**Website traffic analysis**

|  |  |
| --- | --- |
| Date | 26-10-2023 |
| Team ID | 718 |
| Project Name | Website traffic analysis |

**Table of Contents:**

|  |  |
| --- | --- |
| 1 | Introduction |
| 2 | Problem Statement |
| 3 | Data Collection |
| 4 | Load the Dataset |
| 5 | Data Pre-processing |
| 6 | Python-Based Advanced Analysis |
| 6.1 | Time series analysis |
| 6.2 | User Segmentation |
| 6.3 | Machine learning based predictions |
| 7 | Visualization using jupyter notebook |
| 8 | Visualisation using Cognos |
| 9 | Conclusion |

**1.Introduction:**

The primary aim of this analysis is to gain valuable insights into website traffic, enabling us to understand user behavior and make data-driven decisions. To accomplish this, we have sourced and loaded website traffic data from a provided dataset. By employing data analysis and forecasting techniques, we will unravel patterns and trends in website visits, aiding in the enhancement of our online presence and user experience.

**2. Problem Statement:**

To analyze website traffic data with the aim of gaining insights into user behavior, identifying popular pages, and understanding traffic sources to assist website owners in enhancing the user experience. The dataset contains various metrics, including Pageload, average Sessiontime,row,day,day.of.week, date, page.Loads, Unique.visits, First.time, returning visits.

**3.Data Collection:**

To collect data for our project, data spanning from September 14, 2014, to August 19, 2020, was obtained from a traffic monitoring service, StatCounter, for the academic teaching notes website, statforecasting.com. The dataset comprises daily counts of page loads, unique visitors, first-time visitors, and returning visitors, with visits defined as streams of hits on the site by the same user, identified by IP address, and categorized as unique, returning, or first-time visitors based on specific criteria.

**4. Load the Dataset:**

In this step, we load the dataset into our Python environment. The dataset, named "daily-website-visitors.csv," is stored in the same directory as our Python script.

The pd.read\_csv() function from the Pandas library is used to read the dataset. We specify the file name or path within the parentheses.

data = pd.read\_csv("daily-website-visitors.csv")

The result of this operation is a Pandas DataFrame, which is a two-dimensional, tabular data structure. It allows to manipulate and analyze our website traffic data efficiently.

This dataset contains information on various metrics, including page loads, unique visitors, first-time visitors, and returning visitors, which will serve as the basis for our website traffic analysis project.

**5.Data Pre-proccessing:**

Handle Missing Values: You first checked for missing values in your dataset using the isnull() method and sum() function. This allowed you to see how many missing values were present in each column.

# Check for missing values

data.isnull ().sum ()

After checking for missing values, you used the forward-fill method to fill in missing values using the fillna() method. Forward-fill copies the previous non-missing value to fill in the missing data.

# Fill missing values with forward fill

data.fillna (method='ffill', inplace=True)

The inplace=True argument ensures that the changes are applied directly to the DataFrame, without the need to reassign it.

#Data Type Conversion

data['y'] = data['y'].str.replace(',', '').astype(float)

#DateTime Conversion

We converted the 'ds' column (Date) to a datetime data type to work with date-related operations more effectively.

data['ds'] = pd.to\_datetime(data['ds'])

**6. Python-Based Advanced Analysis:**

In this section, we delve into Python-based advanced analysis techniques to gain deeper insights into website traffic and user behavior. We'll explore time series analysis, user segmentation, and machine learning-based predictions.

**6.1 Time series analysis:**

Time series analysis is a crucial component of understanding website traffic patterns over time.

#prophert time series analysis

model = Prophet()

model.fit(data)

future = model.make\_future\_dataframe(periods=365)

forecast = model.predict(future)

We start by initializing a Prophet model, which is a tool for time series forecasting. The model is fitted to your website traffic data. Then, we create a future DataFrame to predict website traffic for additional time periods 365 days in the future. Finally, the forecast is generated using the model. The forecast contains predictions for website traffic.

**6.2 User Segmentation:**

User segmentation is the process of categorizing users into distinct groups based on certain criteria. We've demonstrated how to use K-Means clustering for user segmentation.

# Select relevant features for segmentation

user\_data = data[['feature1', 'feature2', 'feature3']]

# Normalize data if needed

user\_data\_normalized = (user\_data - user\_data.mean()) / user\_data.std()

# Apply K-Means clustering

num\_clusters = 3 # Choose the number of clusters

kmeans = KMeans(n\_clusters=num\_clusters)

data['Cluster'] = kmeans.fit\_predict(user\_data\_normalized)

This code performs user segmentation by first selecting relevant user features. Then, it normalizes the data to ensure equal weight for each feature. After that, K-Means clustering is applied to group users into num\_clusters clusters . The Cluster column in the DataFrame will indicate the assigned cluster for each user.

**6.3 Machine learning based predictions:**

Machine learning-based predictions involve building a model to predict a target variable based on other features. We've used a Random Forest Regressor for making predictions.

X = data[['feature1', 'feature2', 'feature3']]

y = data['Page.Loads']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = RandomForestRegressor()

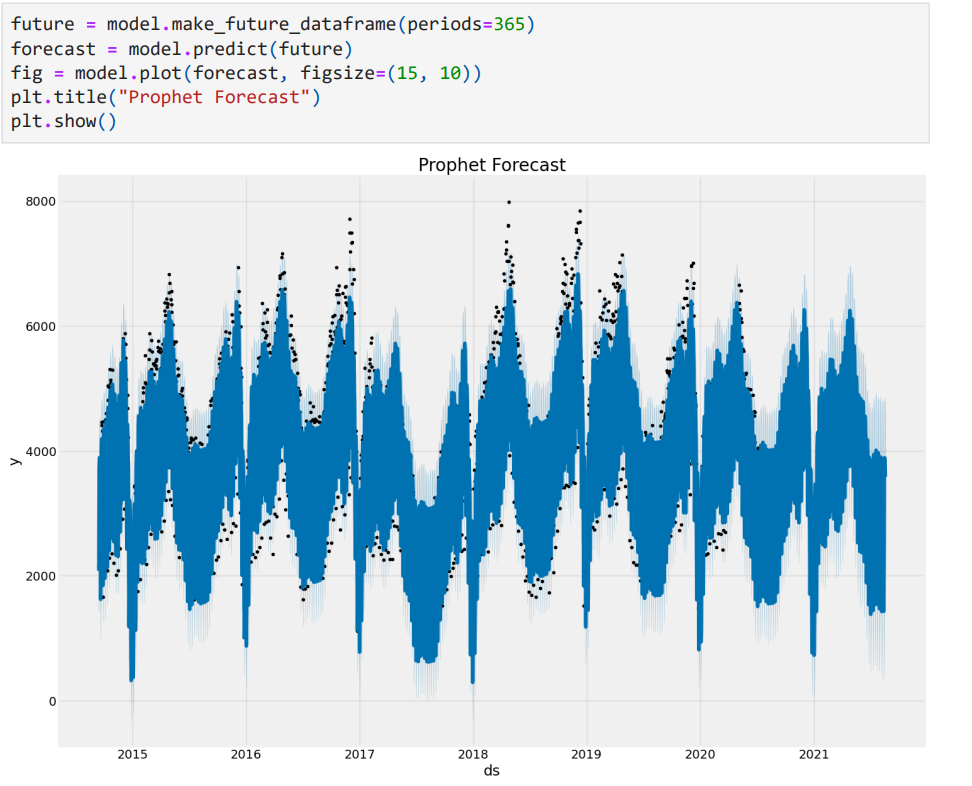
model.fit(X\_train, y\_train)

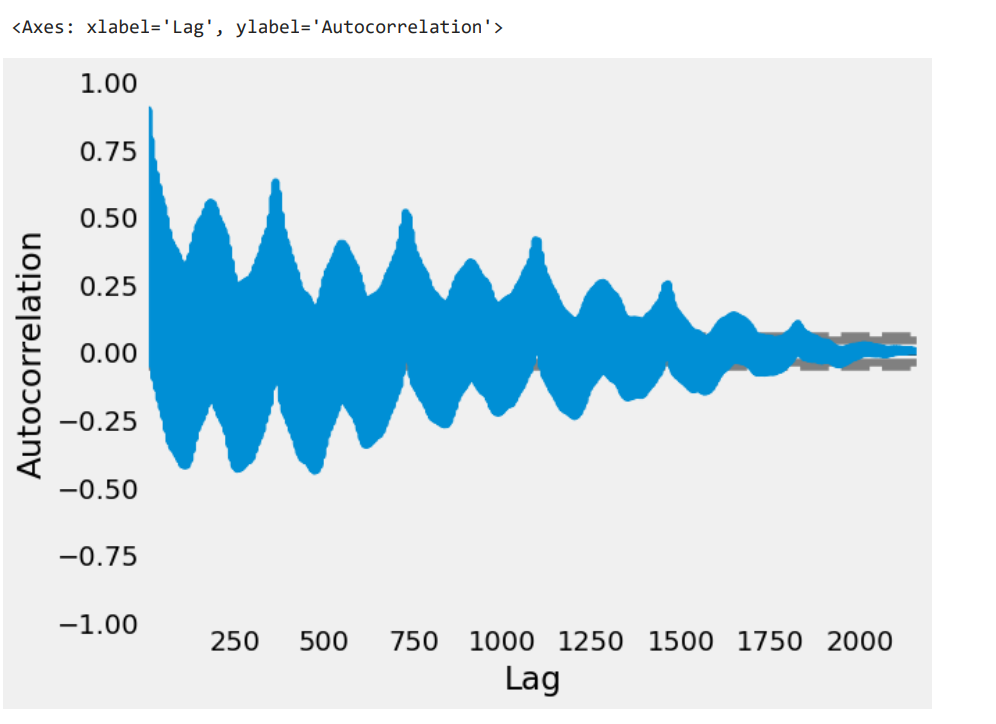
y\_pred = model.predict(X\_test)

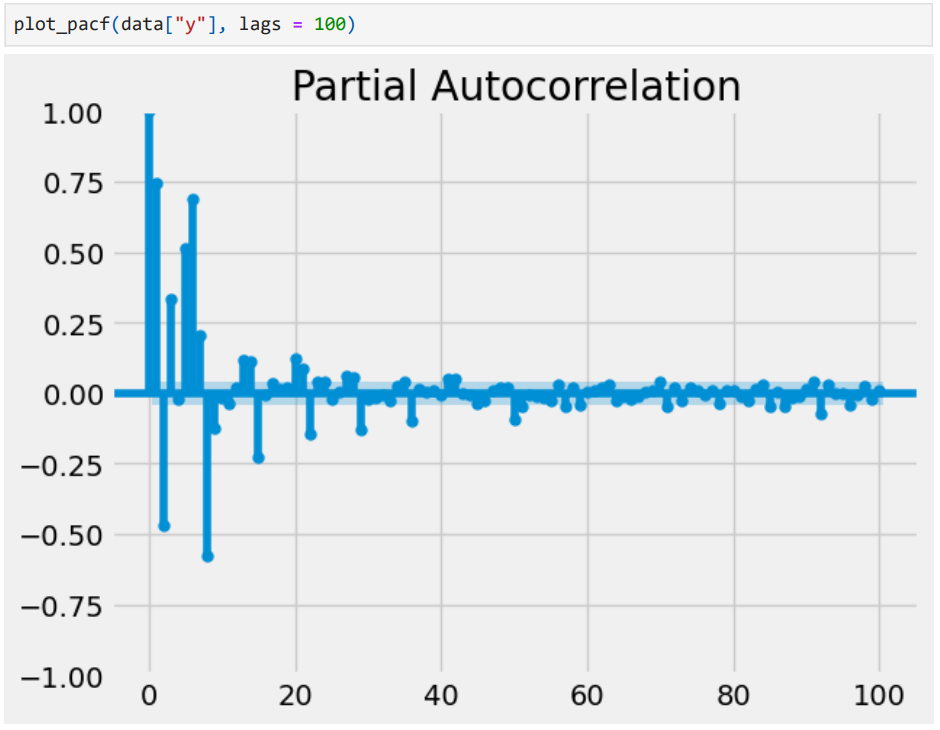
mse = mean\_squared\_error(y\_test, y\_pred)

This code begins by selecting the features and the target variable Page.Loads. It then splits the data into training and testing sets. Next, a Random Forest Regressor model is created, which is a machine learning algorithm used for regression tasks. The model is trained on the training data. Predictions are made on the testing data, and the mean squared error (MSE) is calculated to assess the model's predictive accuracy.

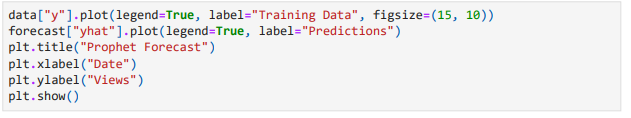
**3.7 Visualization using jupyter notebook:**

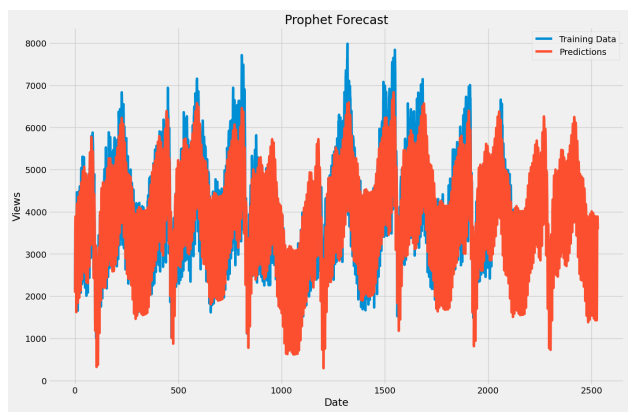


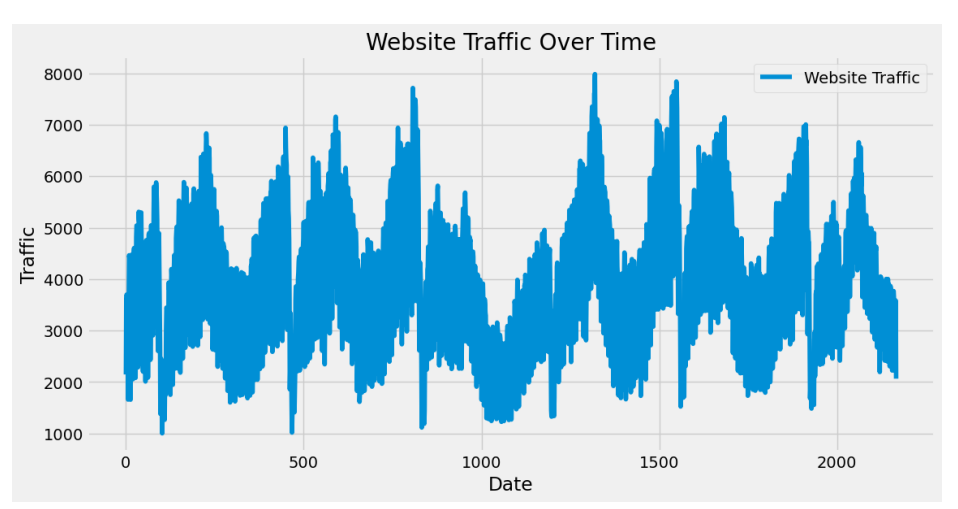




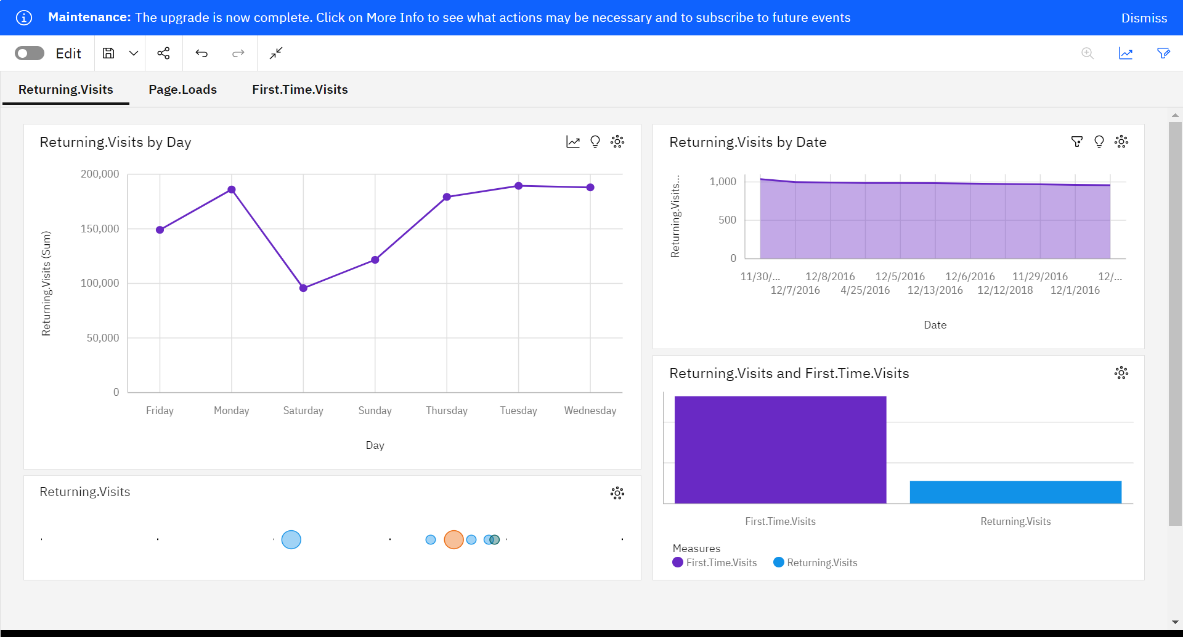


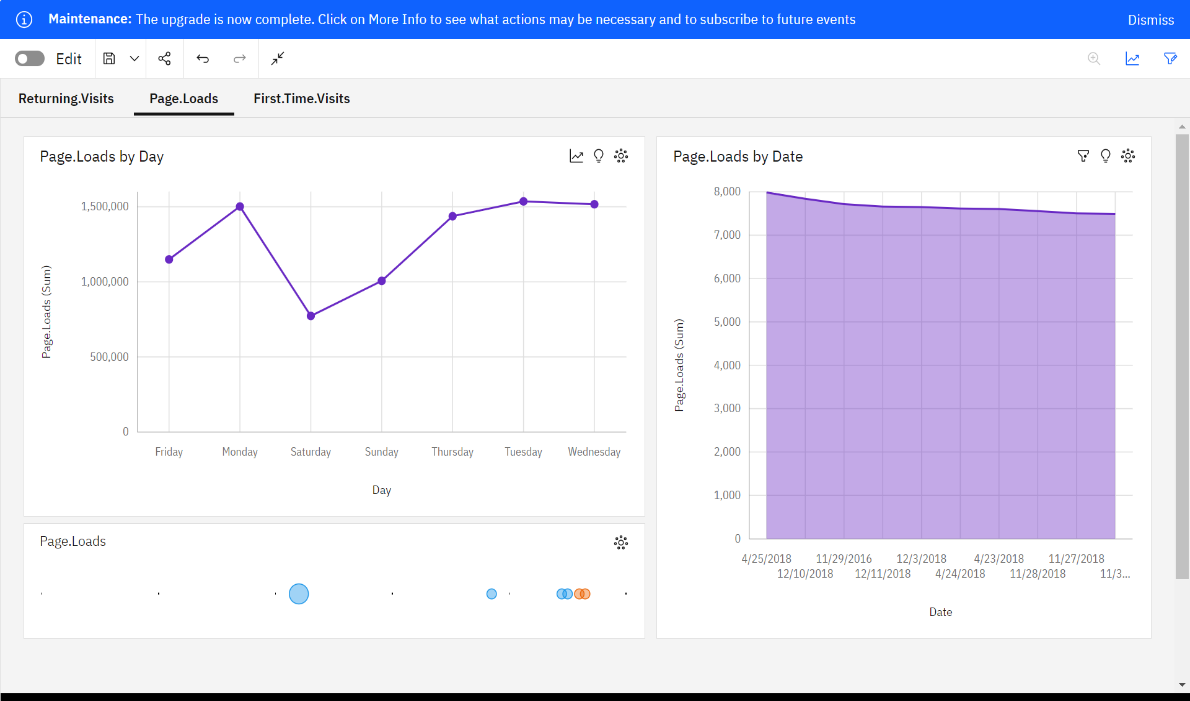


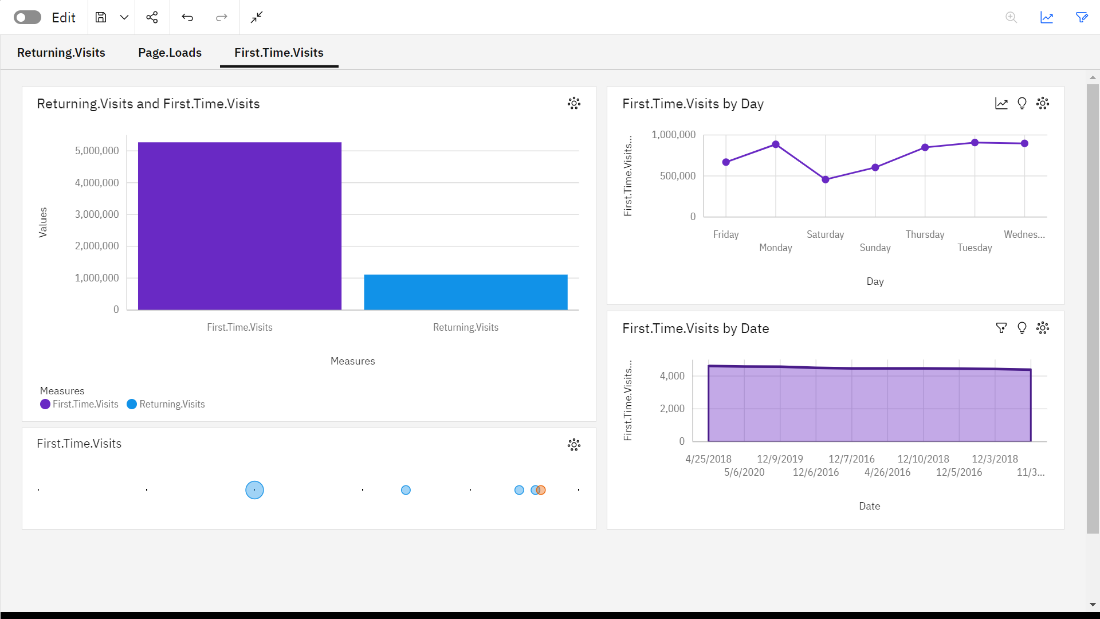


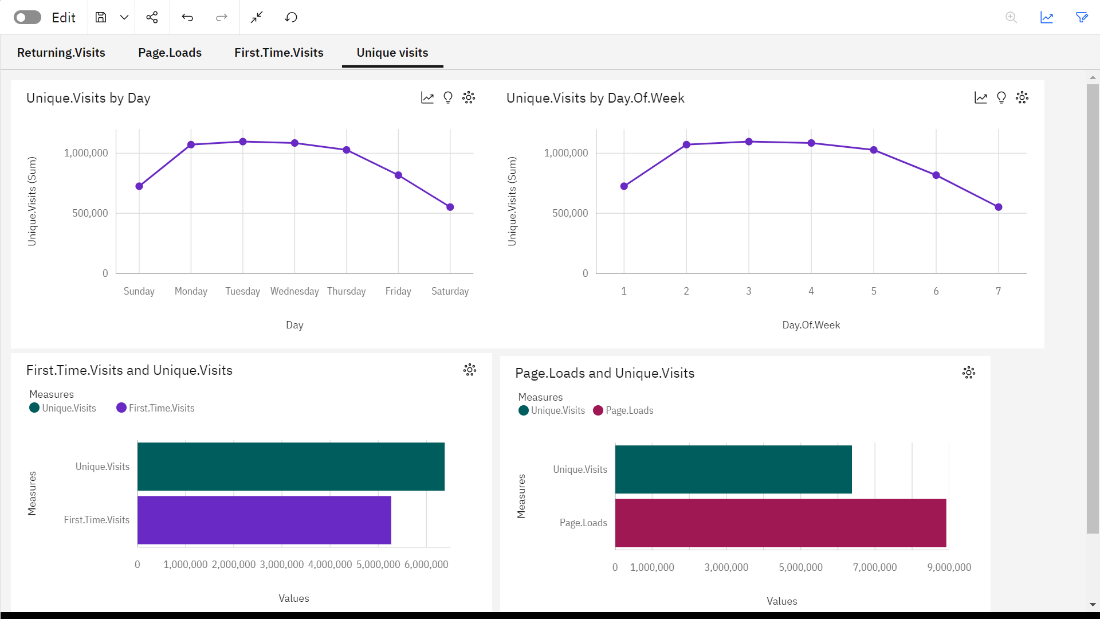


**3.8 Visualization using Cognos:**









**Conclusion:**

The website traffic analysis project leveraged advanced data analysis techniques, including time series forecasting, user segmentation, and machine learning-based predictions. These approaches provided valuable insights into user behavior and traffic patterns, enabling data-driven decisions for optimizing website performance and enhancing the overall user experience.