

DATA ANALYTICS WITH COGNOS

WEBSITE TRAFFIC ANALYSIS – PHASE 4

INTRODUCTION:

In Phase 4, the aim is to analyze website traffic data for insights into user behavior, popular pages, and traffic sources. It involves data collection, visualization using IBM Cognos and Python for advanced analysis. The goal is to optimize user experiences and enhance website performance.

DATASET LINK: <https://www.kaggle.com/datasets/bobnau/daily-website-visitors>

ABOUT DATASET:

The given data set has already been cleaned. Hence, no preprocessing is required.

- **Row:** An identifier or index for each row in the dataset used to identify each record uniquely.
- **Day:** Represents the day of the data record.
- **Day_of_week:** Indicates the day of the week represented as a numeric value (e.g., 1 for Sunday, 2 for Monday, etc.).
- **Date:** Denotes date of the entry where the entire data ranges over a period of 5 years.
- **Page Loads:** Number of page loads on the website for a given day. It's a count of how many times pages on the website were loaded or accessed.
- **Unique Visits:** Count of unique visitors to the website on a given day. It measures the number of distinct individuals whose IP addresses haven't been hits on any page in over 6 hours. A visit is classified as "unique" if a hit from same IP address has not come within the last 6 hours.
- **First-time Visits:** Indicates the number of first-time visitors to the website on a particular day. First-time visits typically represent new visitors who haven't visited the website before.
- **Returning Visits:** Count of returning visitors to the website on a specific day. Returning visits are from individuals who have visited the website before and are returning for subsequent visits.

DATA EXPLORATION:

1.PAGE LOADS AND RETURNING VISITS BY DAY :

Explore data relationships

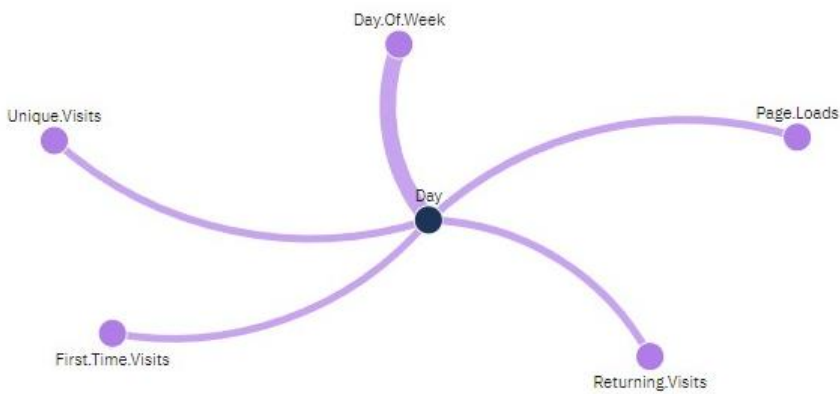
daily-website-visitors.csv

Reset to original

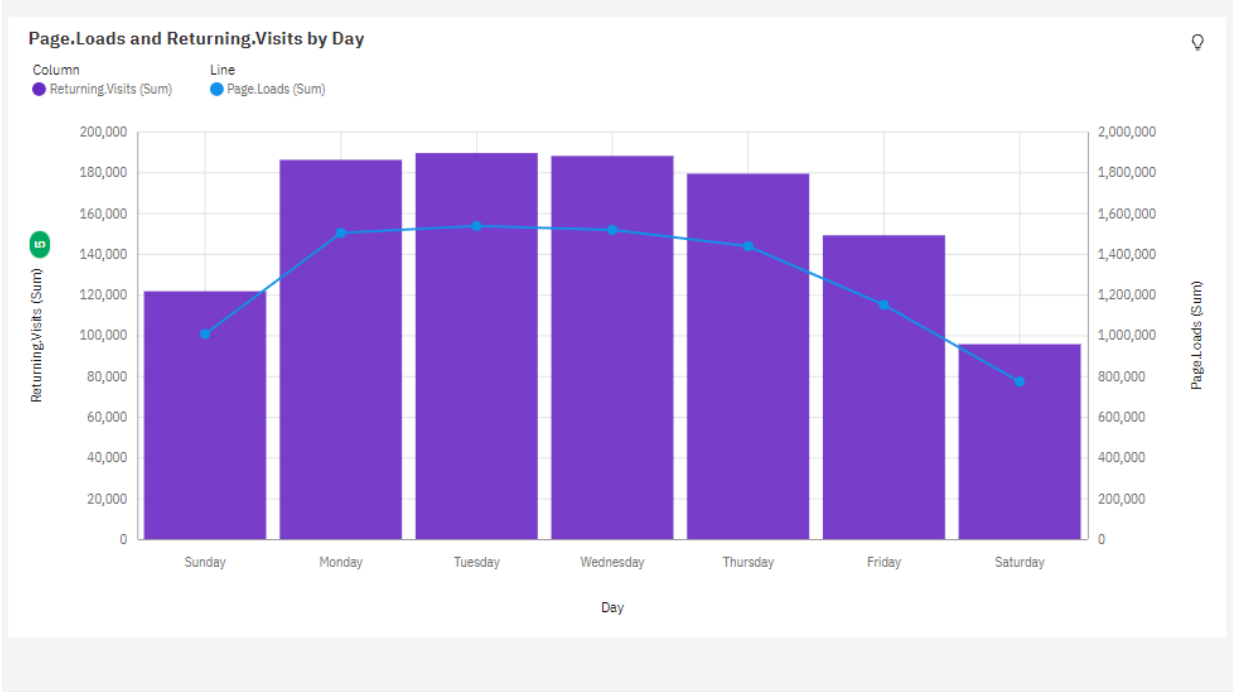
Day

x

Edit diagram



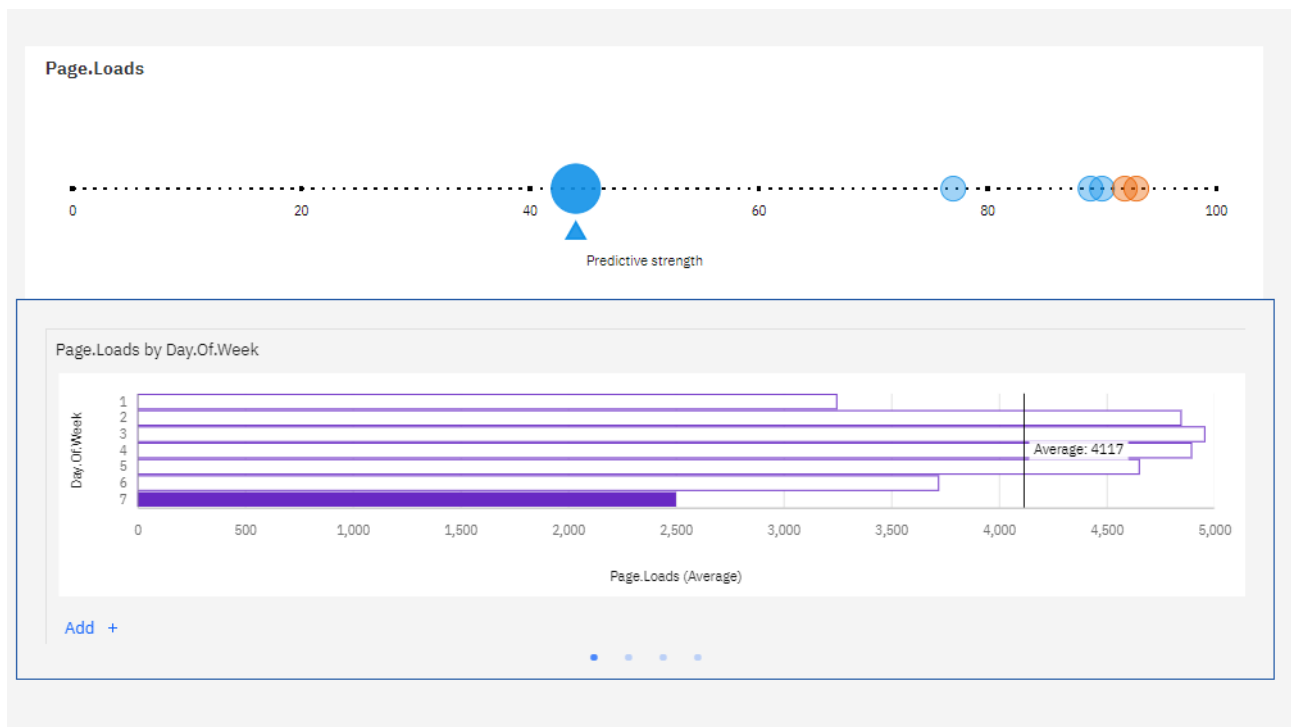
Select single or multiple nodes to see visualizations.



INSIGHTS:

- Across all days, the sum of Returning.Visits is over 1.1 million.
- Returning.Visits ranges from almost 96 thousand, when Day is Saturday, to over 189 thousand, when Day is Tuesday.
- Returning.Visits is unusually low when Day is Saturday.
- For Returning.Visits, the most significant values of Day are Tuesday, Wednesday, Monday, Thursday, and Friday, whose respective Returning.Visits values add up to almost 892 thousand, or 80.4 % of the total.
- Across all days, the sum of Page.Loads is over 8.9 million.
- Page.Loads ranges from nearly 773 thousand, when Day is Saturday, to over 1.5 million, when Day is Tuesday.
- Page.Loads is unusually low when Day is Saturday.
- For Page.Loads, the most significant values of Day are Tuesday, Wednesday, Monday, Thursday, and Friday, whose respective Page.Loads values add up to over 7.1 million, or 80.1 % of the total.

2.PAGE LOADS BY DAY OF WEEK :



INSIGHTS:

Across all values of Day.Of.Week, the average of Page.Loads is over four thousand.

The average values of Page.Loads range from over 2500, occurring when

Day.Of.Week is 7, to nearly five thousand, when Day.Of.Week is 3.

Day.Of.Week moderately affects Page.Loads (44%).

Page.Loads is unusually low when Day.Of.Week is 7.

1 (14.3 %), 2 (14.3 %), 3 (14.3 %), and 4 (14.3 %) are the most frequently occurring categories of Day.Of.Week with a combined count of 1240 items with Page.Loads values (57.2 % of the total).

3.UNIQUE VISITS BY FIRST TIME VISITS AND RETURNING VISITS:



INSIGHTS:

- First.Time.Visits (Group) (3) strongly affects Unique.Visits (94%).
- Unique.Visits is most unusual when First.Time.Visits (Group) (3) is 3934 and above and less than 1205.
- Returning.Visits (Group) (2) strongly affects Unique.Visits (76%).
- Unique.Visits is unusually high when Returning.Visits (Group) (2) is 886 and above.

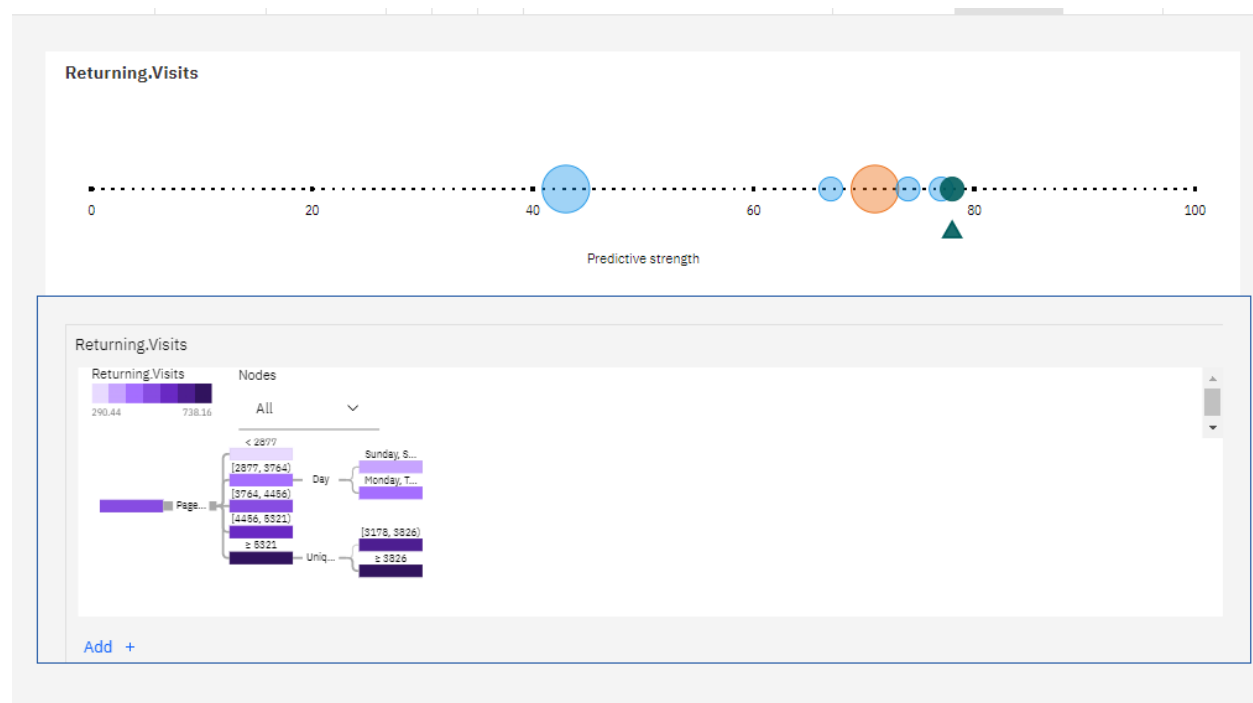
- Over all values of First.Time.Visits (Group) (3) and Returning.Visits (Group) (2), the average of Unique.Visits is nearly three thousand.
- The average values of Unique.Visits range from over a thousand to over five thousand.
- First.Time.Visits (Group) (3) and Returning.Visits (Group) (2) strongly affect Unique.Visits (96%).
- Unique.Visits is unusually high when the combination of First.Time.Visits (Group) (3) and Returning.Visits (Group) (2) is 3934 and above and 886 and above.
- 1887 to < 2569 is the most frequently occurring category of First.Time.Visits (Group) (3) with a count of 666 items with Unique.Visits values (30.7 % of the total).
- 434 to < 585 is the most frequently occurring category of Returning.Visits (Group) (2) with a count of 734 items with Unique.Visits values (33.9 % of the total).
- There is no significant impact of Returning.Visits (Group) (2) on the relationship between First.Time.Visits (Group) (3) and Unique.Visits.



INSIGHTS:


- Unique.Visits is unusually high when the combination of First.Time.Visits (Group) (3) and Returning.Visits (Group) (2) is 3934 and above and 886 and above.
- 1887 to < 2569 is the most frequently occurring category of First.Time.Visits (Group) (3) with a count of 666 items with Unique.Visits values (30.7 % of the total).
- 434 to < 585 is the most frequently occurring category of Returning.Visits (Group) (2) with a count of 734 items with Unique.Visits values (33.9 % of the total).
- There is no significant impact of Returning.Visits (Group) (2) on the relationship between First.Time.Visits (Group) (3) and Unique.Visits.

5.FIRST TIME VISITS AND RETURNING VISITS BY DATE :



INSIGHTS:

- Page.Loads, Unique.Visits, and Day predict Returning.Visits with a strength of 78.1%.
- Page.Loads is the most significant predictor of Returning.Visits being three times better than any other field.

jupyter Untitled1 Last Checkpoint: 13 hours ago (autosaved)  Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

```
In [38]: import numpy as np
import pandas as pd
import warnings
warnings.filterwarnings('ignore')
import datetime
from datetime import date


import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set_style("whitegrid")

# import chart_studio.plotly as py
import plotly.express as px

from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
init_notebook_mode(connected=True)

import plotly.graph_objects as go

from sklearn.model_selection import train_test_split, cross_val_score, GridSearchCV
from sklearn.metrics import accuracy_score
from sklearn.svm import SVR
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
# from prophet import Prophet
```

jupyter Untitled1 Last Checkpoint: 13 hours ago (autosaved)  Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

```
In [40]: df=pd.read_csv('D:/daily-website-visitors.csv')

df.rename(columns = {'Day.Of.Week':'day_of_week'
                    , 'Page.Loads':'page_loads'
                    , 'Unique.Visits':'unique_visits'
                    , 'First.Time.Visits':'first_visits'
                    , 'Returning.Visits':'returning_visits'}, inplace = True)

df=df.replace(' ','',regex=True)

df['page_loads']=df['page_loads'].astype(int)
df['unique_visits']=df['unique_visits'].astype(int)
df['first_visits']=df['first_visits'].astype(int)
df['returning_visits']=df['returning_visits'].astype(int)

df
```

Out[40]:

	Row	Day	day_of_week	Date	page_loads	unique_visits	first_visits	returning_visits
0	1	Sunday	1	9/14/2014	2146	1582	1430	152
1	2	Monday	2	9/15/2014	3621	2528	2297	231
2	3	Tuesday	3	9/16/2014	3698	2630	2352	278
3	4	Wednesday	4	9/17/2014	3667	2614	2327	287
4	5	Thursday	5	9/18/2014	3316	2366	2130	236
...
2162	2163	Saturday	7	8/15/2020	2221	1696	1373	323

2167 rows x 8 columns

In [41]: df.isna().sum()

```
Out[41]: Row      0
Day      0
day_of_week  0
Date      0
page_loads  0
unique_visits  0
first_visits  0
returning_visits  0
dtype: int64
```

In [42]: df.duplicated().sum()

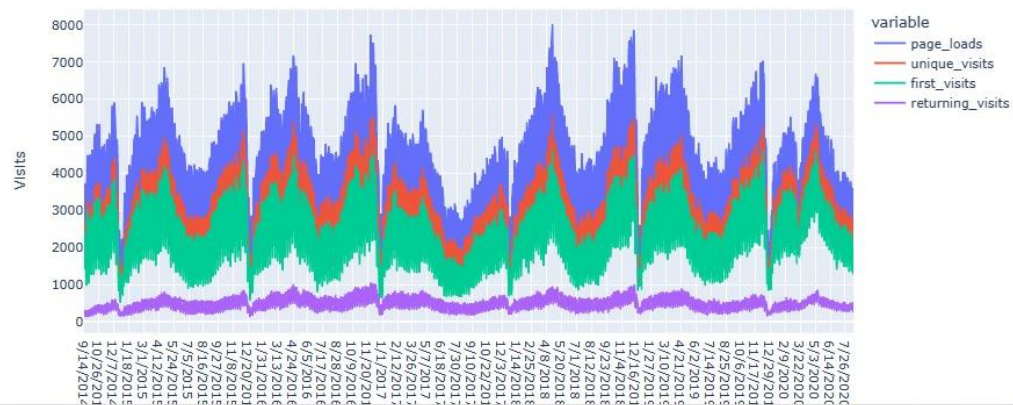
Out[42]: 0

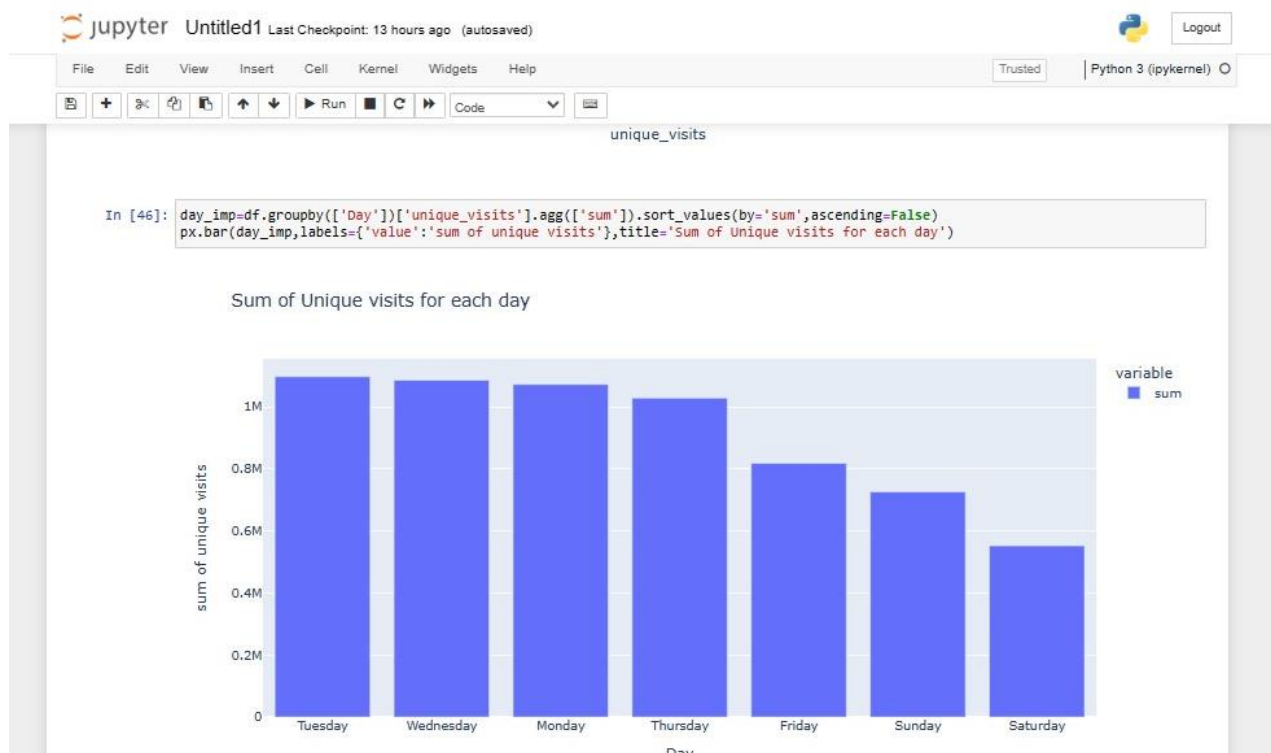
In [43]: df.info()

```
<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   object
4   page_loads   2167 non-null   int32
```

```
In [44]: px.line(df,x='Date',y=['page_loads','unique_visits','first_visits','returning_visits'],
labels={'value':'Visits'},
,title='Page Loads & visitors over Time')
```

Page Loads & visitors over Time





CONCLUSION:

Website traffic analysis using IBM Cognos Analytics allows businesses to gain valuable insights into the performance of their websites. This analysis helps in making data-driven decisions to enhance user experience, optimize content, and improve marketing strategies. IBM Cognos Analytics provides robust tools for tracking and visualizing web traffic data, enabling organizations to monitor key metrics, detect trends, and make informed decisions to drive business success. It offers the ability to create interactive reports and dashboards, making it easier for teams to collaborate and act on the insights derived from website traffic data, ultimately leading to improved online performance and user engagement.