matplotlib.pyplot as plt
```

```
# Convert the 'Date' column to a datetime object
```

```
df['Date'] = pd.to_datetime(df['Date'])
```

```
# Extract the year from the 'Date' column and create a new 'Year' column
```

```
df['Year'] = df['Date'].dt.year
```

```
# Convert the 'Returning.Visits' column to string and then to numeric values
```

```
df['Returning.Visits'] = df['Returning.Visits'].astype(str).str.replace(',',  
'', regex=True).astype(float)
```

```
# Group the data by year and calculate the sum of 'Returning.Visits'
```

```
yearly_data = df.groupby('Year')['Returning.Visits'].sum()
```

```
# Plot the time series of total returning visits by year
```

```
plt.figure(figsize=(12, 6))
```

```
yearly_data.plot(kind='line', marker='o', linestyle='-')
```

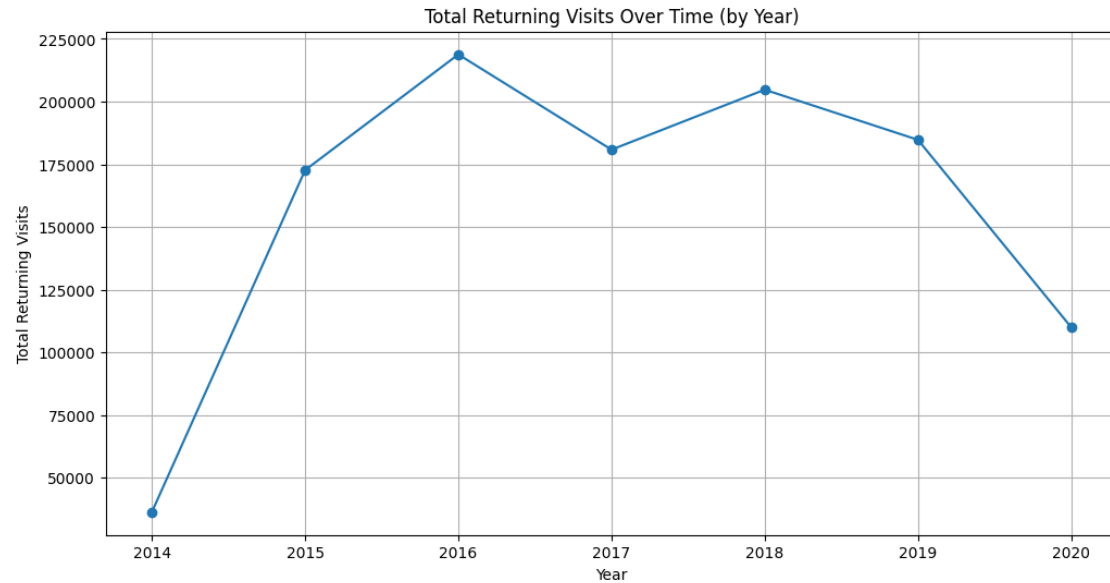
```
plt.xlabel('Year')
```

```
plt.ylabel('Total Returning Visits')
```

```
plt.title('Total Returning Visits Over Time (by Year)')
```

```
plt.grid(True)
```

```
plt.show()
```

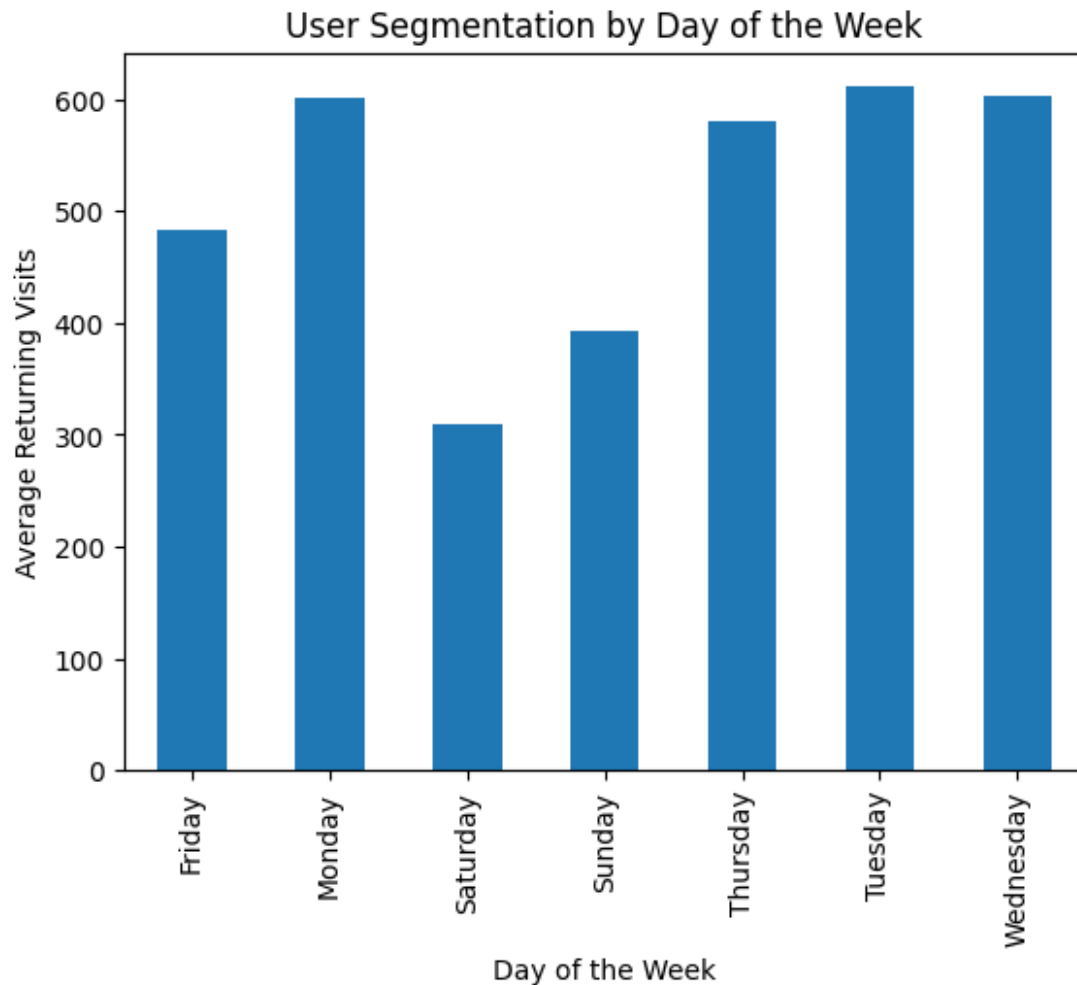


```
# Assuming the 'Returning.Visits' column contains strings of numbers
separated by commas
# Convert the strings to a list of numbers, and then calculate the mean

df['Returning.Visits'] = df['Returning.Visits'].str.split(',').apply(lambda
x: [float(val) for val in x])
df['Returning.Visits'] = df['Returning.Visits'].apply(lambda x: sum(x) /
len(x) if x else 0)

# Group the data by 'Day' and calculate the mean of 'Returning.Visits'
day_of_week_segments = df.groupby('Day')['Returning.Visits'].mean()

# Plot the user segmentation
day_of_week_segments.plot(kind='bar')
plt.xlabel('Day of the Week')
plt.ylabel('Average Returning Visits')
plt.title('User Segmentation by Day of the Week')
plt.show()
```



```
from sklearn.linear_model import LinearRegression

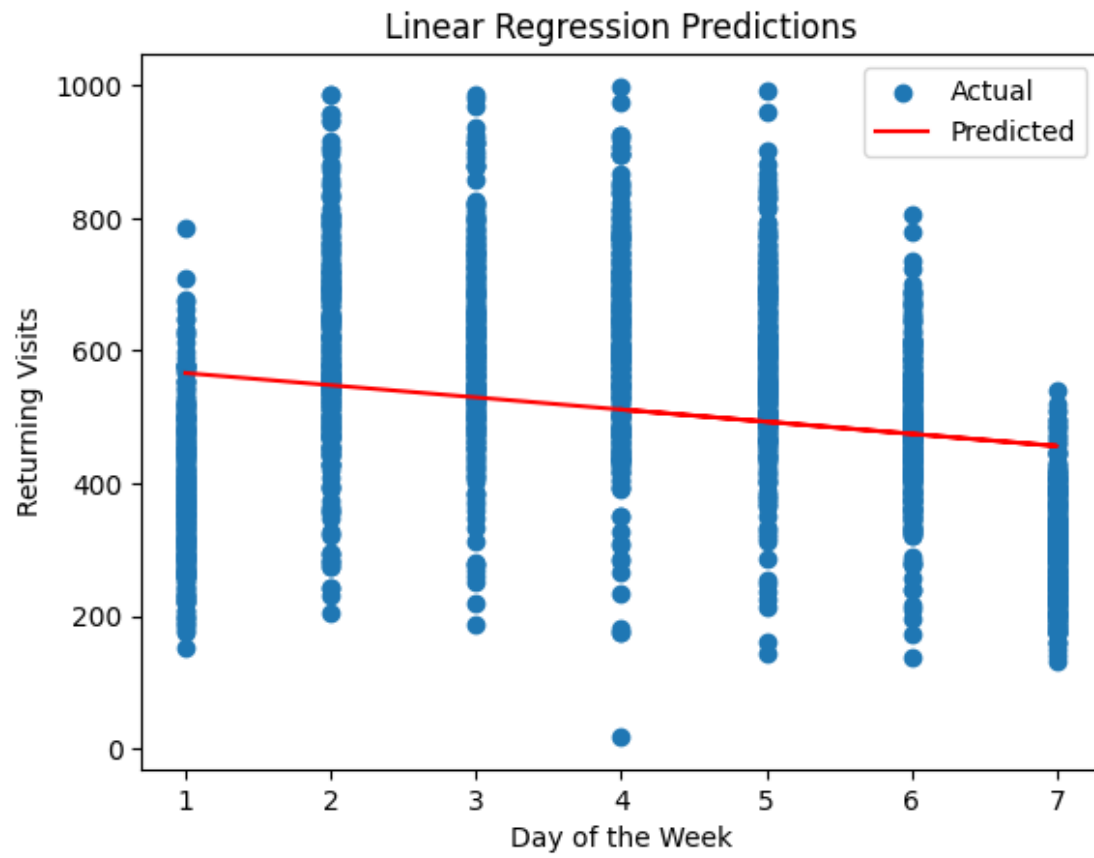
# Prepare the data
X = df[['Day.Of.Week']]
y = df['Returning.Visits']

# Create and train the model
model = LinearRegression()
model.fit(X, y)

# Make predictions
predictions = model.predict(X)

# Visualize the predictions
plt.scatter(df['Day.Of.Week'], y, label='Actual')
plt.plot(df['Day.Of.Week'], predictions, color='red', label='Predicted')
plt.xlabel('Day of the Week')
plt.ylabel('Returning Visits')
plt.legend()
```

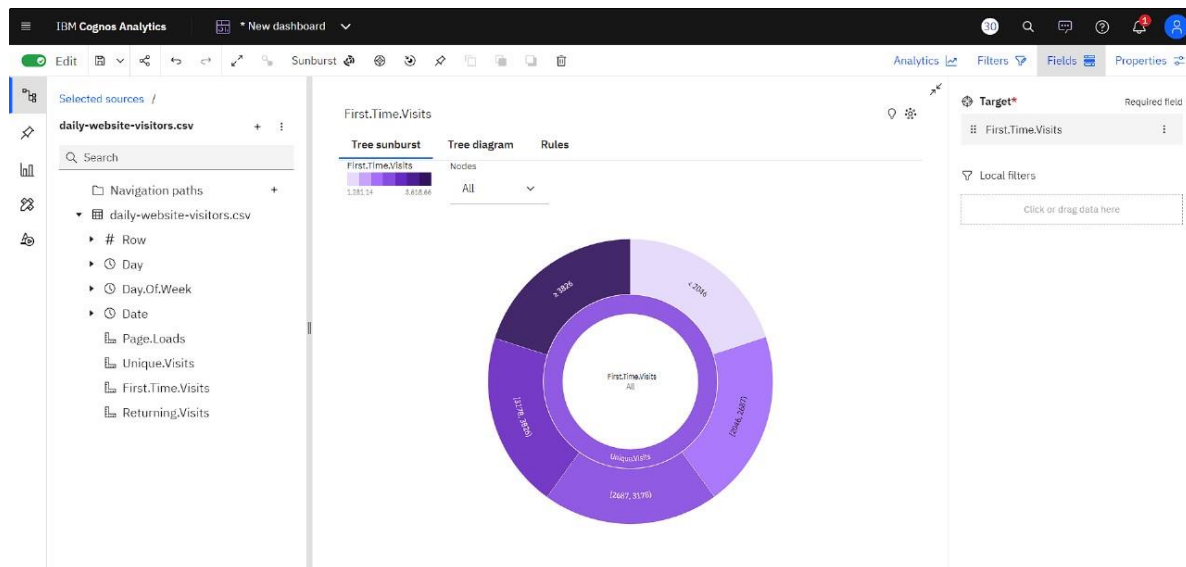
```
plt.title('Linear Regression Predictions')
plt.show()
```



WORKING WITH IBM COGNOS

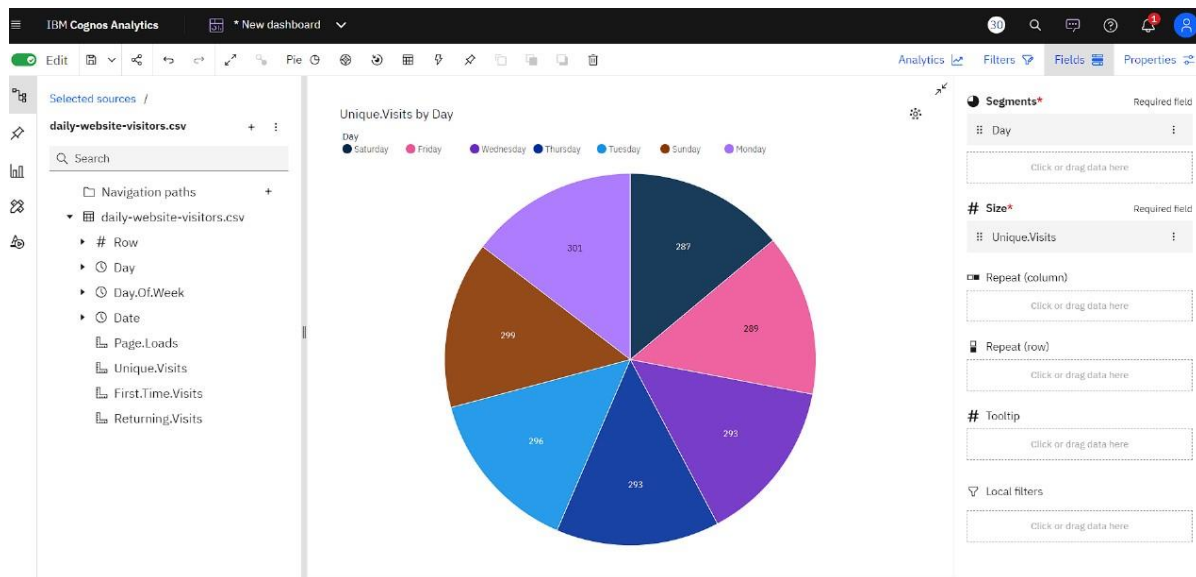
INSIGHTS GATHERED FROM IBM COGNOS

- Unique.Visits has a strong downward trend.
- Add insight to favorites
- Based on the current forecasting, Unique.Visits may reach 281.5 by Day Monday+1.
- Add insight to favorites
- Monday (14.3 %), Sunday (14.3 %), Wednesday (14.3 %), and Tuesday (14.3 %) are the most frequently occurring categories of Day with a combined count of 1240 items with Unique.Visits values (57.2 % of the total).
- Add insight to favorites
- The total number of results for Unique.Visits, across all days, is over two thousand.



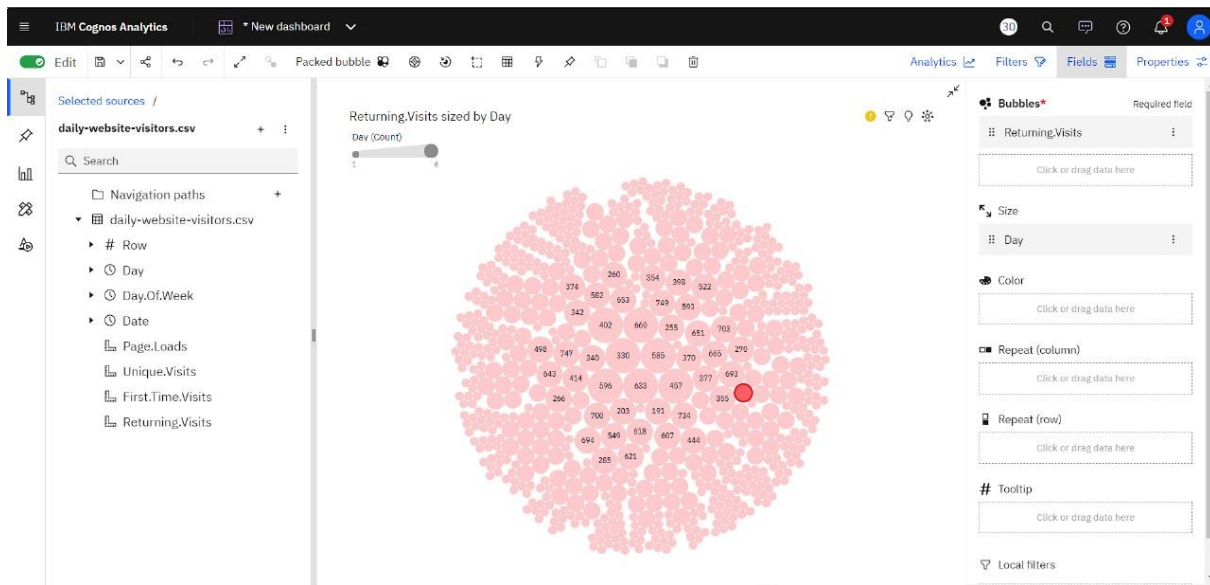
lowest average & highest average & forecasting

- Day Saturday has the lowest average First.Time.Visits at almost 1500, followed by Sunday at nearly 2 thousand.
- Add insight to favorites
- Day Tuesday has the highest average First.Time.Visits at 2928.23, followed by Wednesday at 2895.49.
- Add insight to favorites
- Based on the current forecasting, First.Time.Visits may reach almost 1500 by Day Monday+1.



All Returning.Visits

The total number of results for Day, across all Returning.Visits, is over two thousand.



CONCLUSION

In this phase the given website traffic analysis dataset is preprocessed through such activities like Predictive models and Visualization. Then the dataset is loaded into the IBM COGNOS to perform various visualizations and insights collected from them about the website traffic.