Website Traffic Analysis – Phase 5

TEAM MEMBERS :

KAMARAJ	2021115047
KARTHIKA	2021115049
KARTHIKEYAN	2021115050
KAVIYA	2021115051
JEEVESH	2021115312

PROBLEM STATEMENT:

This project aims to analyze website traffic data to extract valuable insights into user behavior, page popularity, and traffic sources, with the overarching goal of assisting website owners in enhancing the user experience. This endeavor involves defining specific analysis objectives, ensuring the comprehensive collection of website traffic data, utilizing IBM Cognos for data visualization, and integrating Python code to perform advanced analytics. By achieving these milestones, the project endeavors to empower website owners with actionable knowledge that will enable them to optimize their web content, marketing strategies, and overall website performance, aligning with broader organizational objectives.

DESIGN THINKING:

1.) Analysis Objectives:

Primarily, we have to define the objectives on what are things that we are going to extract from the data given to us. So at the end we will be able to get clear insights on

- The time intervals at which the traffic is maximum
- Compare traffic on different days and times
- Perform useful predictions to enhance user experience

2.) Data Collection:

We will be using the dataset provided by kaggle.com to carry on this project
https://www.kaggle.com/datasets/bobnau/daily-website-visitors

The above dataset contains necessary data like day, date etc. It also contains number of unique visits, first visits and returning visits which will be very helpful for us to enhance the user experience by identifying what they need the most.

3.) Visualization Strategy:

We will employ IBM Cognos, a robust data visualization platform, to create insightful visual representations of the collected data. This will include charts, graphs, dashboards, and reports that effectively communicate the findings to stakeholders. The visuals on which we will be most interest on are

- Graphs that depicts user traffic on different days
- Bar graphs which will be useful to compare between traffics

- IBM Cognos allows for the creation of interactive dashboards that provide a holistic view of website performance.
- Beyond predefined dashboards, IBM Cognos enables data exploration. Users can interact with visualizations, apply filters, and drill down into specific data points to uncover deeper insights

4.) Insights Generation:

Insight generation is a pivotal phase in the project, encompassing the extraction of actionable knowledge from website traffic data analysis. This process involves several key components:

- Identifying recurring patterns or trends within the data is essential. For example, spikes in website traffic during specific times can inform optimal marketing campaign scheduling.
- Data can be segmented into categories (e.g., demographics, traffic sources) to reveal distinct user behaviors and preferences, enabling personalized strategies.
- Predictive modeling helps forecast future trends, assisting in proactive planning and mitigation.

Project Definition:

Analysis Objective:

The primary objective of this project is to enhance web traffic analysis by employing design thinking principles to define the problem and leveraging Python for efficient data collection and visualization.

Data Collection:

Data collection is a critical aspect of web traffic analysis. We will explore various data sources, such as web server logs, Google Analytics, and user surveys, to gather comprehensive information about user interactions, demographics, and preferences. Python will be used to automate data retrieval and preprocessing, ensuring a streamlined and error-free data collection process.

Visualization:
Data can be hard to understand. We'll use Python to make cool charts and graphs that show us the important stuff. This will help us see what's working and what needs improvement.

Python Integration:
Python is a handy tool that will help us with many parts of this project. We'll use it to bring data together, make it clean and organized, and create easy-to-understand pictures from the data.

Common Machine Learning Models in Web Traffic Analysis

1. Regression Models

- Linear Regression
- Polynomial Regression
- Logistic Regression
- Purpose: Predict web traffic volume, conversion rates, and numerical metrics.

2. Classification Models

- Decision Trees
- Random Forests
- Support Vector Machines (SVMs)
- Purpose: Identify malicious traffic, categorize user behavior, and classify user interactions.

3. Clustering Models

- K-Means Clustering
- Hierarchical Clustering
- Purpose: Group users with similar behavior patterns for segmentation and personalization.

5. Ensemble Models

- Bagging (e.g., Random Forests)
- Boosting (e.g., AdaBoost, XGBoost)

- Purpose: Combine multiple models for more accurate predictions and robustness.

Machine Learning Models with Cognos

Step 1: Data Collection and Preparation

Data Sources

- Identify and gather relevant data sources including web server logs, user interactions, historical traffic data, and any additional data required for analysis.

Data Preprocessing

- Clean and preprocess the data:
  - Handle missing values.
  - Remove duplicates.
  - Normalize or scale numerical features.
  - Encode categorical variables.
  - Address outliers.

Data Integration

- Integrate data sources into a unified data repository, ensuring data consistency and quality.

Step 2: Feature Engineering

Feature Selection

- Identify relevant features that can influence web traffic and user behavior, such as time-based features, user demographics, and content-related features.

Feature Engineering

- Create new features or transform existing ones to capture meaningful patterns and insights.

Step 3: Model Selection and Training

Model Selection

- Choose machine learning models based on the nature of the problem:

- Time series forecasting models for traffic trends.
- Classification and clustering models for user behavior analysis.

Model Training

- Train the selected models using historical data, splitting it into training and validation sets.
- Fine-tune model hyperparameters to optimize performance.
- Implement ensemble models for improved accuracy if needed.

Step 4: Evaluation and Validation

Model Evaluation

- Evaluate model performance using appropriate metrics:
  - For time series forecasting: RMSE, MAE, MAPE.
  - For classification: Accuracy, F1-score, ROC AUC.
  - For clustering: Silhouette score, Davies-Bouldin index.

Validation

- Perform cross-validation to ensure model robustness.
- Validate results against ground truth data or user feedback.

Step 5: Model Deployment

Deployment Environment

- Set up a production environment for deploying machine learning models:
  - Choose cloud or on-premises infrastructure.
  - Ensure scalability and reliability.

Integration with Web Analytics

- Integrate the machine learning models with web analytics tools, ensuring seamless data flow.

Step 6: Monitoring and Maintenance

Continuous Monitoring

- Implement monitoring systems to track model performance in real-time.
- Set up alerts for anomalies or deteriorating performance.

Retraining

- Establish a retraining schedule to keep models up-to-date with changing web traffic patterns and user behavior.

Step 7: Reporting and Visualization

Dashboard Creation

- Develop interactive dashboards in tools like Cognos to present web traffic analysis results.
- Include visualizations for trends, user behavior insights, and predictions.

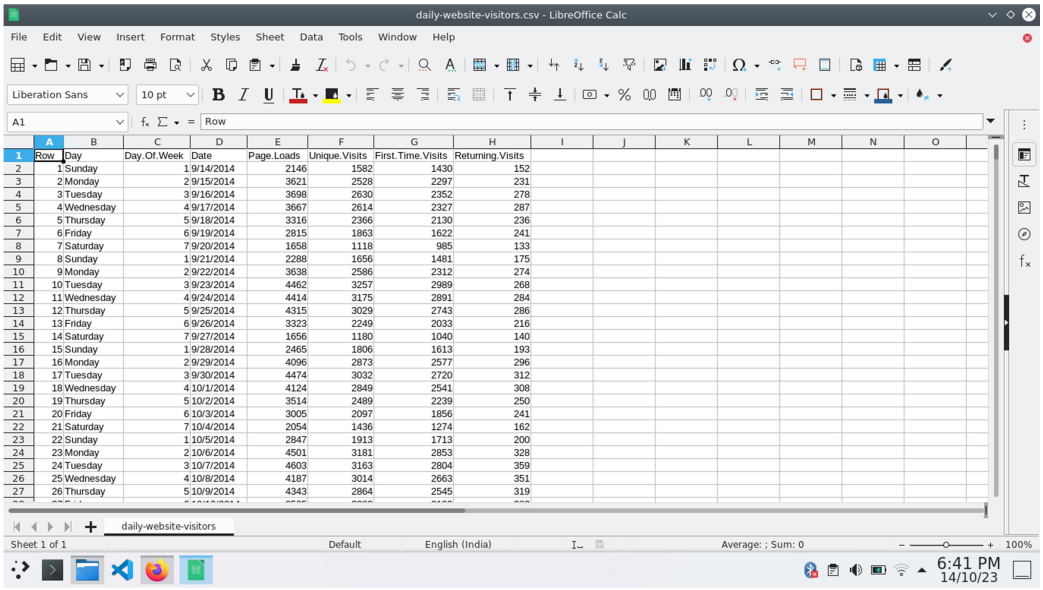
Report Generation

- Generate regular reports for stakeholders, summarizing key findings, trends, and recommendations.

Data source

Dataset is collected from the kaggle.com named “daily-website-visitors.csv” which has a data about the Days, Day of week, Date, page Loads, Unique visits, First-time visits, Returning Visits.

Dataset link: https://www.kaggle.com/datasets/bobnau/daily-website-visitors

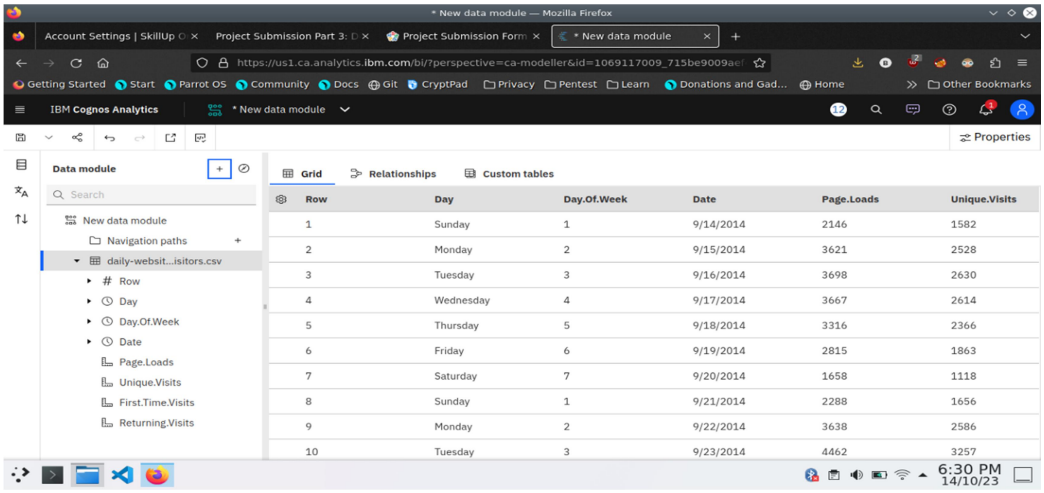


Row	Day	Day.Of.Week	Date	Page.Loads	Unique.Visits	First.Time.Visits	Returning.Visits
1	Sunday	1	9/14/2014	2146	1582	1430	152
2	Monday	2	9/15/2014	3621	2528	2297	231
3	Tuesday	3	9/16/2014	3698	2630	2352	278
4	Wednesday	4	9/17/2014	3667	2614	2327	287
5	Thursday	5	9/18/2014	3316	2366	2130	236
6	Friday	6	9/19/2014	2815	1863	1622	241
7	Saturday	7	9/20/2014	1658	1118	985	133
8	Sunday	1	9/21/2014	2288	1656	1481	175
9	Monday	2	9/22/2014	3638	2586	2312	274
10	Tuesday	3	9/23/2014	4462	3257	2989	268
11	Wednesday	4	9/24/2014	4414	3175	2891	284
12	Thursday	5	9/25/2014	4315	3029	2743	286
13	Friday	6	9/26/2014	3323	2249	2033	216
14	Saturday	7	9/27/2014	1656	1180	1040	140
15	Sunday	1	9/28/2014	2465	1806	1613	193
16	Monday	2	9/29/2014	4096	2873	2577	296
17	Tuesday	3	9/30/2014	4474	3032	2720	312
18	Wednesday	4	10/1/2014	4124	2849	2541	308
19	Thursday	5	10/2/2014	3514	2489	2239	250
20	Friday	6	10/3/2014	3005	2097	1856	241
21	Saturday	7	10/4/2014	2054	1436	1274	162
22	Sunday	1	10/5/2014	2847	1913	1713	200
23	Monday	2	10/6/2014	4501	3181	2853	328
24	Tuesday	3	10/7/2014	4603	3163	2804	359
25	Wednesday	4	10/8/2014	4187	3014	2663	351
26	Thursday	5	10/9/2014	4343	2864	2545	319

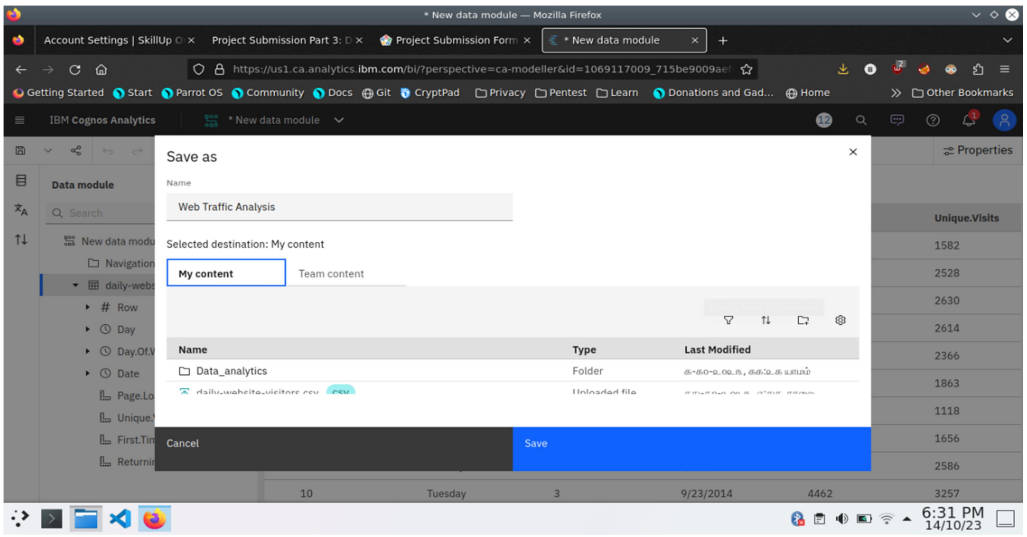
Data Loading

Steps Involved in data loading on IBM cognos.

1. Login to your IBM cognos
2. Click more menu from the left side
3. Select new tab
4. Click Data module tab
5. Upload the dataset for your project and select the Corresponding file
6. preview the data
7. Explore the data



8. save the data module



Data Preprocessing and Cleaning

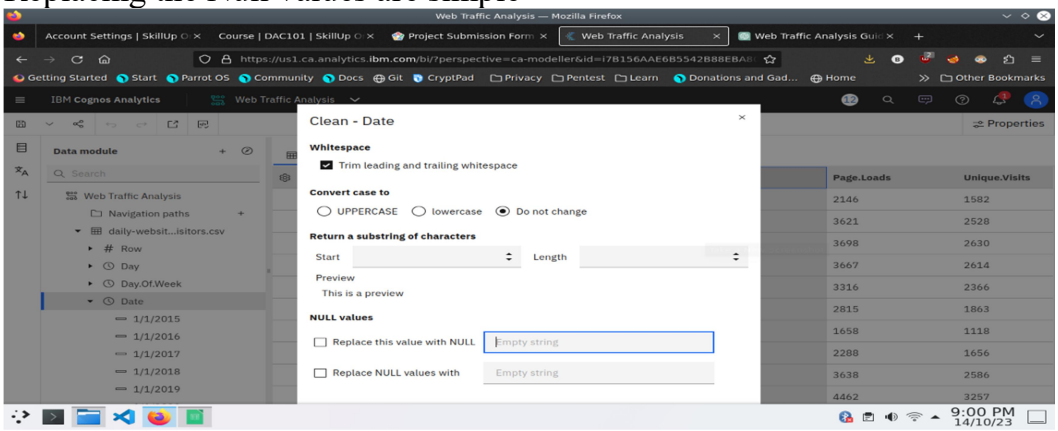
In this phase the following steps will taken

- Handling missing data
- Data Transformation
- Data Type Conversion
- Removing Duplicates
- Dealing Outliers

Once you saved the data module. Click the corresponding dataset on IBM cognos and Preview the mosule

Right Click the row where you want to clean the data

It provides the UI to Clean the data and makes the task easy one, Now Updating and Replacing the Null values are simple

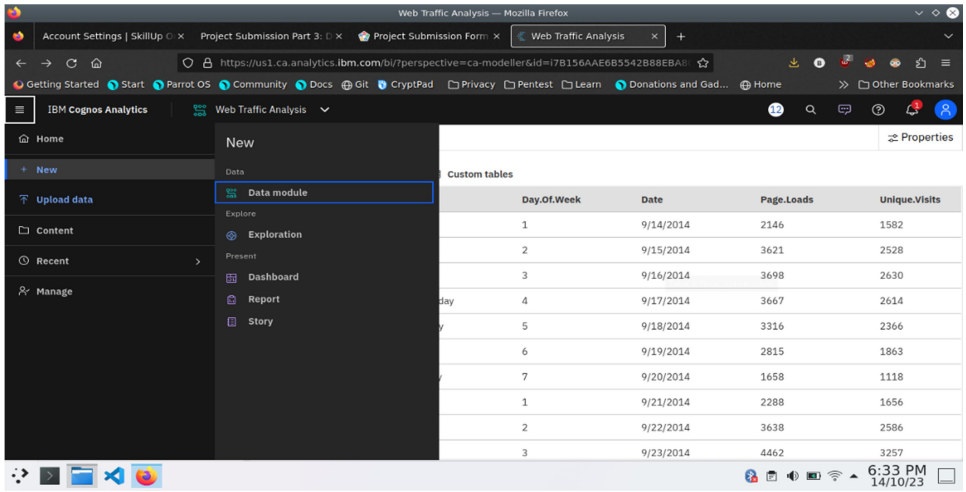


data module will be updated by doing the above process
after the completion of process start creating the dashboard for Visualization

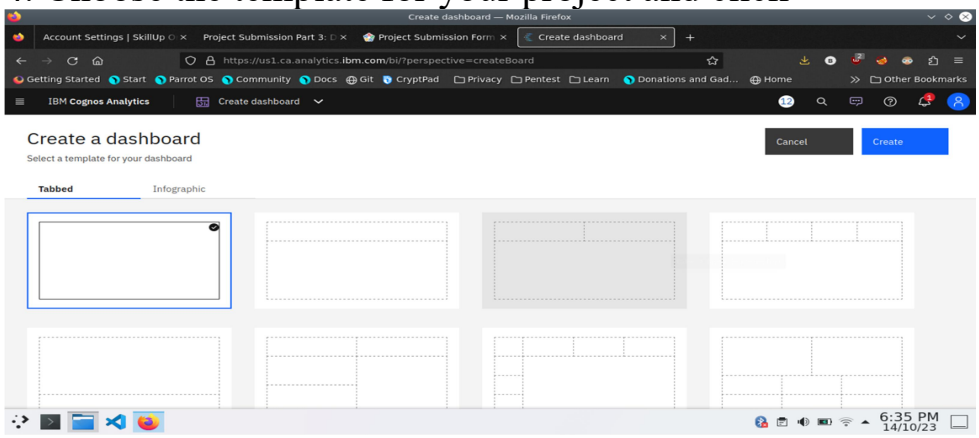
Dashboard Creation

Dashboard creation are helpful to visualizing the data

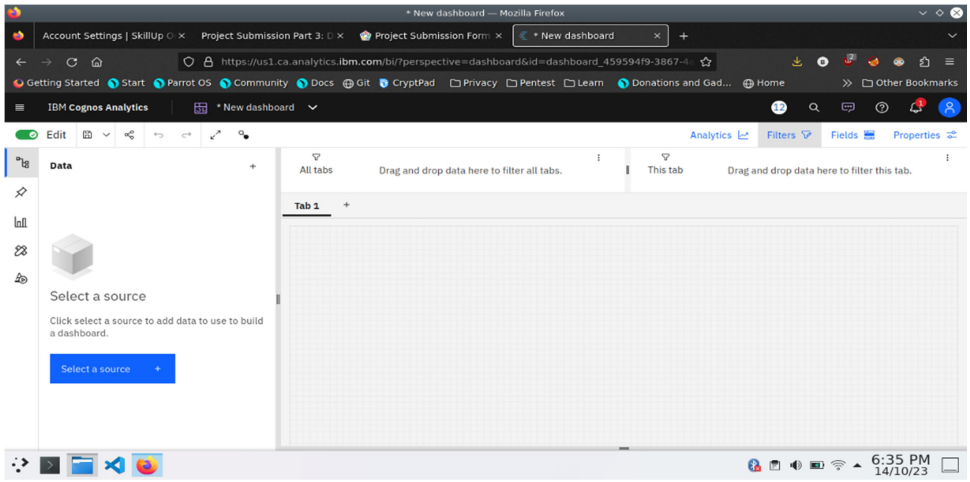
- Goto Home menu
- Select the new tab
- Click dashboard



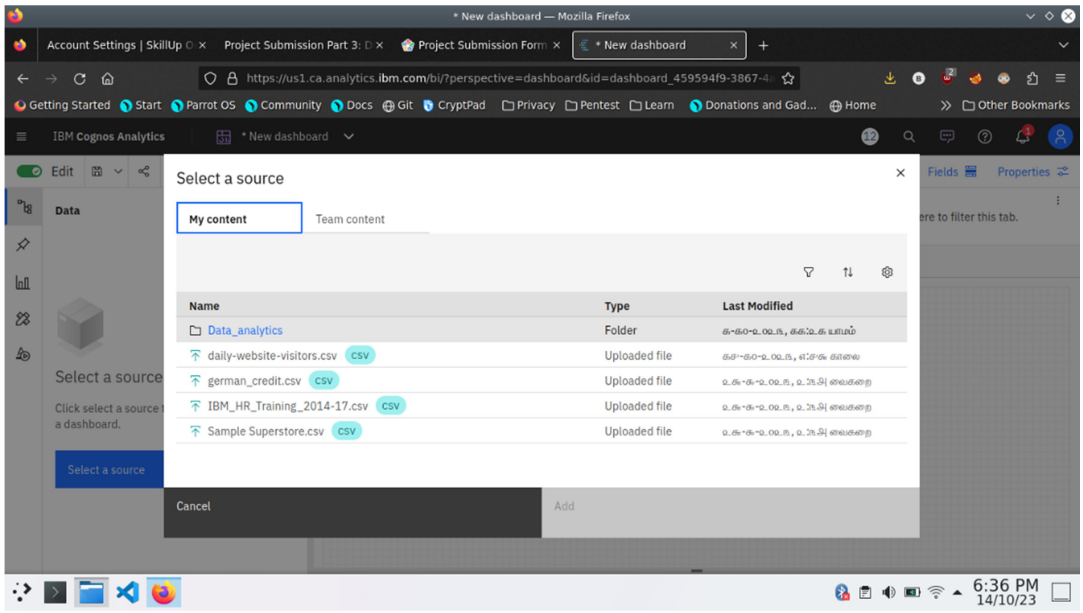
4. Choose the template for your project and click



5.Now Dashboard is created



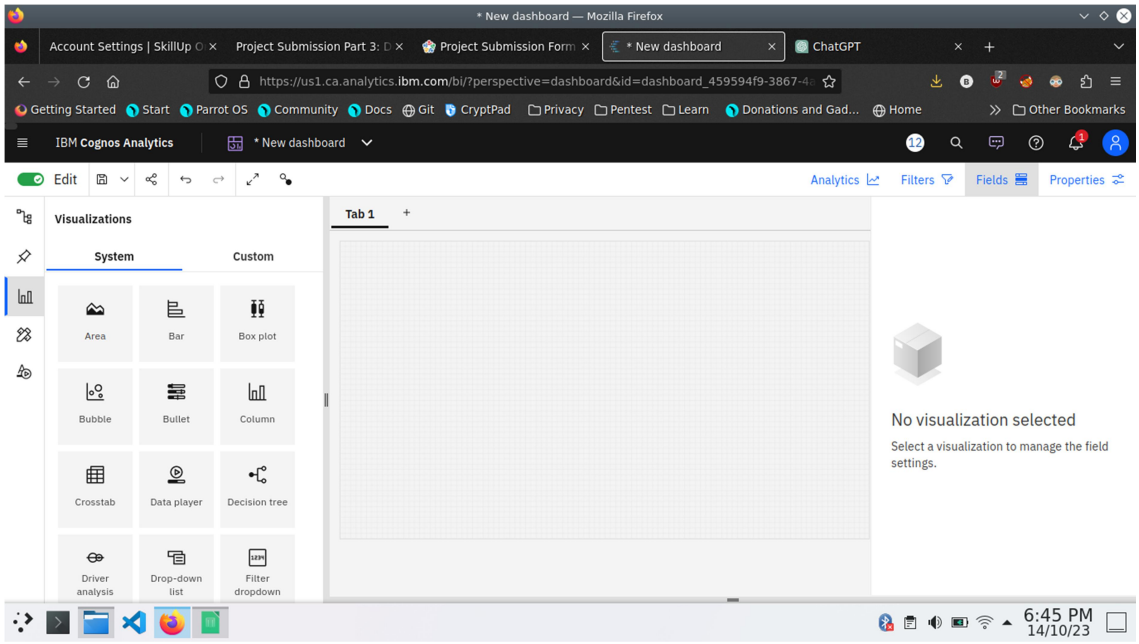
6. Select the data source



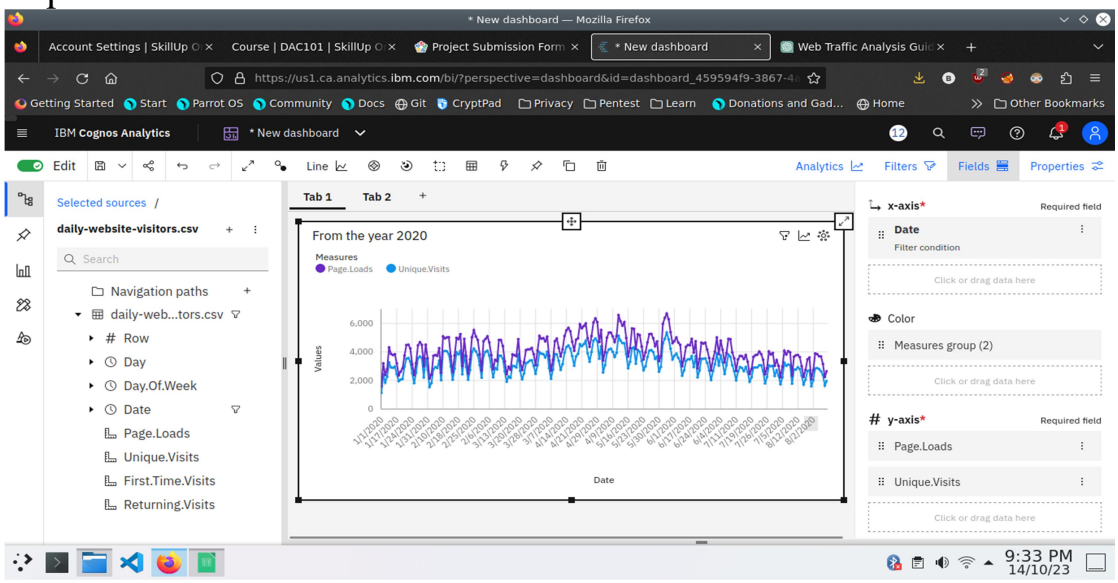
Visualization

After creating the dashboard, the next step is to visualize the data In IBM Cognos

1. Goes to the Corresponding Dashboard
2. select the visualizations tab in the left side of title bar



3.Choose the system as you want and put the data source for the required columns

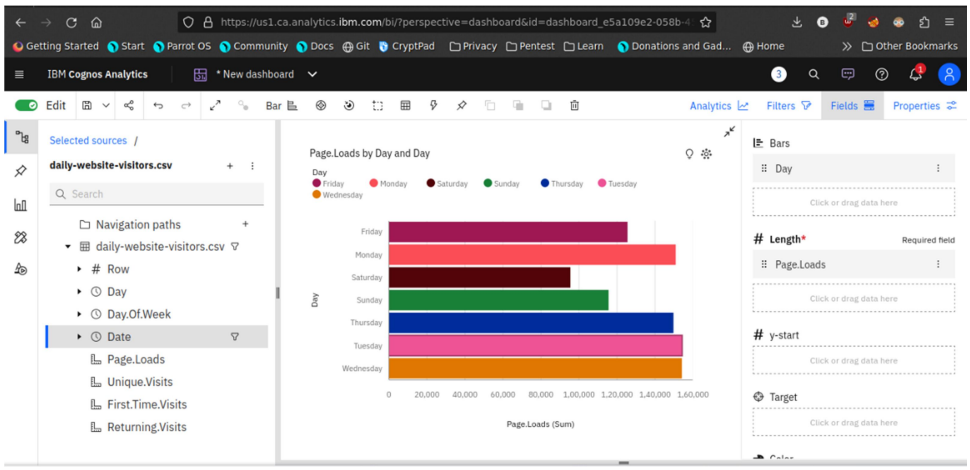


In the above screen shot displays the Line graph and model compares the “ Page.loads” and “Unique.visits” from the time period of 2020

X-axis =Dates
Y-axis = Page.Loads, Unique.visits.

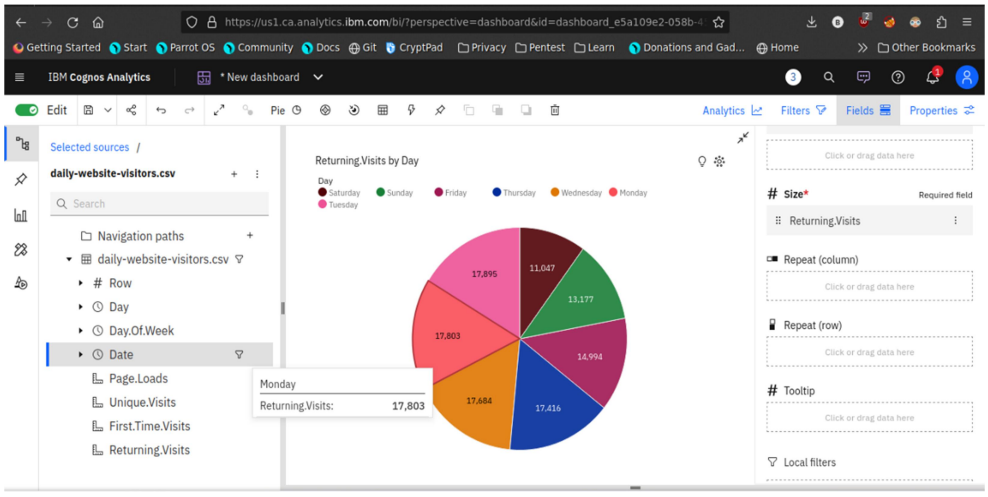
Barchart

In this Bar chart the bars represent the ‘days in week’ and length defines ‘Page.Loads’
It helpful to visualize the maximum pageloads occurs on a day



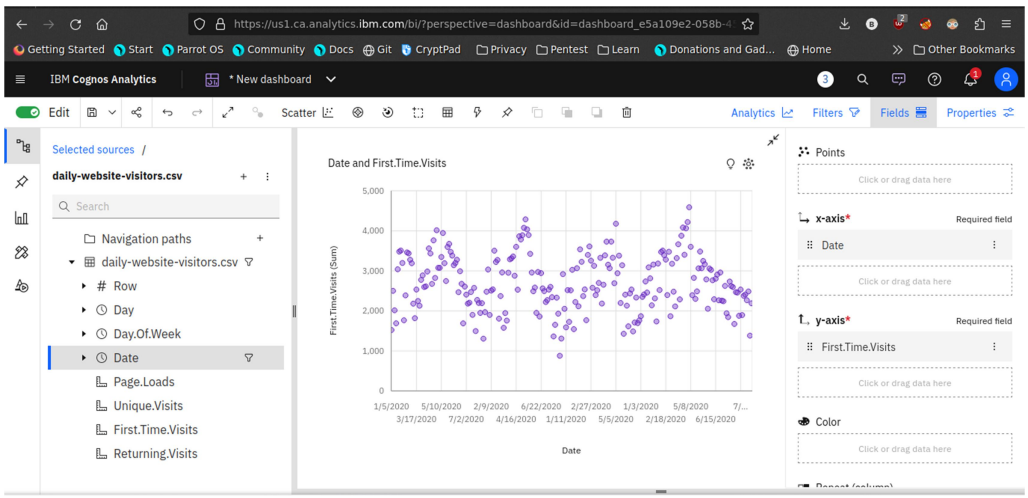
Pie chart

This is same as a bar chart. it helpful to analyze the Returing visits occurs on a particular day

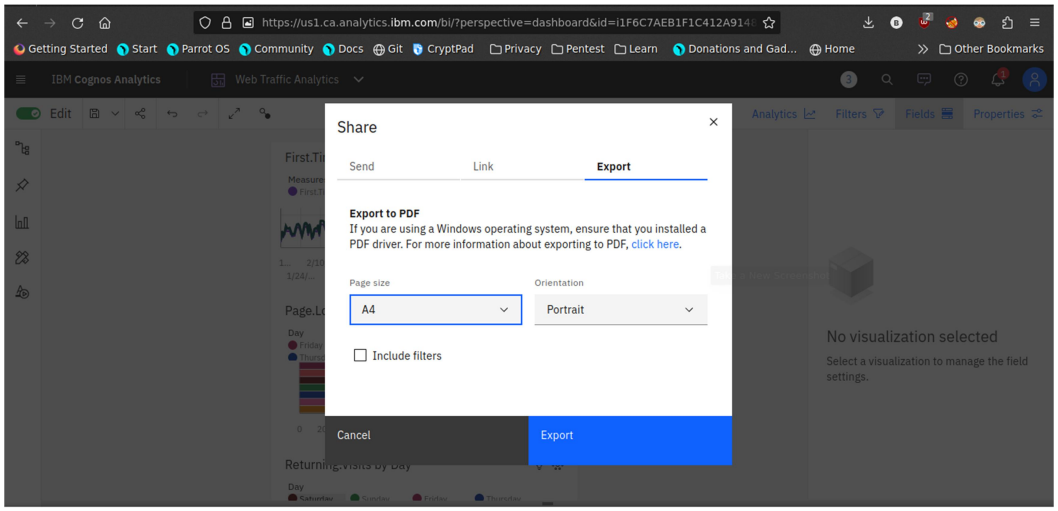


Scatter plot

It is used to display the relationship between two variables and observe the nature of the relationship. The relationships observed can either be positive or negative, non-linear or linear

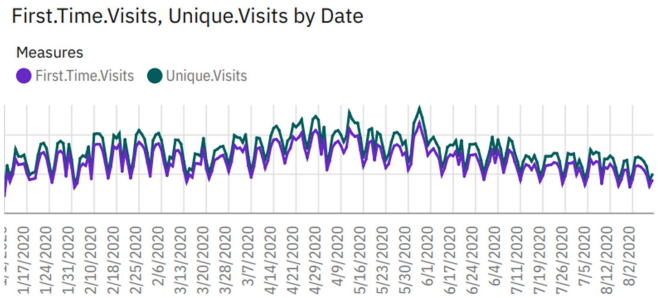


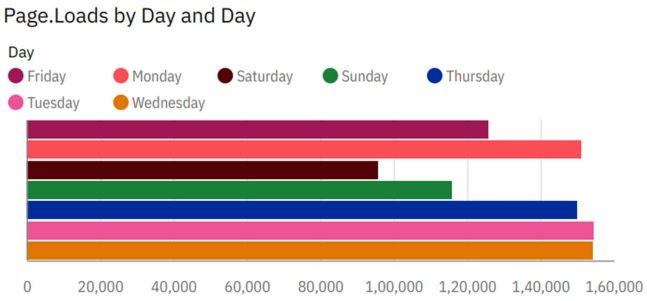
- After completing the dashboard lets export to another format Like pdf

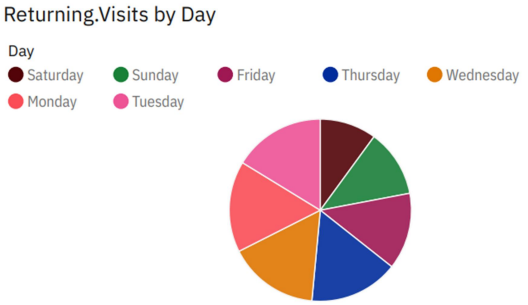


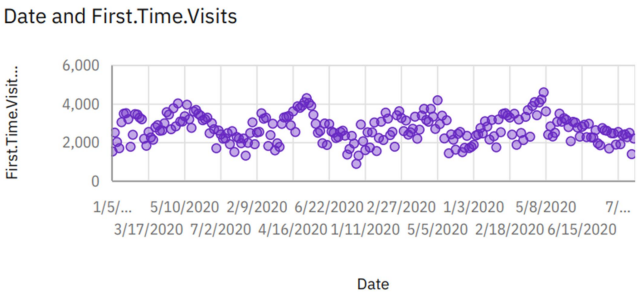
After Completing the above process the complicative report will be generated

Overall, the website traffic analysis dashboard provides website owners with a comprehensive and easy-to-use tool to analyze their website traffic and gain insights into user behavior and content performance. This information can be used to improve the user experience, increase engagement, and boost conversions.









Now the visualization phase where over. lets start analyze the dataset using Python libraries use machine learning models for predictive analysis.

Data Analysis using python

In this steps are used to analyse the given dataset using python libraries

Steps:

- **Import Necessary packages**
  Pandas
  seaborn
  Machine learning models
  Linear regression
- **Make a training and test data**
  Use the train test split model
  Compare the testing and training data set by visualization library
- **Calculate the accuracy of the model**
  Use r2_score to measure the accuracy of the model

In [1]:
```
import numpy as np
import pandas as pd
from pydantic_settings import BaseSettings
import warnings
from warnings import warn
import datetime
from datetime import date

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

import plotly.express as px
from sklearn.linear_model import LinearRegression
```

In [2]:
```
import pandas as pd
df=pd.read_csv("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[2]:

	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
2163	2164	Sunday	1	8/16/2020	2724	2037	1686	351
2164	2165	Monday	2	8/17/2020	3456	2638	2181	457
2165	2166	Tuesday	3	8/18/2020	3581	2683	2184	499
2166	2167	Wednesday	4	8/19/2020	2064	1564	1297	267

2167 rows × 8 columns

In [11]:
```
df.isna().sum()
```

Out[11]:
```
Row                0
Day                0
day_of_week        0
Date               0
page_loads         0
unique_visits      0
first_visits       0
returning_visits   0
dtype: int64
```

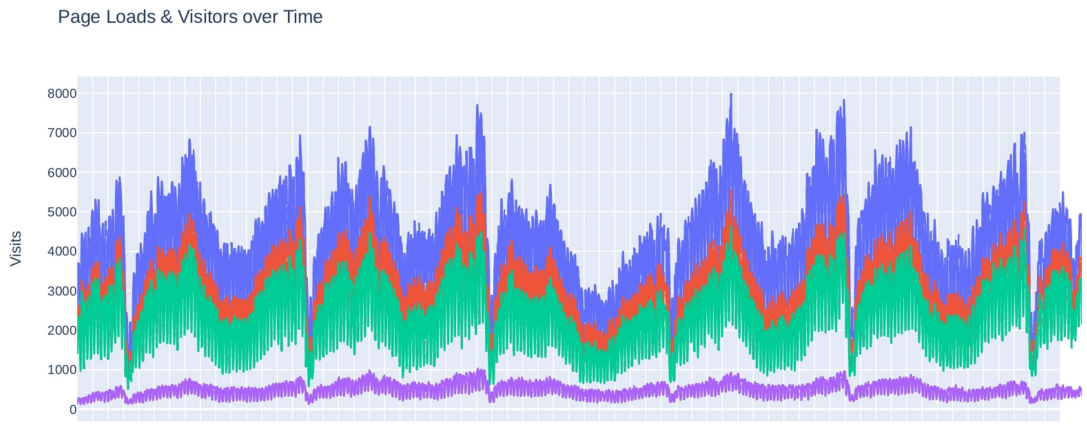
In [12]:
```
df.duplicated().sum()
```

Out[12]:
```
0
```

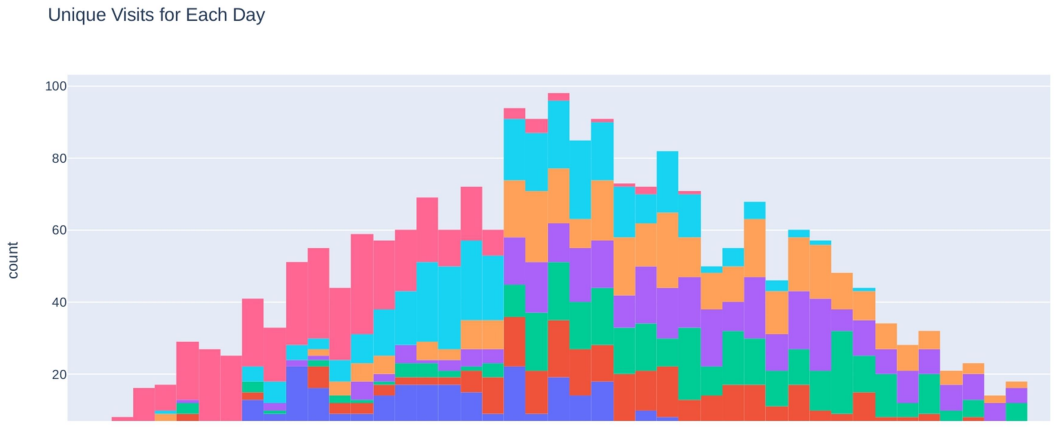
In [13]:
```
df.info()
```

```
<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   int64 
 5   unique_visits     2167 non-null   int64 
 6   first_visits      2167 non-null   int64 
 7   returning_visits  2167 non-null   int64 
dtypes: int64(6), object(2)
memory usage: 135.6+ KB
```

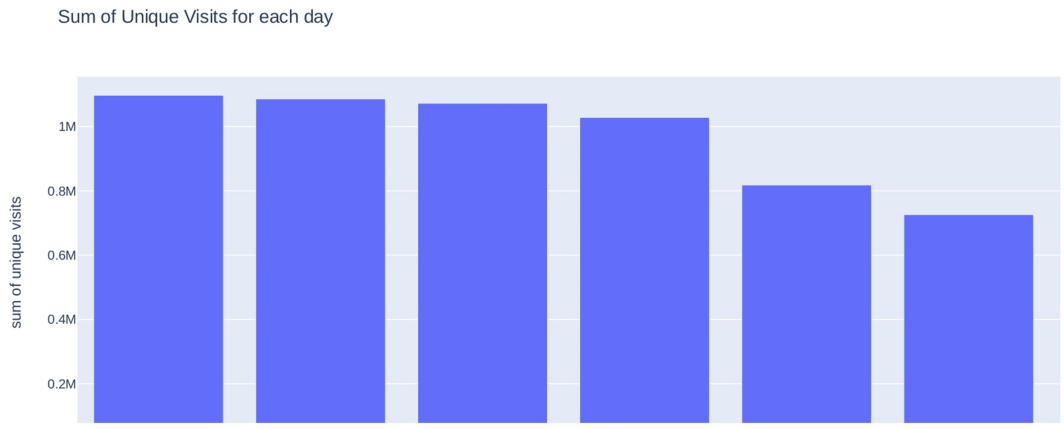
In [3]:
```
px.line(df, x='Date', y=['page_loads', 'unique_visits', 'first_visits', 'returning_visits'],
labels={'value': 'Visits'},
title='Page Loads & Visitors over Time')
```



In [4]:
```
px.histogram(df,x='unique_visits',color='Day',title='Unique Visits for Each Day')
```



In [5]:
```
day_imp=df.groupby(['Day'])['unique_visits'].agg(['sum']).sort_values(by='sum',ascending=False)
px.bar(day_imp,labels={'value':'sum of unique visits'},title='Sum of Unique Visits for each day')
```



In [6]:
```
px.histogram(df,x='Date',y='unique_visits',color='Day',title='Sum of Unique vists for each day over Time')
```

In [20]:
```
sums=df.groupby(['Day'])[['page_loads' ,'unique_visits' ,'first_visits' ,'returning_visits']].sum().sort_values(
    by='unique_visits',ascending=False)
sums
```

Out[20]:

Day	page_loads	unique_visits	first_visits	returning_visits
Tuesday	1536154	1097181	907752	189429
Wednesday	1517114	1085624	897602	188022
Monday	1502161	1072112	886036	186076
Thursday	1437269	1028214	848921	179293
Friday	1149437	817852	668805	149047
Sunday	1006564	725794	604198	121596
Saturday	772817	552105	456449	95656

In [7]:
```
px.bar(sums,barmode='group')
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[7], line 1
----> 1 px.bar(sums,barmode='group')

NameError: name 'sums' is not defined
```

In [19]:
```
pip install seaborn
```

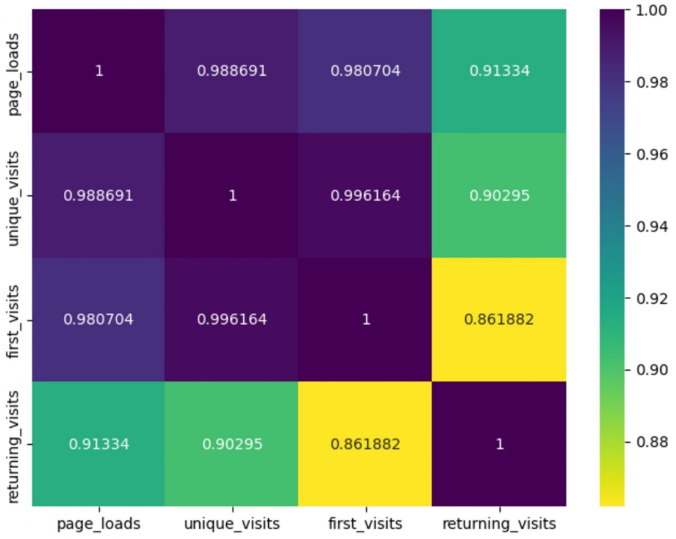
```
Requirement already satisfied: seaborn in /home/badhrinathan/.local/lib/python3.9/site-packages (0.12.2)
Requirement already satisfied: numpy!=1.24.0,>=1.17 in /home/badhrinathan/.local/lib/python3.9/site-packages (from seaborn) (1.24.3)
Requirement already satisfied: pandas>=0.25 in /home/badhrinathan/.local/lib/python3.9/site-packages (from seaborn) (2.0.3)
Requirement already satisfied: matplotlib!=3.6.1,>=3.1 in /usr/lib/python3/dist-packages (from seaborn) (3.3.4)
Requirement already satisfied: python-dateutil>=2.8.2 in /home/badhrinathan/.local/lib/python3.9/site-packages (from pandas>=0.25->seaborn) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/lib/python3/dist-packages (from pandas>=0.25->seaborn) (2021.1)
Requirement already satisfied: tzdata>=2022.1 in /home/badhrinathan/.local/lib/python3.9/site-packages (from pandas>=0.25->seaborn) (2023.3)
Requirement already satisfied: six>=1.5 in /usr/lib/python3/dist-packages (from python-dateutil>=2.8.2->pandas>=0.25->seaborn) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [24]:
```
import matplotlib.pyplot as plt
```

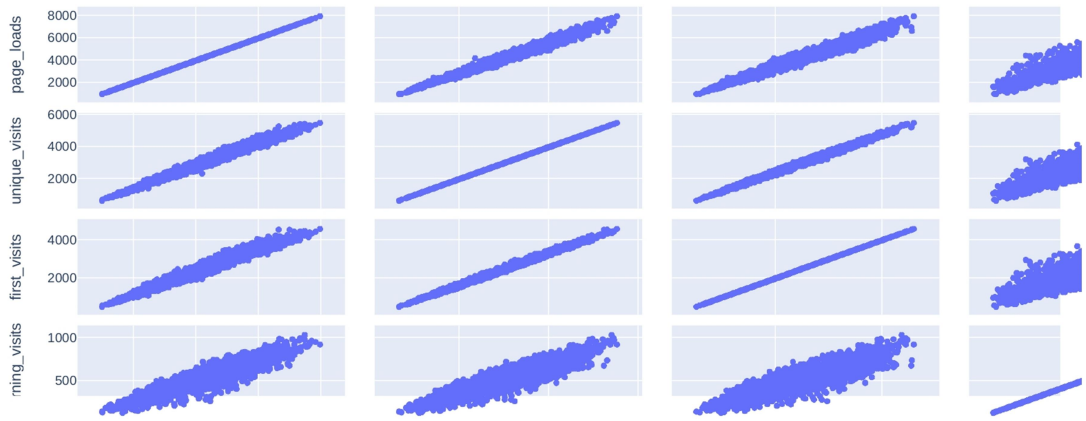
In [25]:
```
import seaborn as sns
```

In [26]:
```
fig, ax = plt.subplots()
fig.set_size_inches(8, 6)
sns.heatmap(df[['page_loads' ,'unique_visits' ,'first_visits' ,'returning_visits']].corr(),
            annot=True,
            cmap='viridis_r',
            fmt='g')
```

Out[26]:
```
<AxesSubplot:>
```



In [8]:
```
px.scatter_matrix(df[['page_loads' ,'unique_visits' ,'first_visits' ,'returning_visits']])
```



In [14]:
```
import plotly.express as px
import numpy as np
```

In [15]:
```
df['days_f']=np.where((df['Day']=='Tuesday') |
                      (df['Day']=='Wednesday') |
                      (df['Day']=='Thursday') |
                      (df['Day']=='Monday'),1,0)
df
```

Out[15]:

	Row	Day	day_of_week	Date	page_loads	unique_visits	first_visits	returning_visits	days_f
0	1	Sunday	1	9/14/2014	2146	1582	1430	152	0
1	2	Monday	2	9/15/2014	3621	2528	2297	231	1
2	3	Tuesday	3	9/16/2014	3698	2630	2352	278	1
3	4	Wednesday	4	9/17/2014	3667	2614	2327	287	1
4	5	Thursday	5	9/18/2014	3316	2366	2130	236	1
...	...	...	...	...	...	...	...	...	...
2162	2163	Saturday	7	8/15/2020	2221	1696	1373	323	0
2163	2164	Sunday	1	8/16/2020	2724	2037	1686	351	0
2164	2165	Monday	2	8/17/2020	3456	2638	2181	457	1
2165	2166	Tuesday	3	8/18/2020	3581	2683	2184	499	1
2166	2167	Wednesday	4	8/19/2020	2064	1564	1297	267	1

2167 rows × 9 columns

In [28]:
```
X2=df[['page_loads','first_visits' ,'returning_visits','days_f']]
y2=df['unique_visits']
```

In [59]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X2, y2, test_size=0.2)
```

In [60]:
```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

Out[60]:
```
▾ LinearRegression
LinearRegression()
```

In [61]:
```
regressor.coef_
```

Out[61]:
```
array([-3.88376916e-15,  1.00000000e+00,  1.00000000e+00, -3.99733415e-14])
```

In [62]:
```
regressor.intercept_
```

Out[62]: 1.4097167877480388e-11

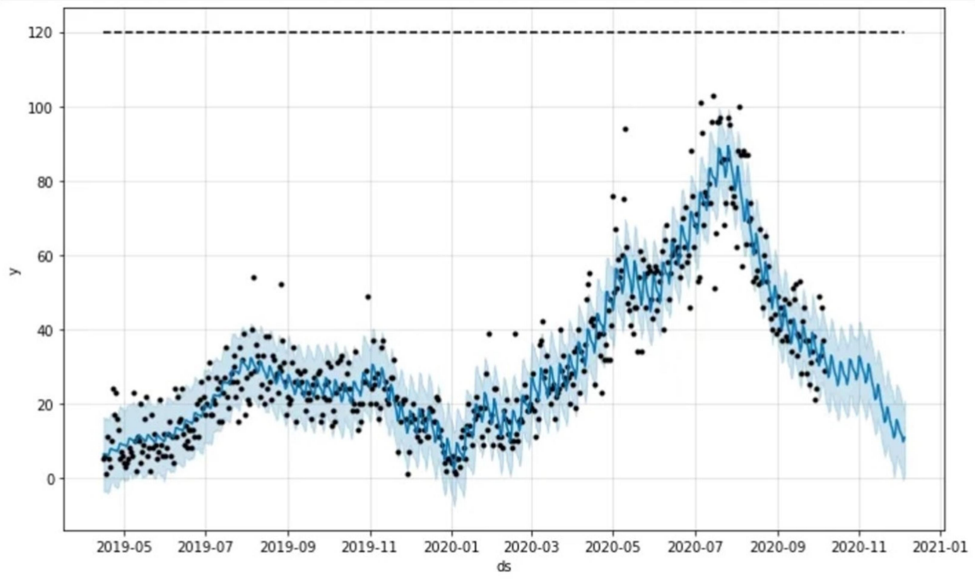
In [63]:
```
y_pred=regressor.predict(X_test)
X_pred=regressor.predict(X_train)
```

In [64]:
```
y_pred
```

Out[64]:
```
array([1011., 4340., 3552., 2293., 3783., 2919., 2031., 3505., 3621.,
       4823., 4474., 2857., 4382., 3493., 4399., 2818., 2238., 4322.,
       4724., 1981., 2325., 2702., 3830., 1999., 4674., 2622., 2898.,
       2604., 2566., 2856., 3767., 4345., 3611., 1281., 2477., 2442.,
       3008., 2621., 4866., 2681., 1791., 3075., 1658., 1928., 2591.,
       1864., 3019., 2643., 2685., 2664., 3925.,  948., 1005., 3215.,
       3194., 3517., 3441., 2397., 2269., 3085., 1981., 3557., 3838.,
       3457., 4960., 3287., 3808., 3689., 2496., 4119., 3469., 1976.,
       2870., 3996., 1407., 3711., 2913., 1177., 3178., 3784., 1502.,
       2199., 3770., 1609., 2658.,  876., 2274., 3604.,  923., 3247.,
       1312., 3601., 2912., 2394., 1376., 3967., 4975., 1223., 4102.,
       3331., 2704., 3780., 2816., 2892., 1977., 3838., 2277., 1577.,
       4520., 2850., 3064., 2366., 4176., 4159., 3827., 1755., 3381.,
       1054., 1567., 2211., 1947., 3011., 4216., 3002., 2508., 3021.,
       2805., 3244., 2860., 3219., 3904., 3236., 2153., 3345., 3793.,
       3812., 2800., 1788., 3372., 2410., 1099., 2979., 3119., 2906.,
       2169., 1777., 1155., 3787., 1015., 2743., 2515., 2145., 1564.,
        930., 3387., 4962., 4983., 3535., 1423., 1319., 1690., 4968.,
       1987., 2026., 1834., 3877., 2520., 3447., 3766., 2568., 1783.,
       2476., 3681., 2640., 3279., 2760., 4377., 3232., 3406., 3761.,
       3194., 2438., 2952., 3613., 5181., 3908., 2530., 2618., 3266.,
       2907., 2078., 4019., 2423., 3324., 3500., 2062., 3277., 3474.,
       4092., 4105., 4132., 2598., 1884., 4547., 3768., 4776., 3932.,
       2208., 3350., 2122.,  667., 2346., 3946., 4672., 1307., 3499.,
       2808., 2143., 1664., 2457., 3665., 2209., 2784., 3454., 2770.,
       3635., 2049., 1876., 1548., 2595., 1319., 4331., 1911., 1410.,
       3047., 2364., 4596., 3791., 4086., 4512., 2299., 4207., 4735.,
       1358., 3444., 3486., 4160., 4184., 3966., 4115., 1380., 3951.,
       2140., 2111., 2127., 2620., 1957., 1168., 3039., 3760., 4226.,
       2265., 3369., 3147., 2239., 1112., 3297., 3189., 1316., 2614.,
       1276., 3943., 3139., 3894., 3007., 2207., 1478., 2954., 1447.,
       2268., 1823., 4073., 3430., 1538., 2015., 2468., 2613., 2351.,
       3145., 4016., 2783., 2518., 2002., 3964., 1902., 2422., 1039.,
       1765., 2911., 2628., 2087., 3533., 4533., 2032., 2853., 2704.,
       2289., 2947., 2791., 1678., 2381., 4006.,  934., 2540., 4020.,
       2724., 3164., 3035., 2539., 1626., 1790., 4031., 3346., 3056.,
       3307., 2281., 4240., 2804., 2231., 3339., 1993., 2838., 2084.,
       1618., 4092., 1649., 3259., 3144., 1192., 2860., 2750., 4127.,
       3682., 3707., 3668., 2769., 3804., 4202., 4134., 4542., 4038.,
       1721., 3988., 2962., 3493., 3241., 5124., 2579., 2827., 1710.,
       3364., 2264., 1913., 2275., 1677., 4112., 1472., 2864., 2715.,
       3437., 3500., 2716., 2976., 4086., 3284., 4917., 2400., 3733.,
       3348., 2625., 2105., 1517., 3357., 3245., 3593., 2762., 2291.,
       1901., 3390., 4520., 3265., 2881., 3321., 1370., 3307., 3597.,
       1277., 3669., 3309., 4266., 2970., 2971., 3762., 3409., 2825.,
       1430., 1211., 4042., 3003., 3388., 3175., 3703., 4106., 4674.,
```

Out[64]:
```
array([1011., 4340., 3552., 2293., 3783., 2919., 2031., 3505., 3621.,
       4823., 4474., 2857., 4382., 3493., 4399., 2818., 2238., 4322.,
       4724., 1981., 2325., 2702., 3830., 1999., 4674., 2622., 2898.,
       2604., 2566., 2856., 3767., 4345., 3611., 1281., 2477., 2442.,
       3008., 2621., 4866., 2681., 1791., 3075., 1658., 1928., 2591.,
       1864., 3019., 2643., 2685., 2664., 3925.,  948., 1005., 3215.,
       3194., 3517., 3441., 2397., 2269., 3085., 1981., 3557., 3838.,
       3457., 4960., 3287., 3808., 3689., 2496., 4119., 3469., 1976.,
       2870., 3996., 1407., 3711., 2913., 1177., 3178., 3784., 1502.,
       2199., 3770., 1609., 2658.,  876., 2274., 3604.,  923., 3247.,
       1312., 3601., 2912., 2394., 1376., 3967., 4975., 1223., 4102.,
       3331., 2704., 3780., 2816., 2892., 1977., 3838., 2277., 1577.,
       4520., 2850., 3064., 2366., 4176., 4159., 3827., 1755., 3381.,
       1054., 1567., 2211., 1947., 3011., 4216., 3002., 2508., 3021.,
       2805., 3244., 2860., 3219., 3904., 3236., 2153., 3345., 3793.,
       3812., 2800., 1788., 3372., 2410., 1099., 2979., 3119., 2906.,
       2169., 1777., 1155., 3787., 1015., 2743., 2515., 2145., 1564.,
        930., 3387., 4962., 4983., 3535., 1423., 1319., 1690., 4968.,
       1987., 2026., 1834., 3877., 2520., 3447., 3766., 2568., 1783.,
       2476., 3681., 2640., 3279., 2760., 4377., 3232., 3406., 3761.,
       3194., 2438., 2952., 3613., 5181., 3908., 2530., 2618., 3266.,
       2907., 2078., 4019., 2423., 3324., 3500., 2062., 3277., 3474.,
       4092., 4105., 4132., 2598., 1884., 4547., 3768., 4776., 3932.,
       2208., 3350., 2122.,  667., 2346., 3946., 4672., 1307., 3499.,
       2808., 2143., 1664., 2457., 3665., 2209., 2784., 3454., 2770.,
       3635., 2049., 1876., 1548., 2595., 1319., 4331., 1911., 1410.,
       3047., 2364., 4596., 3791., 4086., 4512., 2299., 4207., 4735.,
       1358., 3444., 3486., 4160., 4184., 3966., 4115., 1380., 3951.,
       2140., 2111., 2127., 2620., 1957., 1168., 3039., 3760., 4226.,
       2265., 3369., 3147., 2239., 1112., 3297., 3189., 1316., 2614.,
       1276., 3943., 3139., 3894., 3007., 2207., 1478., 2954., 1447.,
       2268., 1823., 4073., 3430., 1538., 2015., 2468., 2613., 2351.,
       3145., 4016., 2783., 2518., 2002., 3964., 1902., 2422., 1039.,
       1765., 2911., 2628., 2087., 3533., 4533., 2032., 2853., 2704.,
       2289., 2947., 2791., 1678., 2381., 4006.,  934., 2540., 4020.,
       2724., 3164., 3035., 2539., 1626., 1790., 4031., 3346., 3056.,
       3307., 2281., 4240., 2804., 2231., 3339., 1993., 2838., 2084.,
       1618., 4092., 1649., 3259., 3144., 1192., 2860., 2750., 4127.,
       3682., 3707., 3668., 2769., 3804., 4202., 4134., 4542., 4038.,
       1721., 3988., 2962., 3493., 3241., 5124., 2579., 2827., 1710.,
       3364., 2264., 1913., 2275., 1677., 4112., 1472., 2864., 2715.,
       3437., 3500., 2716., 2976., 4086., 3284., 4917., 2400., 3733.,
       3348., 2625., 2105., 1517., 3357., 3245., 3593., 2762., 2291.,
       1901., 3390., 4520., 3265., 2881., 3321., 1370., 3307., 3597.,
       1277., 3669., 3309., 4266., 2970., 2971., 3762., 3409., 2825.,
       1430., 1211., 4042., 3003., 3388., 3175., 3703., 4106., 4674.,
```

Out[75]:
```
array([1011., 4340., 3552., 2293., 3783., 2919., 2031., 3505., 3621.,
       4823., 4474., 2857., 4382., 3493., 4399., 2818., 2238., 4322.,
       4724., 1981., 2325., 2702., 3830., 1999., 4674., 2622., 2898.,
       2604., 2566., 2856., 3767., 4345., 3611., 1281., 2477., 2442.,
       3008., 2621., 4866., 2681., 1791., 3075., 1658., 1928., 2591.,
       1864., 3019., 2643., 2685., 2664., 3925.,  948., 1005., 3215.,
       3194., 3517., 3441., 2397., 2269., 3085., 1981., 3557., 3838.,
       3457., 4960., 3287., 3808., 3689., 2496., 4119., 3469., 1976.,
       2870., 3996., 1407., 3711., 2913., 1177., 3178., 3784., 1502.,
       2199., 3770., 1609., 2658.,  876., 2274., 3604.,  923., 3247.,
       1312., 3601., 2912., 2394., 1376., 3967., 4975., 1223., 4102.,
       3331., 2704., 3780., 2816., 2892., 1977., 3838., 2277., 1577.,
       4520., 2850., 3064., 2366., 4176., 4159., 3827., 1755., 3381.,
       1054., 1567., 2211., 1947., 3011., 4216., 3002., 2508., 3021.,
       2805., 3244., 2860., 3219., 3904., 3236., 2153., 3345., 3793.,
       3812., 2800., 1788., 3372., 2410., 1099., 2979., 3119., 2906.,
       2169., 1777., 1155., 3787., 1015., 2743., 2515., 2145., 1564.,
        930., 3387., 4962., 4983., 3535., 1423., 1319., 1690., 4968.,
       1987., 2026., 1834., 3877., 2520., 3447., 3766., 2568., 1783.,
       2476., 3681., 2640., 3279., 2760., 4377., 3232., 3406., 3761.,
       3194., 2438., 2952., 3613., 5181., 3908., 2530., 2618., 3266.,
       2907., 2078., 4019., 2423., 3324., 3500., 2062., 3277., 3474.,
       4092., 4105., 4132., 2598., 1884., 4547., 3768., 4776., 3932.,
       2208., 3350., 2122.,  667., 2346., 3946., 4672., 1307., 3499.,
       2808., 2143., 1664., 2457., 3665., 2209., 2784., 3454., 2770.,
       3635., 2049., 1876., 1548., 2595., 1319., 4331., 1911., 1410.,
       3047., 2364., 4596., 3791., 4086., 4512., 2299., 4207., 4735.,
       1358., 3444., 3486., 4160., 4184., 3966., 4115., 1380., 3951.,
       2140., 2111., 2127., 2620., 1957., 1168., 3039., 3760., 4226.,
       2265., 3369., 3147., 2239., 1112., 3297., 3189., 1316., 2614.,
       1276., 3943., 3139., 3894., 3007., 2207., 1478., 2954., 1447.,
       2268., 1823., 4073., 3430., 1538., 2015., 2468., 2613., 2351.,
       3145., 4016., 2783., 2518., 2002., 3964., 1902., 2422., 1039.,
       1765., 2911., 2628., 2087., 3533., 4533., 2032., 2853., 2704.,
       2289., 2947., 2791., 1678., 2381., 4006.,  934., 2540., 4020.,
       2724., 3164., 3035., 2539., 1626., 1790., 4031., 3346., 3056.,
       3307., 2281., 4240., 2804., 2231., 3339., 1993., 2838., 2084.,
       1618., 4092., 1649., 3259., 3144., 1192., 2860., 2750., 4127.,
       3682., 3707., 3668., 2769., 3804., 4202., 4134., 4542., 4038.,
       1721., 3988., 2962., 3493., 3241., 5124., 2579., 2827., 1710.,
       3364., 2264., 1913., 2275., 1677., 4112., 1472., 2864., 2715.,
       3437., 3500., 2716., 2976., 4086., 3284., 4917., 2400., 3733.,
       3348., 2625., 2105., 1517., 3357., 3245., 3593., 2762., 2291.,
       1901., 3390., 4520., 3265., 2881., 3321., 1370., 3307., 3597.,
       1277., 3669., 3309., 4266., 2970., 2971., 3762., 3409., 2825.,
       1430., 1211., 4042., 3003., 3388., 3175., 3703., 4106., 4674.,
```



In [80]:
```
from sklearn.metrics import r2_score
r2_score(y_test,y_pred)
```

Out[80]: 0.7876426925

In the above asccuracy score was

0.7876426925

Insights to Improve User Experience:

The insights from the website traffic analysis dashboard can help website owners improve user experience in the following ways:

- Identify the most popular pages on the website and make sure that they are easy to find and navigate.
- Understand the user journey through the website and remove any obstacles that may prevent users from completing their desired tasks.
- Identify the sources of traffic to the website and focus marketing efforts on the most effective channels.
- Track the performance of different marketing campaigns and identify the ones that are most effective in driving traffic to the website.
- Measure the impact of website changes on user behavior and make adjustments as needed.

Overall, the website traffic analysis dashboard provides website owners with a comprehensive and easy-to-use tool to analyze their website traffic and gain insights into user behavior and content performance. This information can be used to improve the user experience, increase engagement, and boost conversions.

Conclusion

The website traffic analysis dashboard project was a success. The dashboard was developed using IBM Cognos and Python code integration and provides website owners with insights into their website traffic patterns, user behavior, and content performance. This information can be used to improve the user experience, increase engagement, and boost conversions.