**Website traffic analysis**

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

**Table of Contents:**

|  |  |
| --- | --- |
| 1 | Introduction |
| 2 | Problem Statement |
| 3 | Steps Involved in Model Evaluation: |
| 3.1 | Data Collection |
| 3.2 | Load the Dataset |
| 3.3 | Explore the Dataset |
| 3.4 | Data Pre-processing |
| 3.5 | Predictive Model training |
| 3.7 | Visualization using jupyter notebook |
| 3.8 | Visualisation using Cognos |
| 4 | 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. Steps Involved in Model Evaluation:**

**3.1. 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.

**3.2. 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.

**3.3. Explore the Dataset:**

Before diving into data preprocessing, it's important to understand your dataset. You can use various Pandas functions to explore it:

|  |  |
| --- | --- |
| data.head (): | We used this function to display the first few rows of the dataset. It provided us with an initial view of the data's layout. |
| data.describe (): | It provides basic statistical information about your data, including measures like mean, standard deviation, and quartiles for numerical columns. |
| data.columns: | we obtained a list of column names in the dataset, helping us identify the variables available for analysis. |
| data.isnull ().sum (): | This function allowed us to check for missing values within each column. It provided us with a count of how many missing values were present in each feature, helping us assess data completeness and potential data cleaning needs. |

**3.4. 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'])

**Data Transformation:**

Our dataset contains date or time columns,so we convert them to the datetime data type for time-based analysis.

# Convert a date column to datetime

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

**Data Cleaning:**

Data cleaning is a critical step to ensure the quality and reliability of the dataset. In this section, we perform the following data cleaning and transformation tasks

# Clean the 'y' Column

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

**3.5. Predictive Model training:**

In this section, you are using the Facebook Prophet library to build a predictive model for forecasting website traffic

# Create a Prophet model and fit it to the data

model = Prophet()

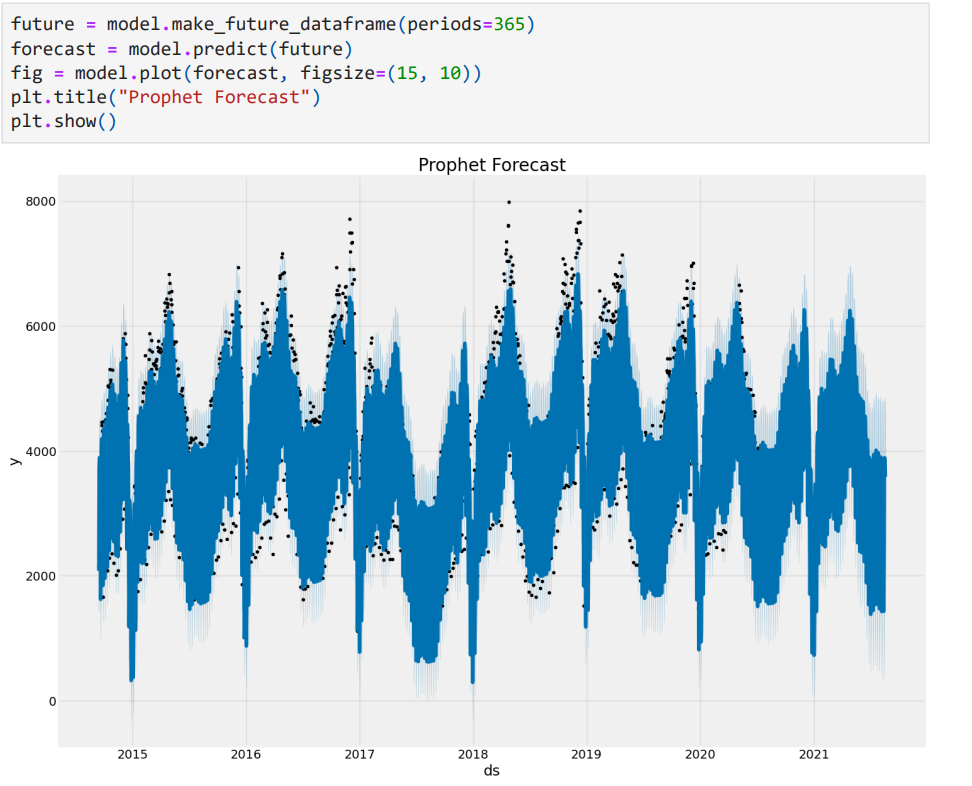
model.fit(data)

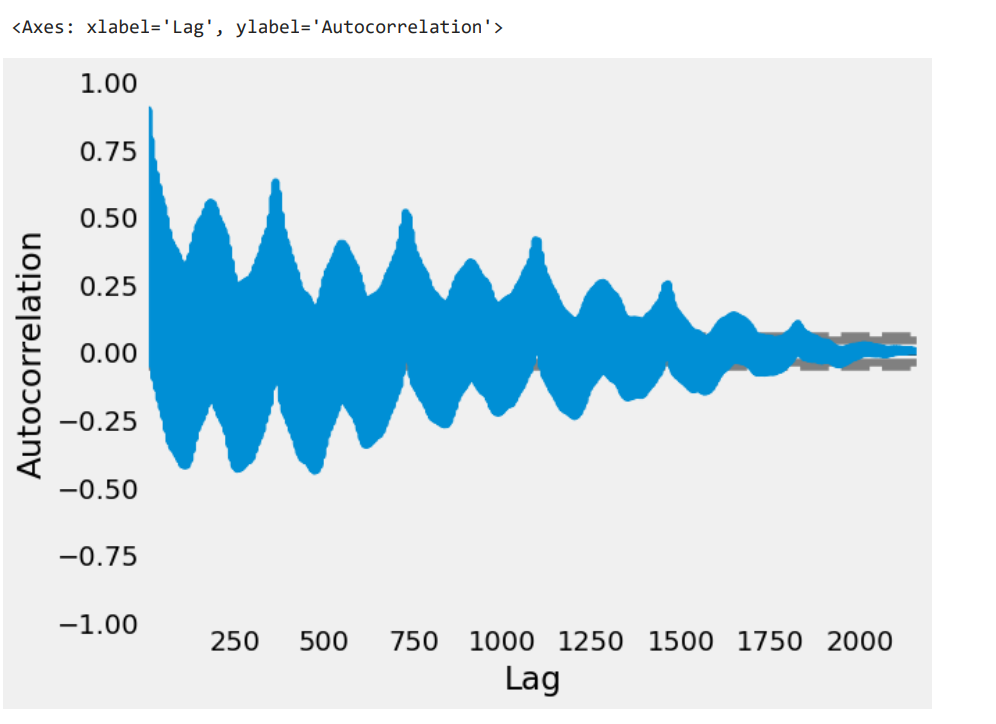
# Make future data points for forecasting

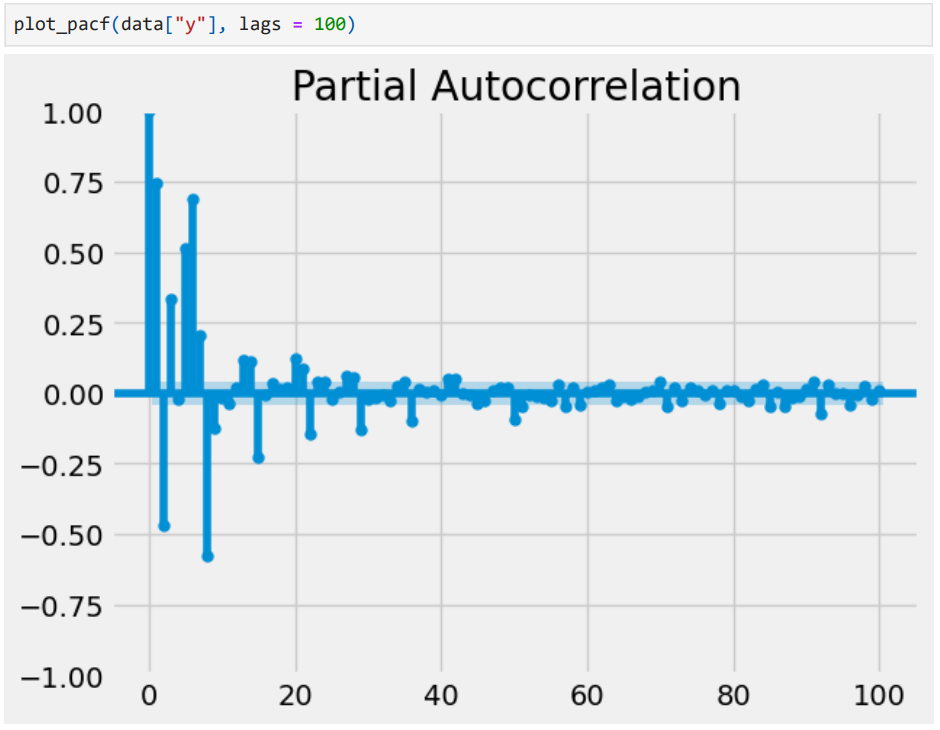
future = model.make\_future\_dataframe(periods=365)

# Generate a forecast forecast = model.predict(future)

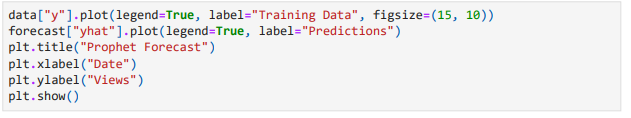
**3.7 Visualization using jupyter notebook**

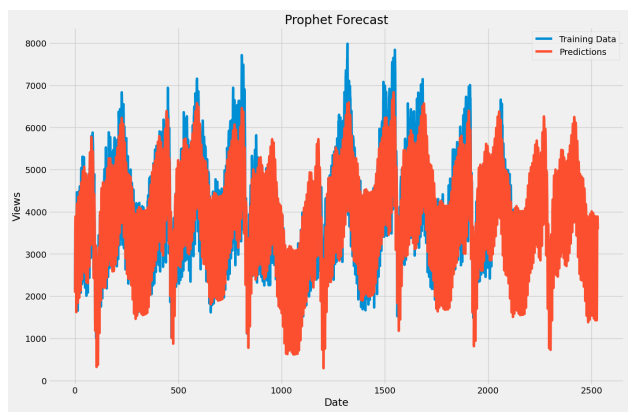




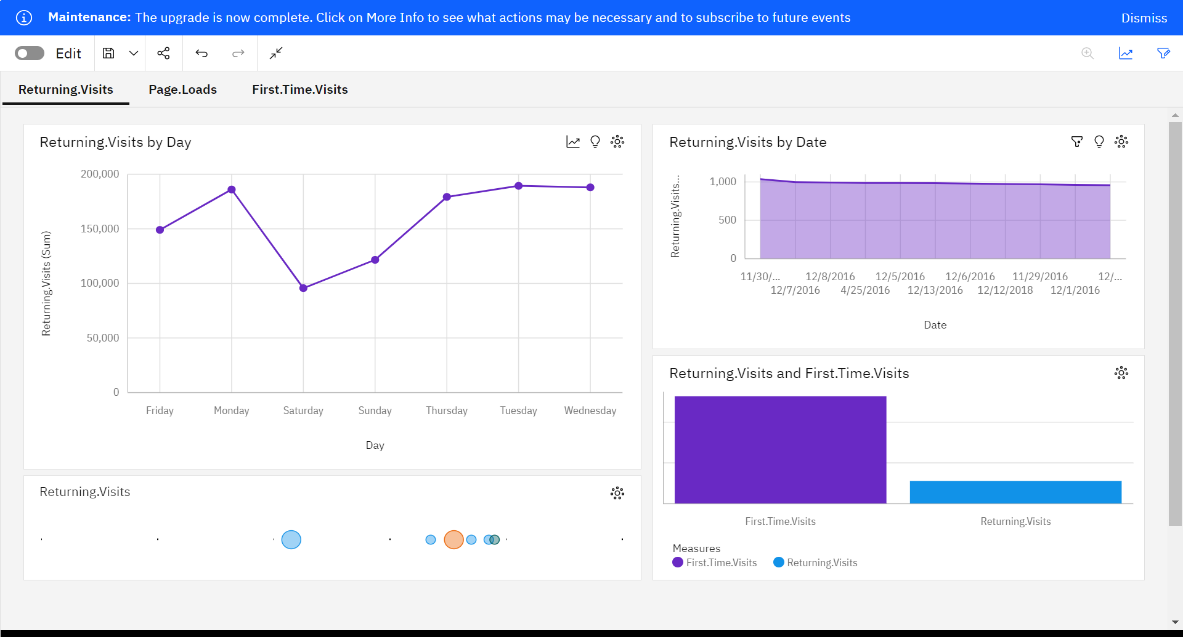


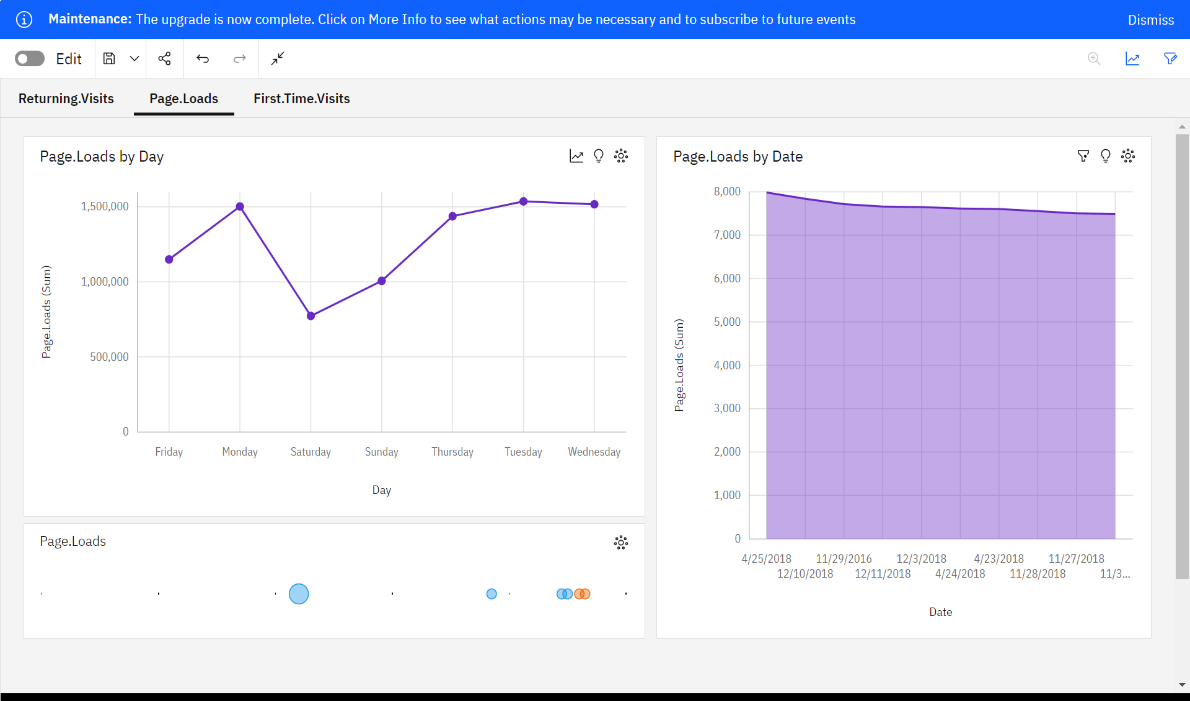


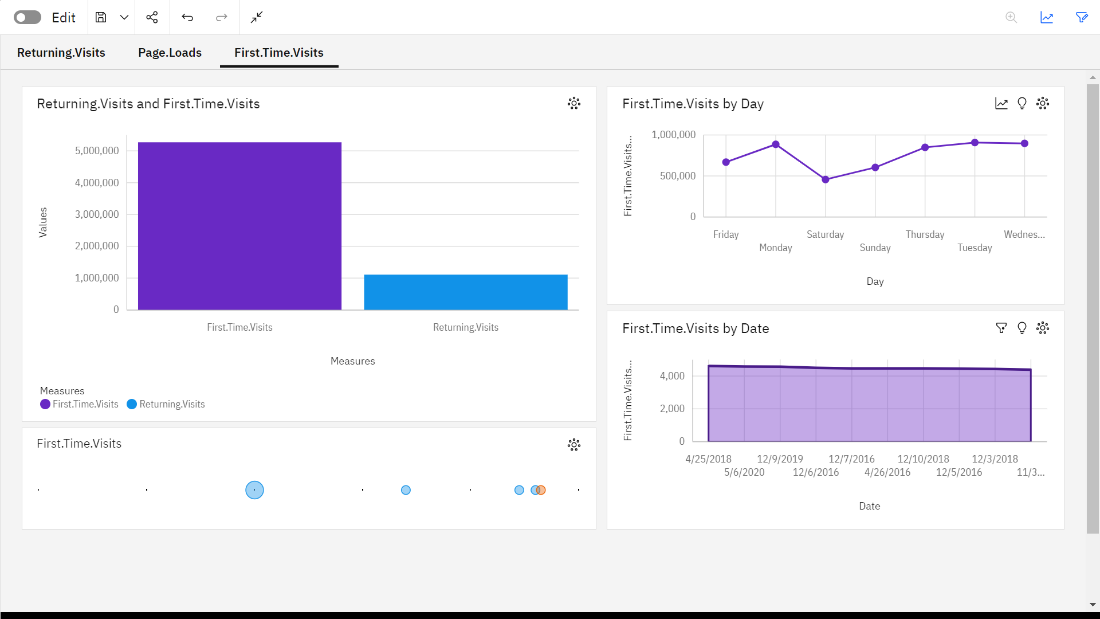




**3.8 Visualization using Cognos:**







**Conclusion:**

Through the application of time series analysis techniques and the use of the Prophet and SARIMA models, we have successfully conducted a comprehensive analysis of website traffic. By processing and exploring the dataset, we prepared it for modeling. The Prophet model, designed for daily observations, provided accurate forecasts of website traffic, and the SARIMA model offered an alternative approach to time series forecasting. These results equip us with valuable insights and predictions, empowering us to make informed decisions for optimizing website performance and content delivery.