DATA ANALYTICS WITH COGNOS

WEBSITE TRAFFIC ANALYSIS – PHASE 5

INTRODUCTION

In Phase V of Website Traffic Analysis, project's objective, design thinking process, analysis objectives, data collection process, data visualization using IBM Cognos for website traffic analysis had been concluded and documented.

PROBLEM STATEMENT

To collect and analyze data on user traffic in websites by applying IBM Cognos analytics tool to obtain valuable data-driven insights on user preferences, understand user interaction and enhance user experience.

PROBLEM DEFINITION

This project involves analysing website traffic data to gain insights into user behaviour, popular pages, and traffic sources. The goal is to help website owners enhance the user experience by understanding how visitors interact with the site. This project encompasses defining the analysis objectives, collecting website traffic data, using IBM Cognos for data visualization, and integrating Python code for advanced analysis.

DESIGN THINKING

1.Analysis Objectives:

The objective of the project is to extract the following key insights based on the given website traffic dataset.

- **Bounce rate analysis:** To analyse the bounce rate for different pages. A high bounce rate may indicate that visitors are not finding what they're looking for or that there's a problem with the page's content or design.
- **Time-series analysis of traffic trends:** To examine traffic patterns over time, including daily, weekly, and seasonal fluctuations which helps in making business decisions on augmenting additional resources to the server to better handle user traffic.
- **Website traffic forecasting:** To predict future website visitor trends, volumes, and user behaviour based on historical data which helps in resource allocation and capacity planning.
- **Conversion Rate Analysis:** To analyse the behavior of users who return after an extended period can provide insights into their conversion rates compared to more active users. This information can inform conversion optimization strategies.
- **Popularity of the site:** To derive insights on the popularity of the site based on visitor behavior (inactive and returning visitors) to help improve content quality and overall user experience. This could also help tailor marketing and content strategies.
- **Identification of Technical Issues:** To identify and eliminate technical issues, such as slow page loading, which lead to prolonged inactivity thereby addressing performance issues.

2.Data Collection:

Generic website traffic analysis involves collection of user traffic data from various sources such as server logs, content management systems, content delivery networks, third party cookies, user surveys and feedback etc... The below dataset is the major source of data for the analysis phase. Given below is the overall description of the entities, attributes and tuples available in the dataset.

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

This file contains 5 years of daily time series data for several measures of traffic on a statistical forecasting teaching notes website. There are 2167 rows of data spanning the date range from September 14, 2014, to August 19, 2020. A visit is defined as a stream of hits on one or more pages on the site on a given day by the same user, as identified by IP address. The following are the attributes in the above mentioned dataset.

- **Row (Type : Integer)** : Denotes the row number which is used to uniquely identify a particular data entry.
- **Day (Type : String)** : Consists of the day of the week that pertains to that row.
- Day_of_week (Type : Integer) : Denotes the day of the week in numerical form (numbered from 1 through 7 for each day of the week in that respective order).
- **Date (Format : dd/mm/yyyy)** : Denotes the date of entry (The entire data ranges over a period of 5 years).
- **Page Loads (Type : Integer)** : No. of page loads handled by the web server on that particular day on a daily basis.
- **Unique visits (Type : Integer)** : Daily number of visitors from whose IP addresses there haven't been hits on any page in over 6 hours. A visit is classified as "unique" if a hit from the same IP address has not come within the last 6 hours.
- **First time visits (Type : Integer) :** Number of unique visitors who do not have a cookie identifying them as a previous customer. Returning visitors are identified by cookies if those are accepted.
- **Returning visits (Type : Integer)** : Number of unique visitors minus first time visitors

3.Visualisation of Collected Data:

Creating meaningful dashboards and reports using IBM Cognos for visualizing insights from website traffic data requires a well-structured plan. Here's a step-by-step plan to guide you through the process:

- **Objectives of Data Visualization :** To visualise user demographics, website popularity and traffic magnitude assessment metrics.
- **Data Preprocessing :** To clean , structure and integrate data into a format that can be easily accessed by IBM Cognos.
- **Design of the data model :** To design a data model that represents the website traffic data. This may involve defining data sources, relationships, and calculations.

- **Create data queries :** Use IBM Cognos Query Studio or Report Studio to create data queries that extract the specific metrics and dimensions required for the visualizations.
- **Develop visualizations :** Use IBM Cognos Report Studio or Dashboard Studio to create a variety of visualizations, including bar charts, line graphs, pie charts, tables and funnel charts etc… to display trends in page views, traffic sources, distribution of traffic sources or demographics and user engagement over time and to track conversion rates and the user journey.
- **Define Filters and Parameters and create dashboards:** Implement interactive filters and parameters in the dashboards and reports. This allows users to customize the data they see, facilitating deeper exploration. Dashboards provide a consolidated view of key insights and allow users to explore the data dynamically.

By following this plan, IBM Cognos can be leveraged to create meaningful dashboards and reports that enable website owners to gain actionable insights from their traffic data, make informed decisions, and enhance the user experience.

4.Python Integration for further analysis:

Incorporating machine learning models to predict future traffic trends or user behavior patterns can provide valuable insights for website optimization and business decision-making. Here are several possible ways to leverage machine learning in this context:

- **Time Series Forecasting :** Use time series forecasting models such as ARIMA, Exponential Smoothing, or Prophet to predict future website traffic trends based on historical data. This can help you anticipate traffic spikes, seasonality, and overall traffic patterns.
- **Regression Analysis :** Apply regression models to predict specific metrics like page views, bounce rates, or conversion rates based on various features such as marketing spend, advertising channels, or content quality. Linear regression, polynomial regression, and multiple regression are commonly used techniques.
- **Time Series Anomaly Detection :** Detect anomalies in website traffic data using machine learning models. Unusual spikes or drops in traffic can indicate technical issues, bot attacks, or significant user behavior changes.
- **User Churn Prediction :** Predict user churn or attrition by building models that identify patterns leading to users discontinuing their engagement with the website. This can inform retention strategies.
- **Online Learning :** Implement online learning models that adapt to changing user behavior patterns in real-time. These models can continuously update and retrain as new data becomes available.
- **Data Integration and Fusion :** Combine website traffic data with other data sources, such as CRM data, social media data, or external market data, to enhance the predictive power of your models.

Incorporating machine learning models for predicting future traffic trends or user behavior patterns can provide a competitive advantage by enabling data-driven decision-making and proactive strategies to enhance the user experience on your website.

ALGORITHM USED:

To achieve the above stated objective, we introduce an efficient machine learning algorithm called ARIMA, which is one of the commonly used methods in time series analysis and prediction.

ARIMA:

ARIMA, which stands for Auto Regressive Integrated Moving Average, is a popular time series forecasting algorithm used in machine learning and statistics. It is designed to model and predict time series data by capturing and accounting for the temporal dependencies and patterns within the data.

1. Auto Regressive (AR) Component:

- The "AR" component represents the autoregressive part of the model. It captures the relationship between the current observation and previous observations in the time series. In other words, it models the dependency of the current value on its own past values. The order of the autoregressive component is denoted as "p," and it signifies the number of past observations to consider.

2. Integrated (I) Component:

- The "I" component represents differencing, which is the process of making a non-stationary time series stationary. Stationarity is a crucial assumption for many time series models, including ARIMA. The "I" component indicates how many differences are needed to achieve stationarity. If the data is already stationary, "I" is set to 0.

3. Moving Average (MA) Component:

- The "MA" component represents the moving average part of the model. It models the relationship between the current observation and past white noise (random) error terms in the time series. The order of the moving average component is denoted as "q," and it specifies the number of past error terms to consider.

The order of the ARIMA model is typically denoted as (p, d, q), where "p" is the order of the autoregressive component, "d" is the order of differencing, and "q" is the order of the moving average component. The ARIMA model is trained on historical time series data.

Once the ARIMA model is trained, it can be used to make future predictions. The model utilizes the past observations and its own parameters to forecast future values in the time series.

WHY ARIMA?

ARIMA (Auto Regressive Integrated Moving Average) is well-suited for time series analysis for several reasons:

- Modelling Temporal Dependencies: ARIMA models capture temporal dependencies in time series data. They account for how the current value is related to past values, making them effective at modeling and forecasting sequences with patterns that repeat over time.

- ARIMA models provide interpretable coefficients for each component (AR, I, MA), which can help in understanding the temporal relationships within the data. This

interpretability is valuable for making business decisions or gaining insights from the data.

ARIMA models are relatively simple and lightweight, making them computationally efficient and easy to implement. They don't require extensive computational resources compared to more complex models like deep neural networks. Popular libraries like `statsmodels` in Python and the "forecast" package in R provide tools for implementing ARIMA models.

APPLYING ARIMA TO WEB TRAFFIC PREDICTION:

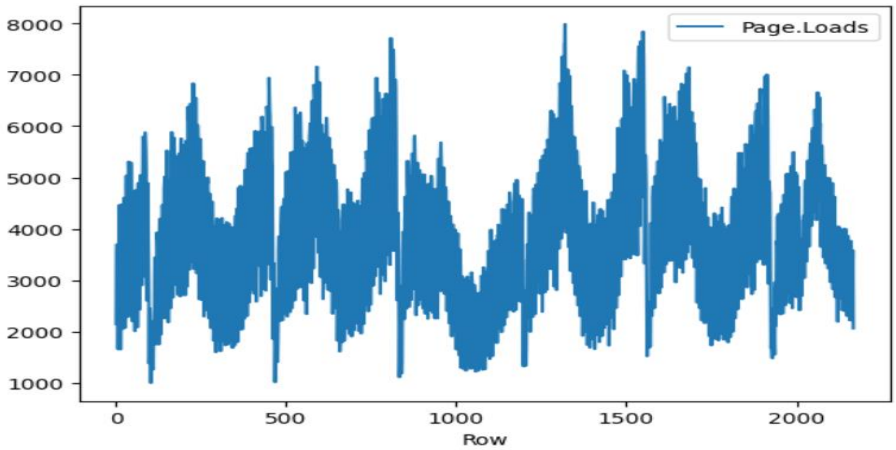
- **Data Collection**

  The primary step involves cleaning and preprocessing the collected data which includes page views, unique visitors, or other relevant metrics, ensuring that it's in a suitable format for time series analysis. This involves handling missing values and other discrepancies such as datatype mismatch etc…

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2167 entries, 0 to 2166
Data columns (total 6 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Row                2167 non-null   int64
 1   Day.Of.Week        2167 non-null   int64
 2   Page.Loads         2167 non-null   int64
 3   Unique.Visits      2167 non-null   int64
 4   First.Time.Visits  2167 non-null   int64
 5   Returning.Visits   2167 non-null   int64
dtypes: int64(6)
memory usage: 101.7 KB
```

- **Time Series Exploration**

  The next step is to visualize the website traffic data to identify trends, seasonality, and any other patterns. This initial exploration helps in understanding the data.

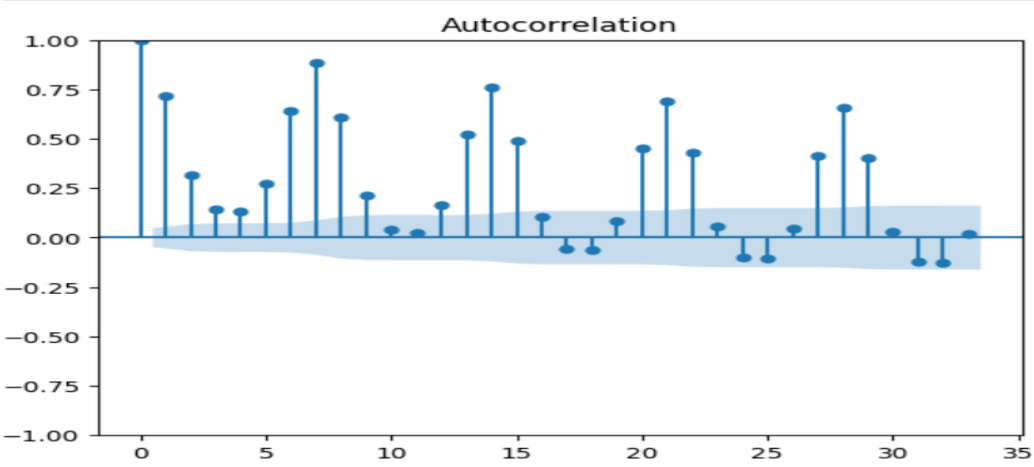


- **Stationarity Check**

In time series analysis, ACF (Auto Correlation Function) and PACF (Partial Auto Correlation Function) are two important tools used to understand the autocorrelation or temporal dependence within a time series.

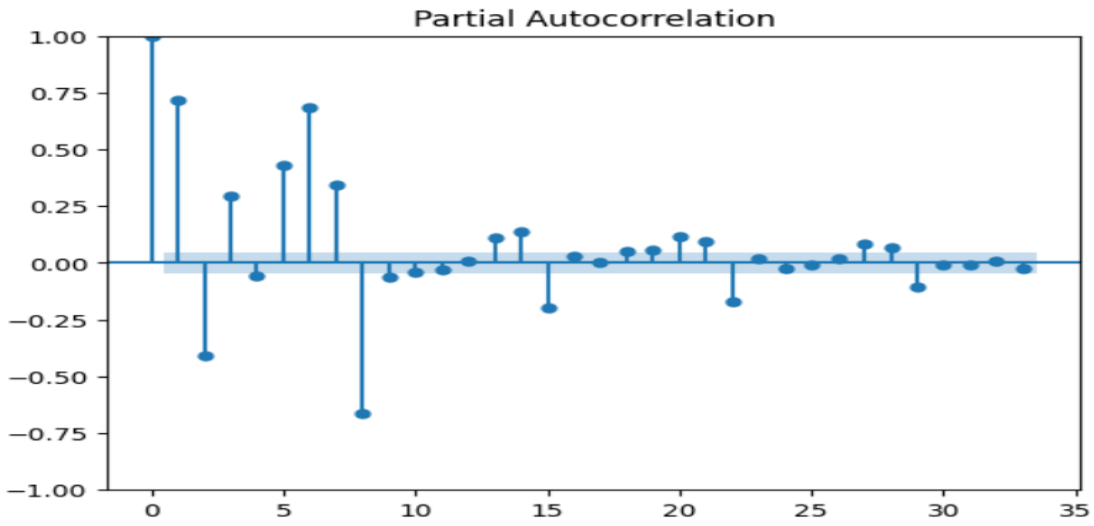
1. **Auto Correlation Function (ACF)**

The ACF measures the linear relationship between a time series and its lagged values. It computes the correlation coefficient between the time series at a given time point and the same time series at previous time points (lags). The ACF plot shows how the current value of a time series is related to its past values at different lags. A strong positive or negative correlation at a specific lag suggests that past observations at that lag have an impact on the current observation.



2. Partial Auto Correlation Function (PACF)

The PACF measures the correlation between a time series and its lagged values while controlling for the influence of intervening lags. It helps identify the direct (partial) effect of a specific lag on the current value, excluding the effects of the intervening lags. The PACF plot is useful for identifying the order (p) of the autoregressive (AR) component in an ARIMA model. It indicates which lags have a direct influence on the current value.



Both ACF and PACF plots are commonly used in the process of selecting appropriate orders (p and q) for an ARIMA model, where:

- p represents the order of the autoregressive (AR) component (number of lags to consider in the past).
- q represents the order of the moving average (MA) component (number of past forecast errors to consider).

Next we use the Augmented Dickey-Fuller (ADF) test or other methods to check whether the time series is stationary.

The ADF test is primarily used to check for the presence of a unit root in a time series, which is indicative of non-stationarity. To determine whether to reject the null hypothesis, you compare the ADF statistic to critical values at a chosen significance level (alpha). Common significance levels are 0.05 and 0.01.

```
from statsmodels.tsa.stattools import adfuller
adf_test = adfuller(df_train[param])
print(f'p-value: {adf_test[1]}')
```

```
p-value: 0.00022288390389911994
```

If the data is not stationary, we apply differencing to make it stationary, otherwise we proceed with further steps. Here p-value is 0.0002 which is less than 0.05 (under the threshold), hence we continue without differencing.

- Model Selection

  The next step is to choose an appropriate ARIMA model order (p, d, q). This involves determining the number of autoregressive (AR) terms (p), differencing order (d), and moving average (MA) terms (q).

  Manual estimation of parameters:

  In our project, manually we have chosen the parameters to be p = 2, d = 1, q = 0 to train the ARIMA model using training data set. The entire data set is split into two sets, one containing 1667 entries in the training data set and another containing 500 entries in the test data set.

```
from statsmodels.tsa.arima.model import ARIMA
model = ARIMA(df_train[param], order=(2,1,0))
model_fit = model.fit()
print(model_fit.summary())
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:              Page.Loads   No. Observations:                 1667
Model:                 ARIMA(2, 1, 0)   Log Likelihood                -17.660
Date:                Wed, 11 Oct 2023   AIC                             41.321
Time:                        21:31:22   BIC                             57.575
Sample:                             0   HQIC                            47.344
                                - 1667
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.3110      0.022     14.157      0.000       0.268       0.354
ar.L2         -0.4765      0.025    -18.869      0.000      -0.526      -0.427
sigma2         0.0598      0.002     25.125      0.000       0.055       0.064
===================================================================================
Ljung-Box (L1) (Q):                   1.76   Jarque-Bera (JB):                46.37
Prob(Q):                              0.18   Prob(JB):                         0.00
Heteroskedasticity (H):               0.85   Skew:                            -0.34
Prob(H) (two-sided):                  0.05   Kurtosis:                         2.53
===================================================================================
```

Model selection is also done through analysis, experimentation, or automated tools like auto-ARIMA.

```
import pmdarima as pm
auto_arima = pm.auto_arima(df_train[param], stepwise=False, seasonal=False)
auto_arima
```

```
▾              ARIMA

ARIMA(5,1,0)(0,0,0)[0] intercept
```

- Model Training
  The selected ARIMA model is then trained on a portion of the historical data. This allows the model to estimate its parameters based on past traffic patterns.

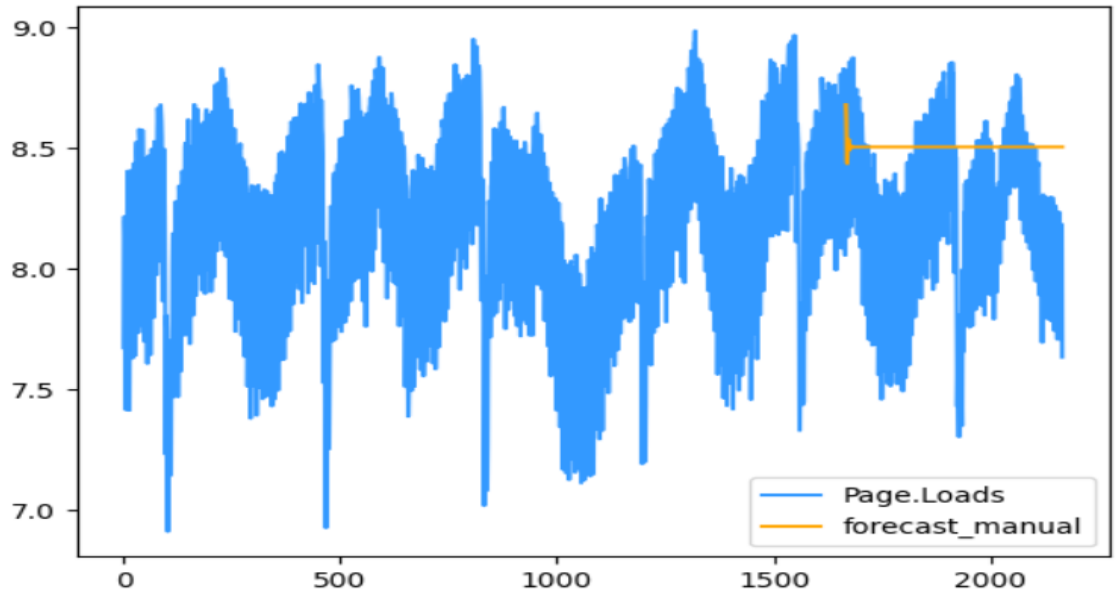
Dep. Variable:	y	**No. Observations:**	1667
Model:	SARIMAX(5, 1, 0)	**Log Likelihood**	791.511
Date:	Wed, 11 Oct 2023	**AIC**	-1569.021
Time:	22:06:08	**BIC**	-1531.094
Sample:	0	**HQIC**	-1554.966
	- 1667		
Covariance Type:	opg		

- Model Validation
  The performance of the ARIMA model is then validated over a holdout dataset.

- Forecasting
  The trained ARIMA model is employed to make traffic predictions for future periods. These predictions can be valuable for planning and resource allocation.

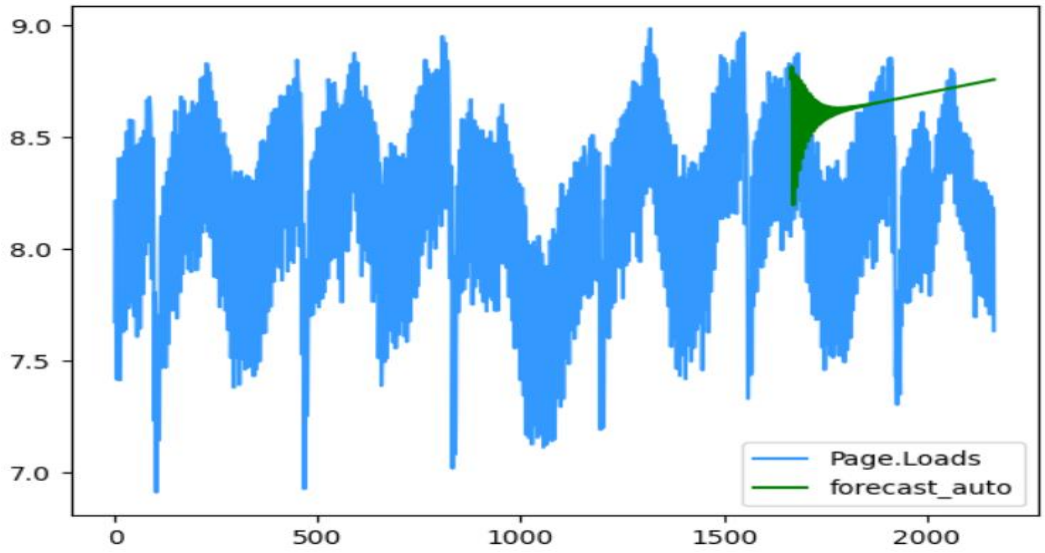
  Manual Parameter based prediction:

```
forecast_test = model_fit.forecast(len(df_test))

df['forecast_manual'] = [None]*len(df_train) + list(forecast_test)

df[[param,'forecast_manual']].plot(color=['#3399ff','orange'])
```



Auto – ARIMA based prediction:

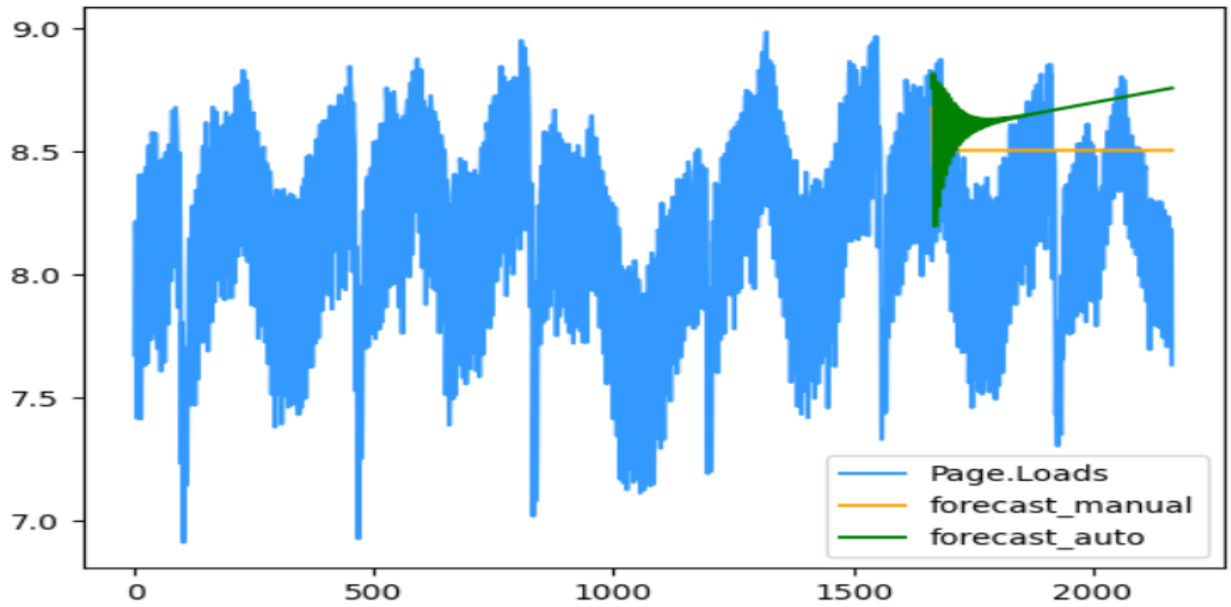
```
forecast_test_auto = auto_arima.predict(n_periods=len(df_test[param]))
df['forecast_auto'] = [None]*len(df_train[param]) + list(forecast_test_auto)

df[[param,'forecast_auto',]].plot(color=['#3399ff','green'])
```



ARIMA model parameters may need tuning and refinement, and different orders can be experimented with to achieve better forecasting accuracy.

MODEL EVALUATION:

The performance of the ARIMA model is assessed using various metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE).



Evaluation metrics:

- Mean Absolute Error (MAE): The average absolute difference between the actual and forecasted values.
- Mean Squared Error (MSE): The average of the squared differences between actual and forecasted values.
- Root Mean Squared Error (RMSE): The square root of the MSE, providing error in the same units as the data.
- Mean Absolute Percentage Error (MAPE): The average percentage difference between actual and forecasted values.

Manual method:

```
mae - manual: 0.29704959231892286
mape - manual: 0.03701199782139722
mse - manual: 0.15254232756192615
rmse - manual: 0.39056667492494307
```

Auto ARIMA method:

```
mae - auto: 0.4017289158308398
mape - auto: 0.04997490341769592
mse - auto: 0.24986339528970478
rmse - auto: 0.49986337662375785
```

The above evaluation metrics show that the values predicted using manually estimated p, d and q parameters are more closer to the original values in the test data set than those predicted using auto – ARIMA estimated parameters. The error values shown above show the deflection of the model's prediction from the original trend of the historical data.

COMPARISON OF ARIMA WITH OTHER MODELS:

A comparison of ARIMA with other ML models in terms of their efficiency for website traffic prediction:

ARIMA model	Other ML based prediction models
Well-suited for capturing linear temporal dependencies and seasonality in time series data.	Flexibility: ML models can capture a wide range of data patterns, including non-linear relationships and complex interactions
Simple and interpretable, making it easy to understand the model's parameters	Ability to handle large and high-dimensional data. Suitable for handling diverse types of data sources beyond time series data.
Often effective for data with straightforward trends and seasonality.	Capacity to incorporate external factors and feature engineering for more accurate predictions.

May not perform well on complex, non-linear, or irregular data patterns.	Complexity: Some ML models, particularly deep learning models, can be complex and computationally intensive, requiring significant data and resources.
Assumes that the data is stationary, which might not hold true for all website traffic data.	May require substantial preprocessing and feature engineering.
Limited ability to incorporate external factors or exogenous variables that can influence website traffic	May not provide as much interpretability as ARIMA, making it challenging to explain model decisions.

1.ARIMA vs. Exponential Smoothing:

Efficiency Comparison:

- **Complexity:** ARIMA models can be more complex due to the need for manual parameter tuning. Exponential smoothing is relatively simpler as it automatically assigns weights.
- **Handling Complexity:** ARIMA is better suited for time series data with complex patterns, such as multiple seasonalities and trends.
- **Ease of Use:** Exponential smoothing may be more user-friendly for beginners as it doesn't require the same level of parameter selection and tuning.
- **Performance on Simple Data:** Exponential smoothing may perform well on time series data with simple patterns, while ARIMA might be overkill.

BOUNCE RATE ANALYSIS:

The bounce rate is a crucial metric that measures the percentage of visitors who navigate away from a website after viewing only a single page or spending a very brief amount of time on that page. In essence, it quantifies how many users "bounce" off your website without engaging further by visiting additional pages or taking any action. Understanding and managing bounce rates is essential for optimizing website performance and user experience.

Bounce Rate is typically defined as the percentage of single-page visits (i.e., visitors who leave your site after viewing only one page) over the total number of visits to your website.

Bounce Rate = (Number of Single-Page Visits) / (Total Number of Visits)

```
total_page_loads = df['Page.Loads'].sum()
total_bounces = df['First.Time.Visits'].sum()

bounce_rate = (total_bounces / total_page_loads) * 100
print(f"Bounce Rate: {bounce_rate:.2f}%")
```

```
Bounce Rate: 93.56%
```

OBSERVATIONS MADE FROM BOUNCE RATES:

- A high bounce rate indicates that visitors are not finding what they expected or that the website's content or design needs improvement.
- It also suggests that visitors are not engaging with the user interface of the website.
- Relevance of Landing Page: If the landing page doesn't match the visitor's expectations or search intent, they are more likely to bounce.
- Website Speed: Slow-loading pages can lead to higher bounce rates as users become frustrated and abandon the site.
- User Experience: A cluttered, confusing, or unattractive website design can deter visitors from exploring further.
- Content Quality: Low-quality, irrelevant, or uninformative content can lead to quick exits.

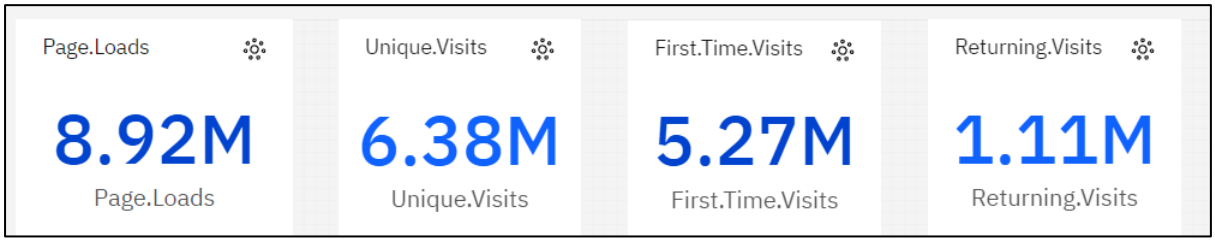
ANALYTIC INSIGHTS

- **Trend Analysis:** Identify long-term and short-term trends in page loads, unique visitors, and other metrics to understand how your website's popularity and performance change over time.
- **Day-of-Week Analysis:** Determine which days of the week receive the most traffic. This can help you schedule content updates or marketing campaigns for optimal days.
- **Seasonal Patterns:** Recognize seasonal trends(daily, monthly and yearly page loads and unique visits etc…) or recurring patterns in website traffic, which can guide your content and marketing strategy.
- **Page Load Time Analysis:** Correlate page load times with changes in page loads and visitor behaviour.
- **Forecasting:** Use time series data to make predictions about future website traffic, allowing for better resource allocation and planning. The dataset consists of time series data from 14.09.2019 to 19.08.2020. We use the IBM Cognos tool to visualize the forecast of page_loads, first_time_visits and returning_visits from 20.08.2020 onwards

VISUALISATION USING COGNOS:

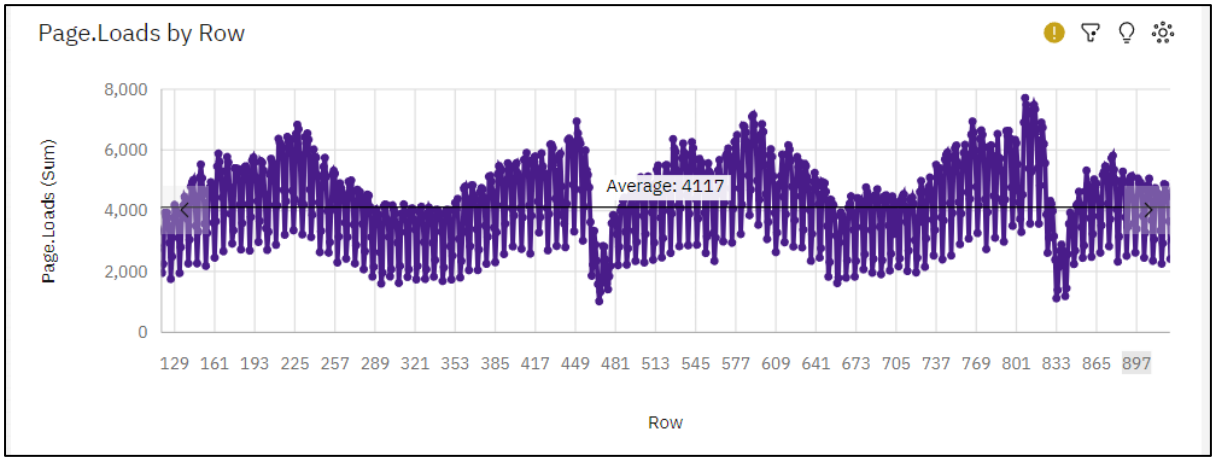
1.Count of Page Loads, Unique Visits, First Time Visits, Returning Visits

Used the summary visualisation to get a summary of page loads, unique visits, first time visits, returning visits.



2. Page Loads vs Row

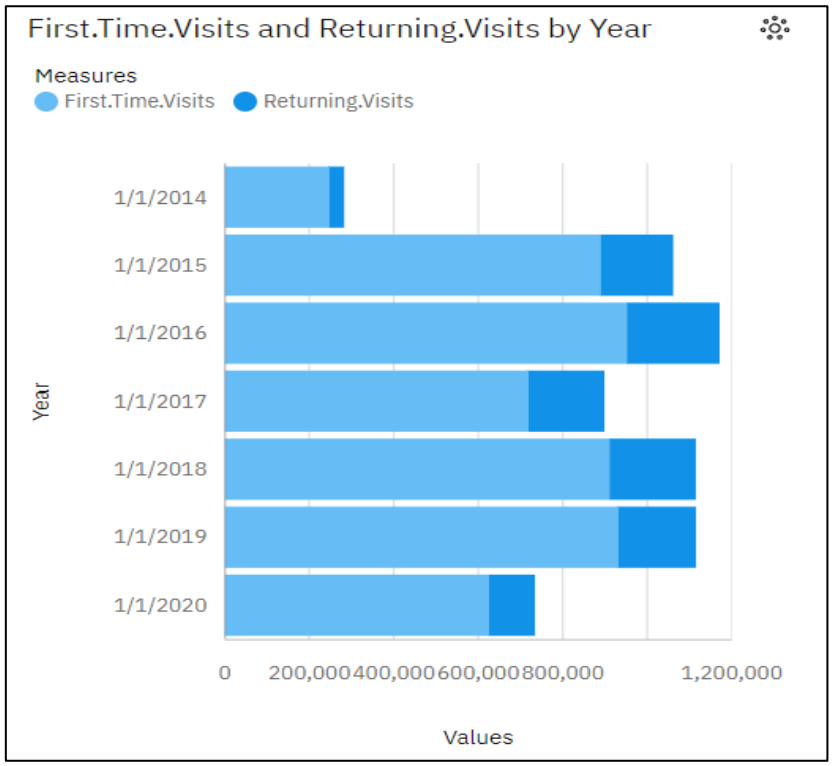
The below line graph is used to visualise the trend followed by the no. of page visits on the website.



OBSERVATIONS:

- Row 1320 has the highest total Page Loads due to Day Wednesday.
- Day Tuesday has the highest Page Loads at over 1.5 million, out of which Row 808 contributed the most at over 7500.
- Across all rows, the sum of Page Loads is over 8.9 million.
- Page Loads ranges from over a thousand, when Row is 103, to nearly eight thousand, when Row is 1320.

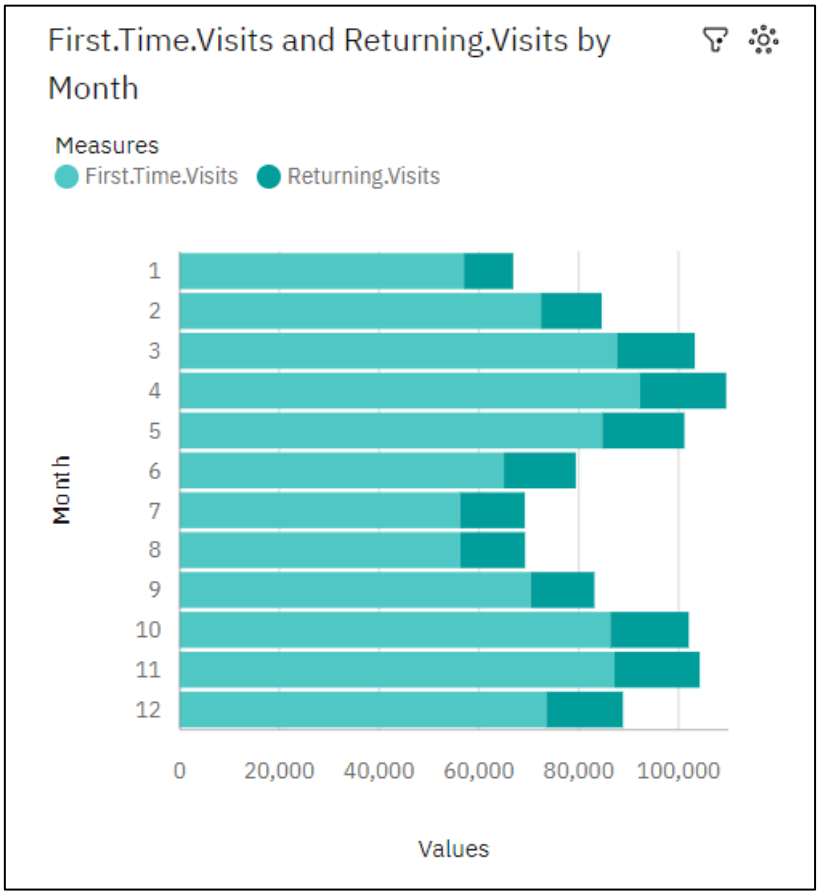
3. First Time Visits and Returning Visits by Year



OBSERVATIONS

- Year 2016 has the highest values of both First Time Visits and Returning Visits
- First Time Visits is unusually low in 2014.
- Based on the current forecasting, Returning Visits may reach over 87 thousand by Day Monday+1.
- 2016 (16.9 %), 2019 (16.8 %), 2018 (16.8 %), 2017 (16.8 %), and 2015 (16.8 %) are the most frequently occurring categories of Year with a combined count of 1826 item with First Time Visits values (84.3 % of the total).
- 2016 (16.9 %), 2019 (16.8 %), 2018 (16.8 %), 2017 (16.8 %), and 2015 (16.8 %) are the most frequently occurring categories of Year with a combined count of 1826 items with Returning Visits values (84.3 % of the total).
- Overall years, the average of First Time Visits is almost 2500.
- Overall years, the average of Returning Visits is 511.8.
- The total number of results for First Time Visits, across all years, is over two thousand.
- The total number of results for Returning Visits, across all years, is over two thousand.
- First Time Visits ranges from over 246 thousand, in 2014, to over 951 thousand, in 2016.
- Returning Visits ranges from over 36 thousand, in 2014, to nearly 220 thousand, in 2016.

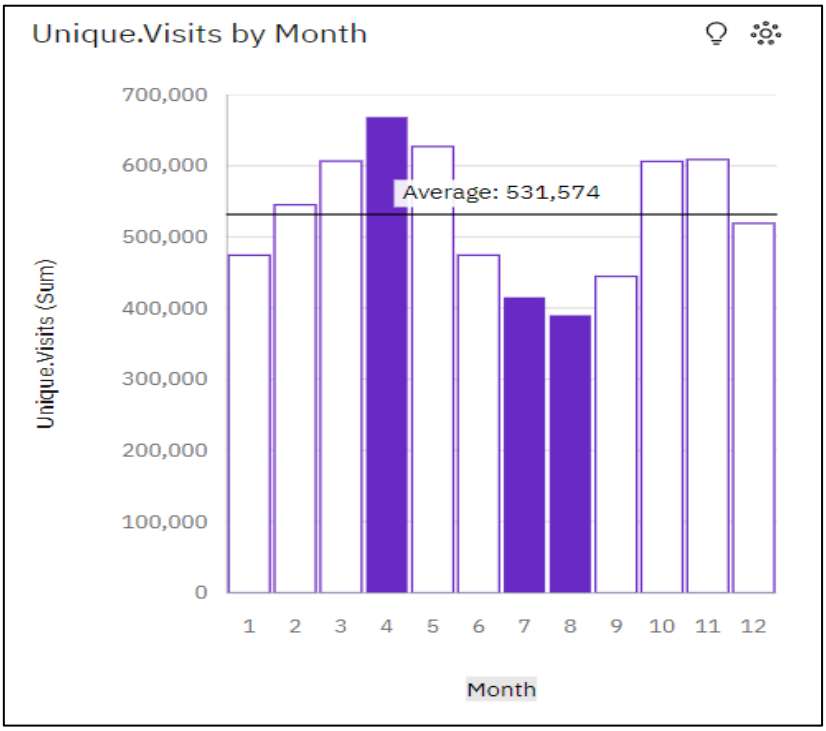
4. First Time Visits and Returning Visits by Month



OBSERVATIONS:

- Month 4 has the highest values of both First Time Visits and Returning Visits.
- Based on the current forecasting, Returning Visits may reach nearly 13 thousand by Day Monday+1.
- 1 (8.5 %), 12 (8.5 %), 3 (8.5 %), 5 (8.5 %), and 7 (8.5 %) are the most frequently occurring categories of Month with a combined count of 155 items with First Time Visits values (42.5 % of the total).
- 1 (8.5 %), 12 (8.5 %), 3 (8.5 %), 5 (8.5 %), and 7 (8.5 %) are the most frequently occurring categories of Month with a combined count of 155 items with Returning Visits values (42.5 % of the total).
- Over all months, the average of First Time Visits is almost 2500.
- Over all months, the average of Returning Visits is 473.
- The total number of results for First Time Visits, across all months, is 365.
- The total number of results for Returning Visits, across all months, is 365.
- First Time Visits ranges from over 56 thousand, when Month is 7, to over 92 thousand, when Month is 4.
- Returning Visits ranges from nearly ten thousand, when Month is 1, to over seventeen thousand, when Month is 4.

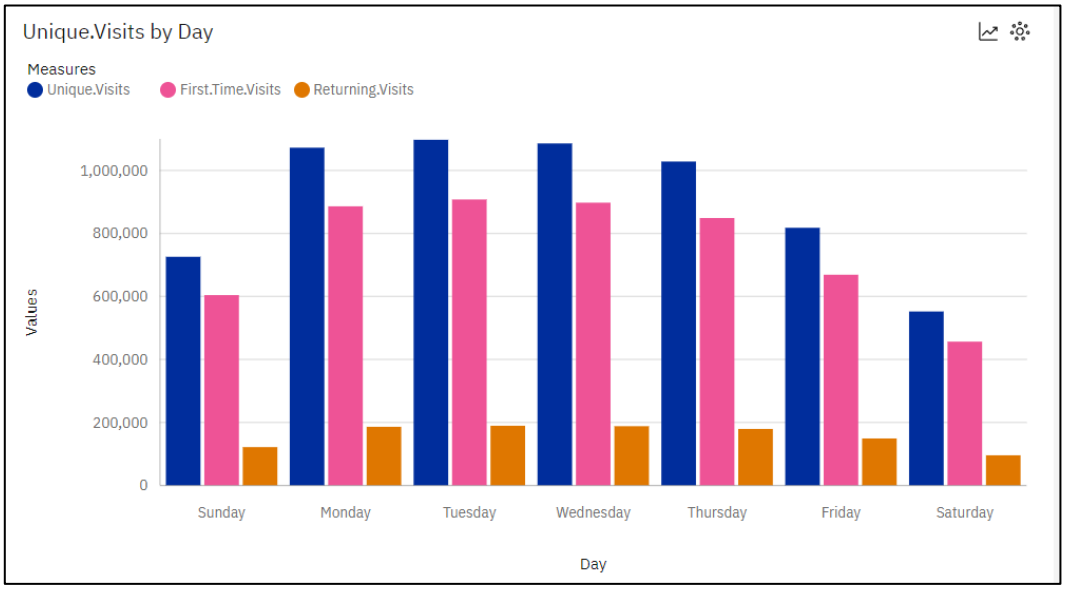
5. Unique Visits by Month



OBSERVATIONS

- Unique Visits is most unusual in 8, 4 and 7.
- It is projected that by Monday+1, 2 will exceed 6 in Unique Visits by over 3500.
- Over all months, the sum of Unique Visits is almost 6.4 million.
- Unique Visits ranges from nearly 390 thousand, when Month is 8, to over 668 thousand, when Month is 4.

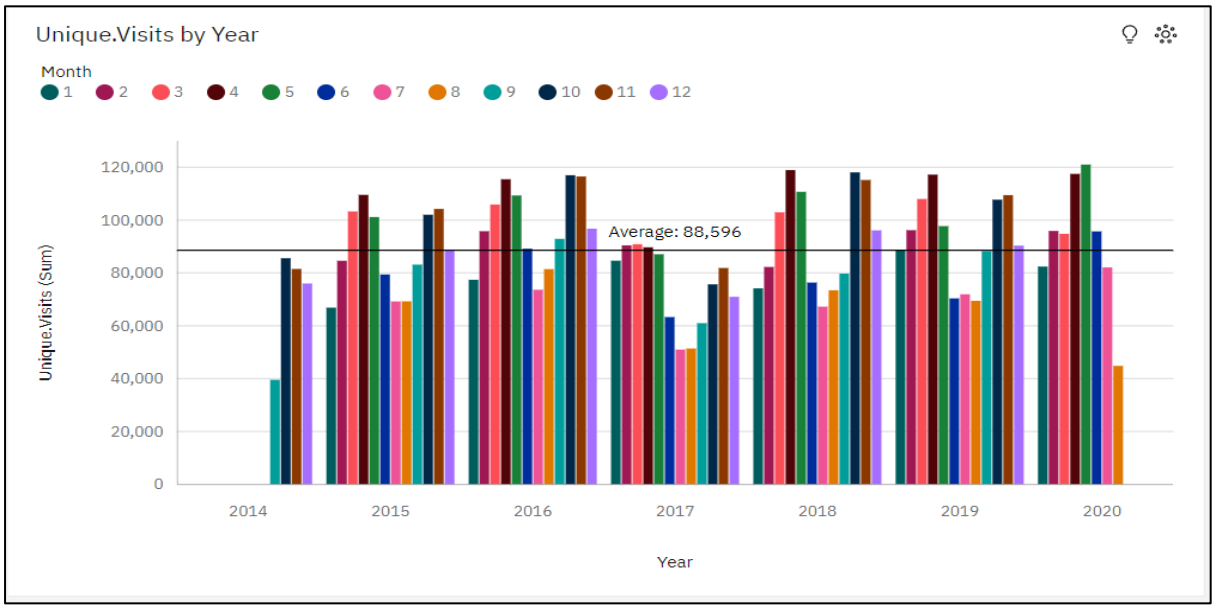
6. Unique Visits by Day



OBSERVATIONS

- Unique Visits is unusually low when Day is Saturday.
- Based on the current forecasting, Unique Visits may reach almost 481 thousand by Day Monday+1.
- 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 First Time Visits values (57.2 % of the total).
- 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 Returning Visits values (57.2 % of the total).
- 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).
- Over all days, the average of First Time Visits is almost 2500.
- Over all days, the average of Returning Visits is 511.8.
- Over all days, the average of Unique Visits is nearly three thousand.
- The total number of results for First Time Visits, across all days, is over two thousand.
- The total number of results for Returning Visits, across all days, is over two thousand.
- The total number of results for Unique Visits, across all days, is over two thousand.
- First Time Visits ranges from over 456 thousand, when Day is Saturday, to nearly 908 thousand, when Day is Tuesday.
- Returning Visits ranges from almost 96 thousand, when Day is Saturday, to over 189 thousand, when Day is Tuesday.
- Unique Visits ranges from over 552 thousand, when Day is Saturday, to nearly 1.1 million, when Day is Tuesday.

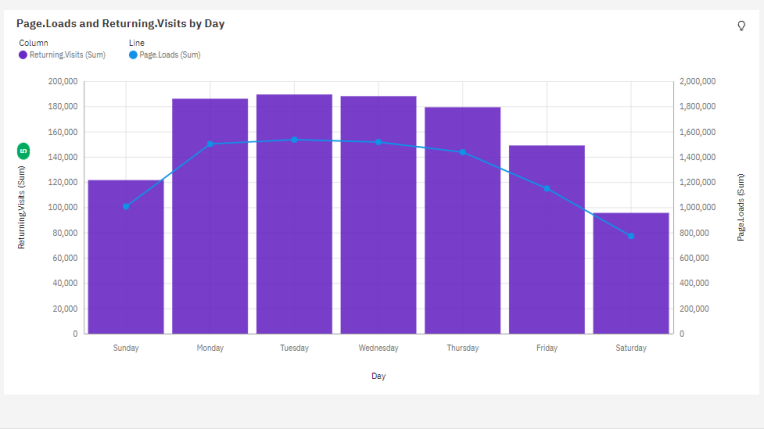
7. Unique Visits by Year



OBSERVATIONS

- Unique Visits is most unusual in 8, 4 and 7.
- Unique Visits is unusually low in 2014.
- It is projected that by Monday+1, 2 will exceed 6 in Unique Visits by over 3500.
- Month 4 has the highest Unique Visits at over 668 thousand, out of which Year 2018 contributed the most at almost 119 thousand.
- Year 2016 Unique Visits from Month 10 is nearly 117 thousand, whereas 2019 is only almost 108 thousand.
- Year 2016 has the highest total Unique Visits due to Month 10.
- 2020 has a Unique Visits of over 121 thousand for Month 5.
- Overall years and months, the sum of Unique Visits is almost 6.4 million.
- The summed values of Unique Visits range from nearly 40 thousand to over 121 thousand.
- For Unique Visits, the most significant value of Month is 4, whose respective Unique Visits values add up to over 668 thousand, or 10.5 % of the total.
- For Unique Visits, the most significant values of Year are 2016, 2019, 2018, 2015, and 2017, whose respective Unique Visits values add up to almost 5.4 million, or 84.1 % of the total.

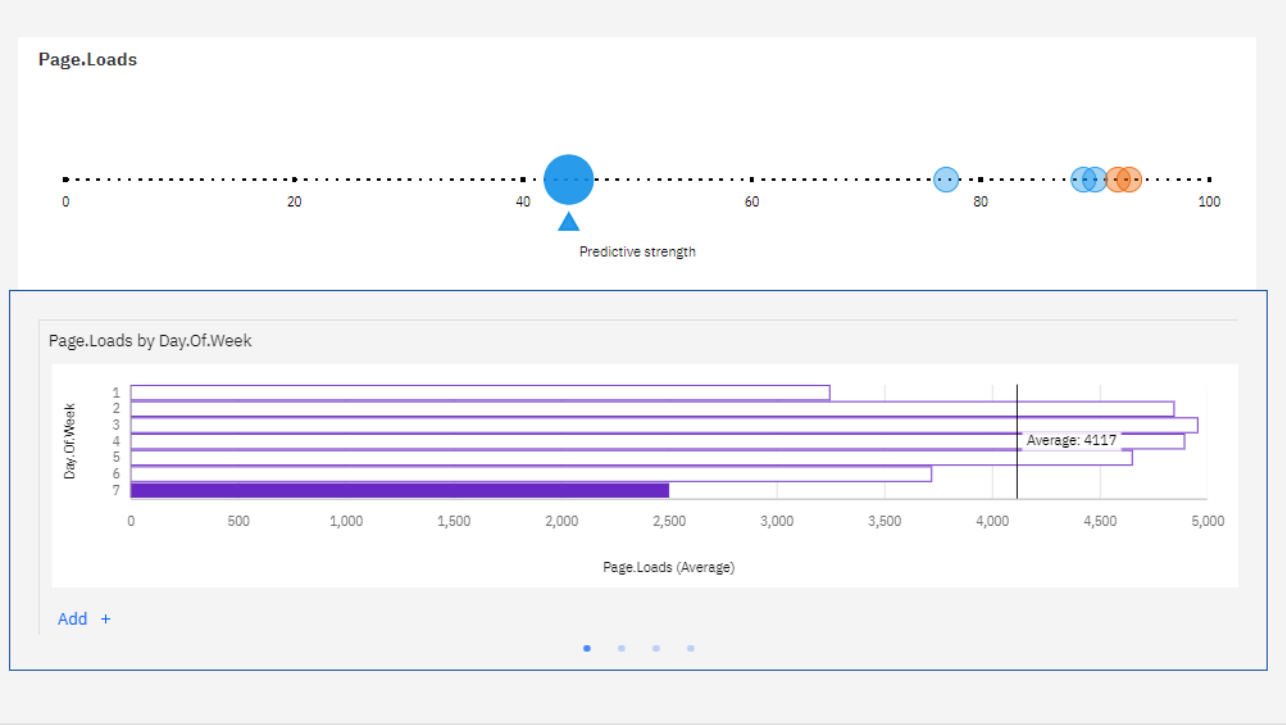
8. Page Loads and Returning Visits by Day:



OBSERVATIONS

- 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.

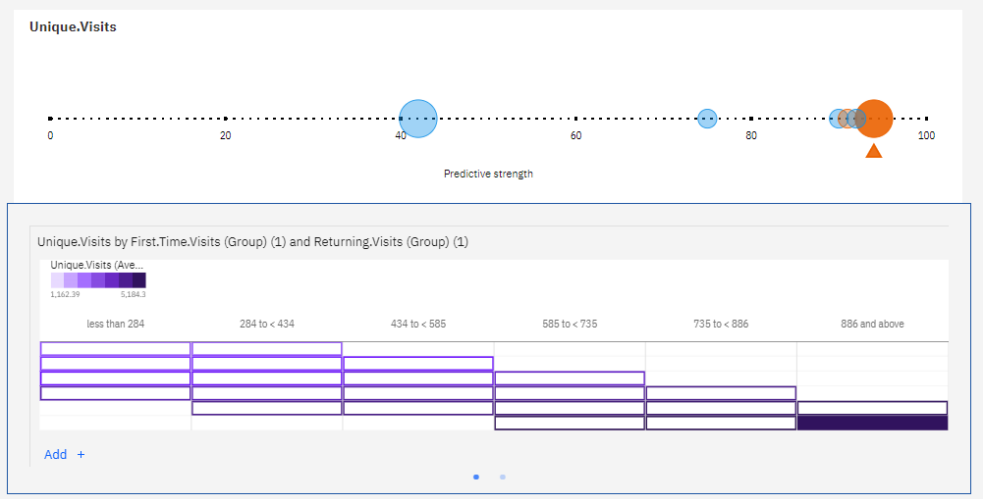
9. Page Loads by Day of Week:



OBSERVATIONS

- 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).

10. Unique Visits by First Time Visits and Returning Visits:



OBSERVATIONS

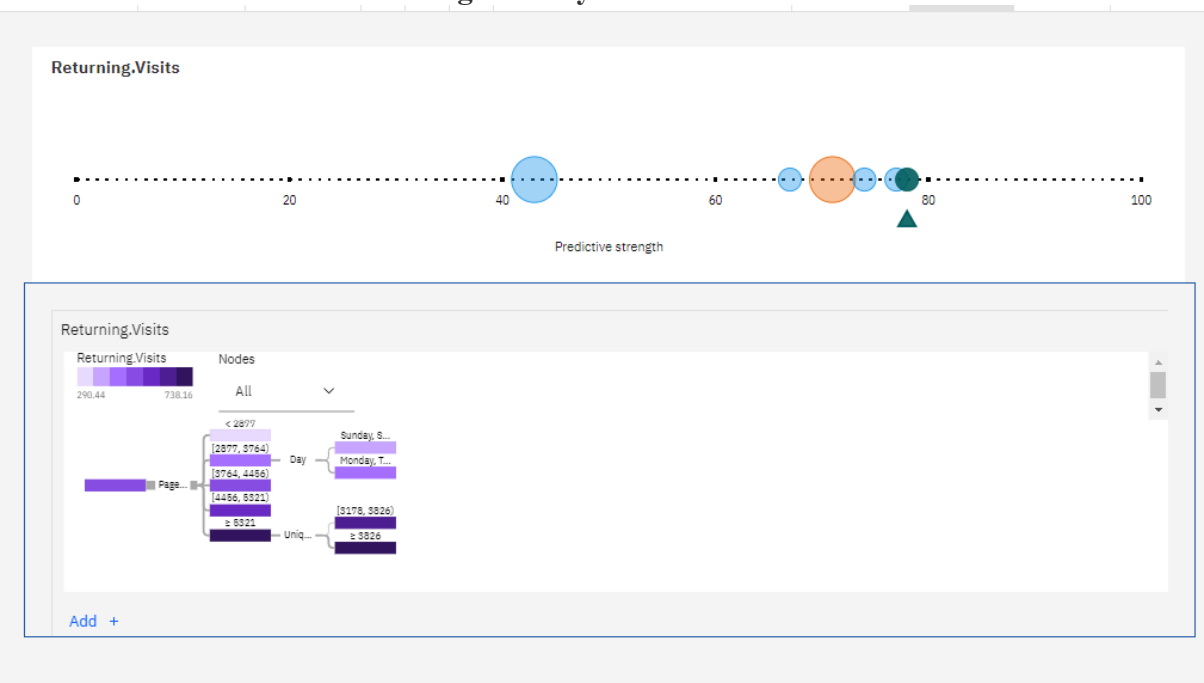
- First.Time.Visits (Group) (3) strongly affects Unique.Visits (94%).
- Unique.Visits is most unusual when First.Time.Visits (Group) (3) is 3934 and above and less than 1205.
- Returning.Visits (Group) (2) strongly affects Unique.Visits (76%).
- Unique.Visits is unusually high when Returning.Visits (Group)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

11. First Time Visits by Unique Visits and Page Loads

OBSERVATIONS

- 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.

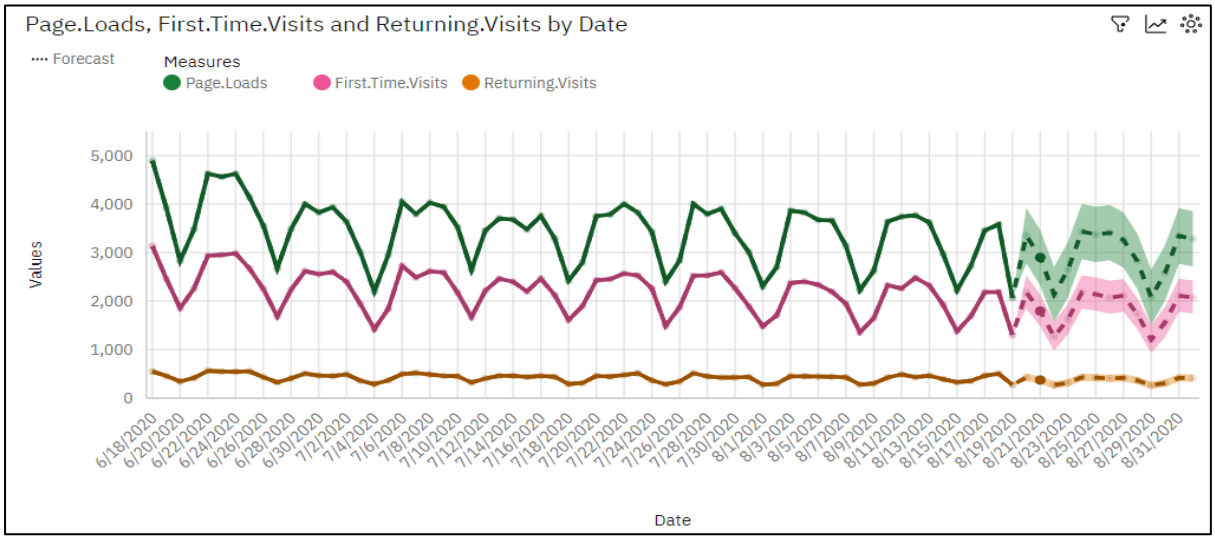
12. First time Visits and Returning visits by date:



OBSERVATIONS

- 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.

13. Forecast of page loads, first time visits and returning visits by date



OBSERVATIONS

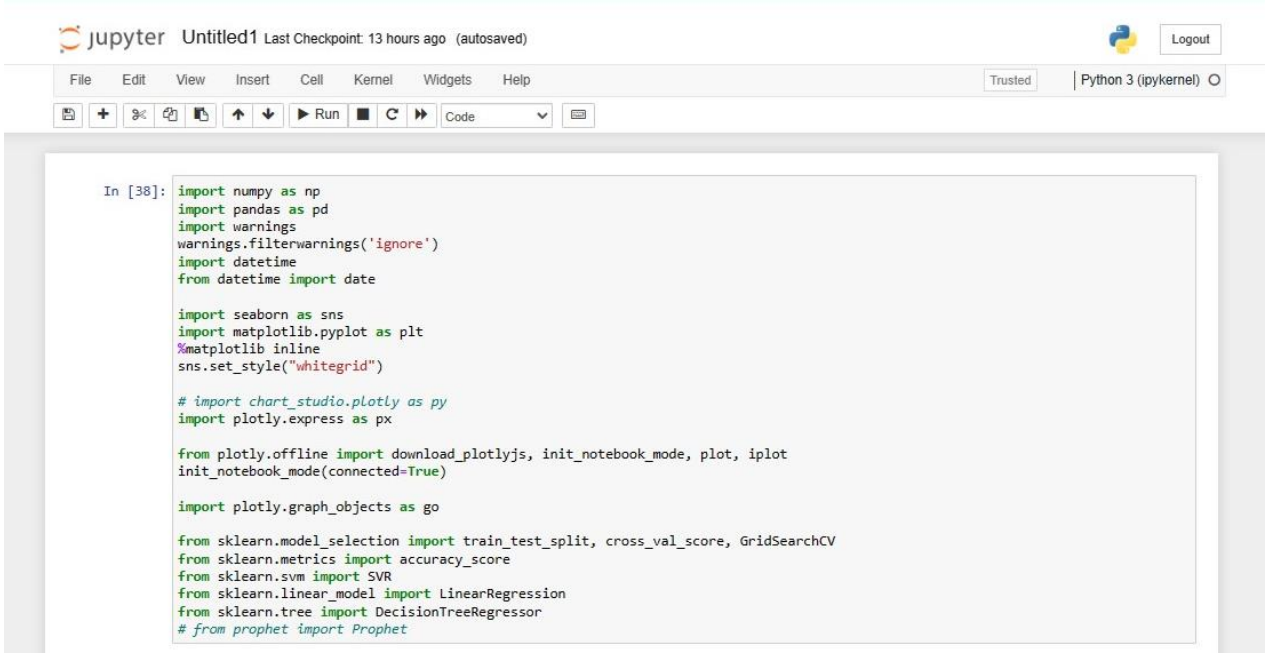
- Based on the current forecasting, Page Loads may reach almost 3500 by Date 2020-09-01.
- The value of Page Loads at the last observed time 2020-08-19 is unusual. This may indicate incomplete data or a recent event that might require investigation.
- Page Loads has an unusually low value at time point 2020-08-19.
- Over all dates, the sum of Page Loads is almost 217 thousand.
- Page Loads ranges from over two thousand, when Date is 2020-08-19, to nearly five thousand, when Date is 2020-06-18.

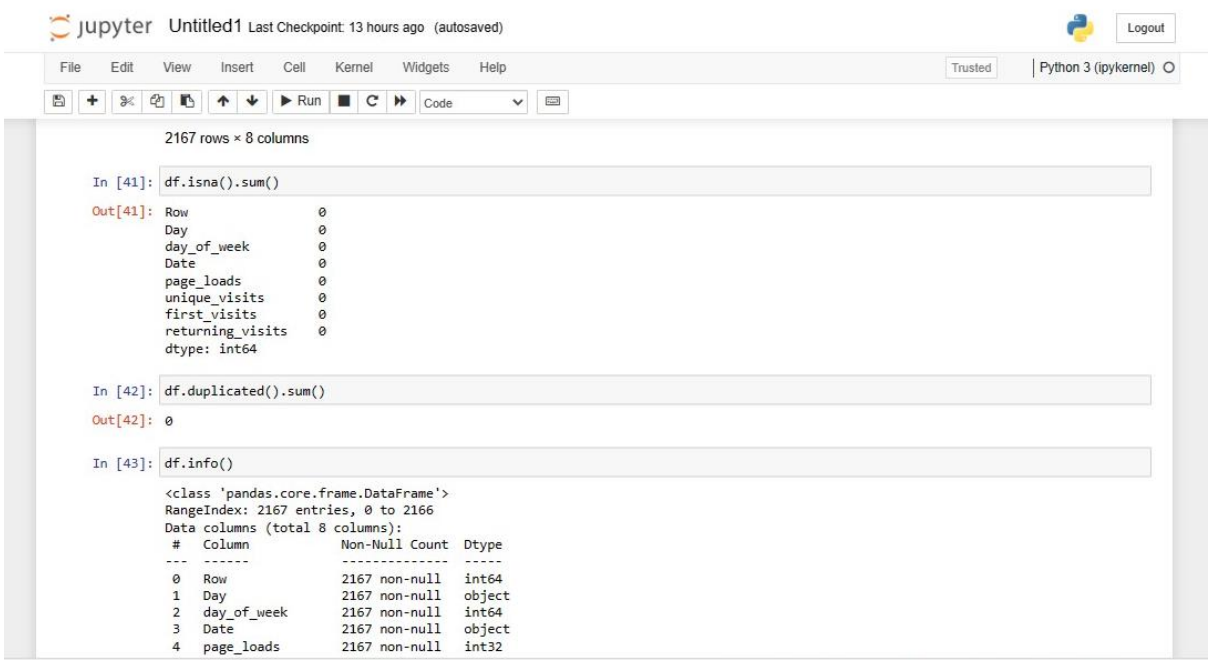
ANALYTIC INSIGHTS INCREASES USER EXPERIENCE

- **Enhanced User Engagement:** By identifying and leveraging positive trends and optimizing content and performance based on data, website owners can improve user engagement and retention.
- **Improved Resource Allocation:** Day-of-week and seasonal analysis can help allocate resources more efficiently, ensuring that user experience remains consistent even during high-traffic periods.
- **Reduced Bounce Rates:** Addressing page load time issues can reduce bounce rates, leading to higher user satisfaction and increased conversion rates.
- **Better Planning and Decision-Making:** Forecasting enables website owners to make informed decisions and be well-prepared for future challenges and opportunities.

  In summary, these analytic insights are powerful tools for website owners to understand, adapt to, and optimize user experiences based on traffic patterns, load times, and future projections, ultimately leading to a more satisfying and user-friendly online environment.

PYTHON CODE INTEGRATION:





jupyter Untitled1 Last Checkpoint: 13 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code

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

jupyter Untitled1 Last Checkpoint: 13 hours ago (autosaved)

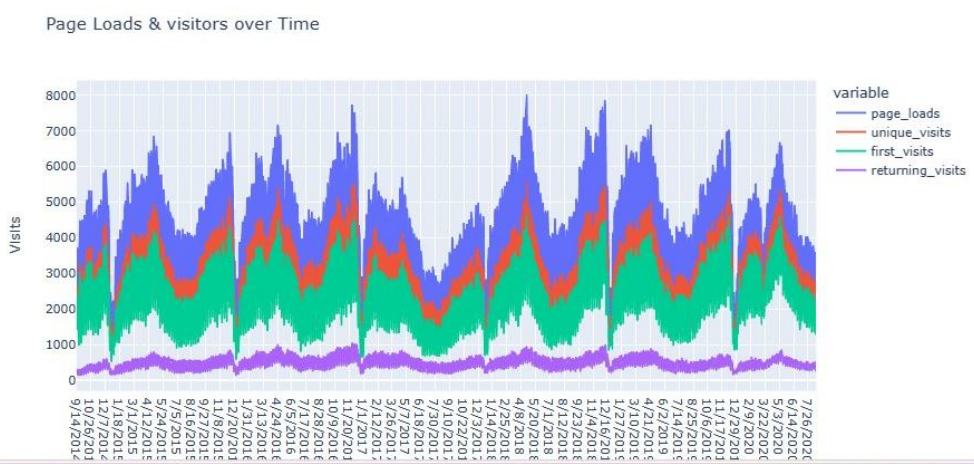
Logout

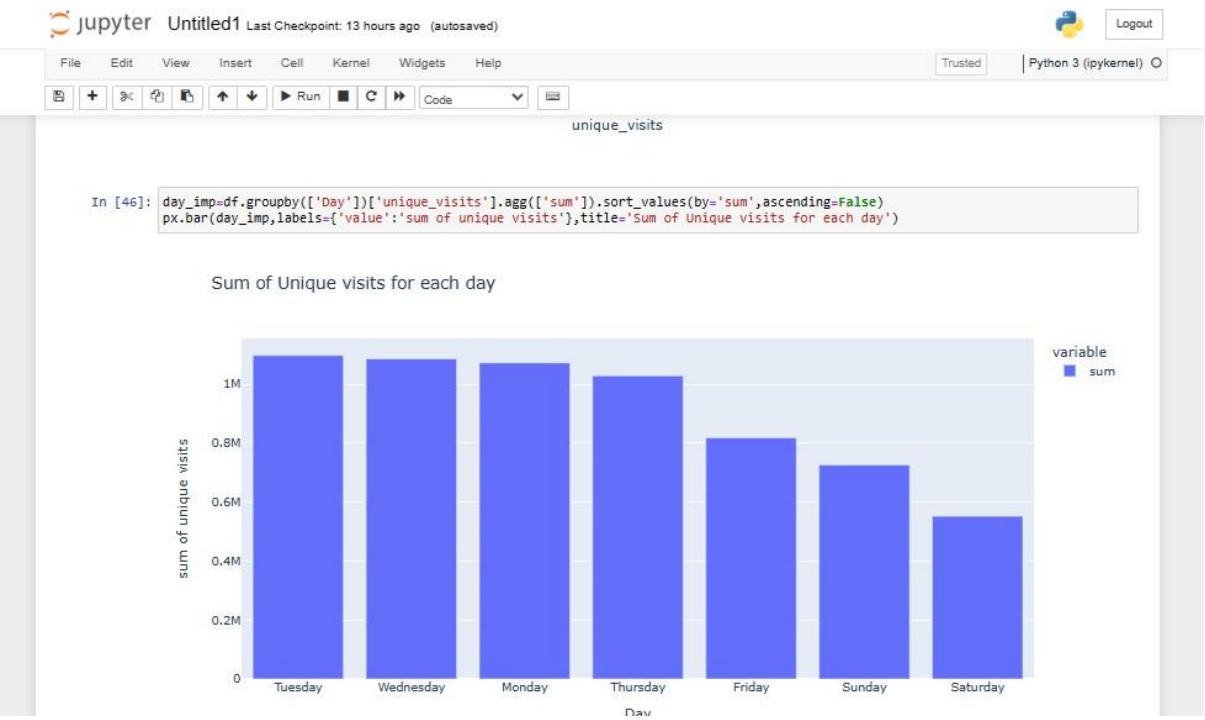
File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code

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')
```





CONCLUSION

Thus, design thinking process, analysis objectives and data visualization using Cognos for Website Traffic Analysis using ARIMA model were inferred and documented.