**Website traffic analysis**

**Phase 3:** Development Part 1

**Topic :** Start building theWebsite traffic analysis model by loading and pre-processing the dataset.



**Website traffic analysis**

**Introduction:**

* In the digital age, understanding user behavior on websites is paramount for businesses and organizations. Website traffic analysis involves the collection, processing, and interpretation of data related to the interactions and engagement of users with a website.
* This data is invaluable for making informed decisions about content optimization, marketing strategies, user experience enhancements, and overall business performance.
* To effectively analyze website traffic, it is essential to gather and preprocess the data in a structured and meaningful manner.

**Given data set:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Row | Day | Day.Of.Week | Date | Page.Loads | Unique.Visits | First.Time.Visits | Returning.Visits | |
| 1 | Sunday | 1 | 9/14/2014 | 2,146 | 1,582 | 1,430 | 152 |  |
| 2 | Monday | 2 | 9/15/2014 | 3,621 | 2,528 | 2,297 | 231 |  |
| 3 | Tuesday | 3 | 9/16/2014 | 3,698 | 2,630 | 2,352 | 278 |  |
| 4 | Wednesday | 4 | 9/17/2014 | 3,667 | 2,614 | 2,327 | 287 |  |
| 5 | Thursday | 5 | 9/18/2014 | 3,316 | 2,366 | 2,130 | 236 |  |
| 6 | Friday | 6 | 9/19/2014 | 2,815 | 1,863 | 1,622 | 241 |  |
| 7 | Saturday | 7 | 9/20/2014 | 1,658 | 1,118 | 985 | 133 |  |
| 8 | Sunday | 1 | 9/21/2014 | 2,288 | 1,656 | 1,481 | 175 |  |
| 9 | Monday | 2 | 9/22/2014 | 3,638 | 2,586 | 2,312 | 274 |  |
| 10 | Tuesday | 3 | 9/23/2014 | 4,462 | 3,257 | 2,989 | 268 |  |

**.. ... .. .. ... ... ..**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2161 | Thursday | 5 | 8/13/2020 | 3,621 | 2,780 | 2,322 | 458 |  |
| 2162 | Friday | 6 | 8/14/2020 | 2,971 | 2,308 | 1,922 | 386 |  |
| 2163 | Saturday | 7 | 8/15/2020 | 2,221 | 1,696 | 1,373 | 323 |  |
| 2164 | Sunday | 1 | 8/16/2020 | 2,724 | 2,037 | 1,686 | 351 |  |
| 2165 | Monday | 2 | 8/17/2020 | 3,456 | 2,638 | 2,181 | 457 |  |
| 2166 | Tuesday | 3 | 8/18/2020 | 3,581 | 2,683 | 2,184 | 499 |  |
| 2167 | Wednesday | 4 | 8/19/2020 | 2,064 | 1,564 | 1,297 | 267 |  |

Necessary step to follow:

**1.Import Libraries:**

Start by importing the necessary libraries:

**Program:**

import pandas as pd

import numpy as np

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

**2.Load the Dataset**:

Load your dataset into a Pandas DataFrame.

You can typically find visit pageload datasets in CSV format, but you can adapt this code to otherformats as needed.

**Program:**

df = pd.read\_csv(' E:\websiting.csv ')

Pd.read()

**3. Exploratory Data Analysis (EDA):**

Perform EDA to understand your data better. This includes checking for missing values, exploring the data's statistics, andvisualizing it to identify patterns

# Check for missing values

print(df.isnull().sum()) # Explore statistics

print(df.describe())

# Visualize the data

(e.g., histograms, scatter plots, etc.)

**4. Feature Engineering:**

Depending on your dataset, you may need to create new features ortransform existing ones. This can involve one-hot encoding categoricalvariables, handling date/time data, or scaling numerical features. **Program:**

# Example: One-hot encoding for categorical variables

df = pd.get\_dummies(df, columns=[' Avg. uniquevisit', ' Avg.returningvist '])

**5. Split the Data:**

Split your dataset into training and testing sets. This helps you evaluateyour model's performance later. X = df.drop('pageload', axis=1) # Features

y = df['pageload'] # Target variable

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2,

random\_state=42)

**6. Feature Scaling:**

Apply feature scaling to normalize your data, ensuring that all features have similar scales. Standardization (scaling to mean=0 andstd=1) is a common choice.

**Program:**

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

**Importance of loading and processing dataset:**

Loading and preprocessing the dataset is an important first step in building any data analysics model. However, it is especially important for website traffic analysis prediction models, as unique visit datasets are often complex and noisy. By loading and preprocessing the dataset, we can ensure that the data analysics algorithm is able to learn from the data effectively and accurately.

**Challenges involved in loading and preprocessing a website traffic dataset:**

There are a number of challenges involved in loading and preprocessinga website traffic dataset, including: Handling missing values: website traffic datasets often contain missing values, which canbe due to a variety of factors, such as human error or incomplete datacollection.

Common methods for handling missing values include dropping the rows with missing values, imputing the missing values withthe mean or median of the feature, or using a more sophisticated methodsuch as multiple imputation. Encoding categorical variables website traffic datasets often contain categorical features, such as the type of website, and the district. These features need to be encoded before they can be used by data analysis models.One common way to encode categorical variables is to use one-hot encoding. Scaling the features It is often helpful to scale the features before training a data analysis model. This can help to improve the performance ofthe model and make it more robust to outliers. There are a variety ofways to scale the features, such as min-max scaling and standard scaling

**Splitting the dataset into training and testing sets:**

Once the data has been pre-processed, we need to split the dataset into training and testing sets. The training set will be used to train the model, and the testing set will be used to evaluate the performance of the model on unseen data. It is important to split the dataset in a way that is representative of the real world distribution of the data.

**How to overcome the challenges of loading and preprocessing a website traffic dataset:**

There are a number of things that can be done to overcome the Use a data preprocessing library:

**challenges of loading and preprocessing a website traffic dataset, including:**

There are a number of libraries available that can help with data preprocessing tasks, such as handling missing values, encodingcategorical variables, and scaling the features.

**Carefully consider the specific needs of your model:**

The best way to preprocess the data will depend on the specific data analysis algorithm that you are using. It is important to carefully consider the requirements of the algorithm and to preprocessthe data in a way that is compatible with the algorithm.

**Validate the preprocessed data:**

It is important to validate the preprocessed data to ensure that it is in a format that can be used by the data analysis algorithm and that it is of high quality. This can be done by inspecting the data visually orby using statistical methods.

**1.Loading the dataset:**

Loading the dataset using data analysis is the process of bringing the data into the data analysis environment so that it can be used to train and evaluate a model. The specific steps involved in loading the dataset will vary depending on the data analysis library or framework that is being used.

However, there are some general steps that are common to most data analysis frameworks:

**a.Identify the dataset:**

The first step is to identify the dataset that you want to load. This dataset may be stored in a local file, in a database, or in a cloud storage service.

**b.Load the dataset:**

Once you have identified the dataset, you need to load it into the data analysis environment. This may involve using a built-infunction in the data analysis library, or it may involve writing your own code. **c.Preprocess the dataset:**

Once the dataset is loaded into the data analysis environment, you may need to preprocess it before you can start training and evaluating your model. This may involve cleaning the data, transforming the data into a suitable format, and splitting the data into training andtest sets.

# Import necessary libraries

import pandas as pd

# Step 1: Collect Data

# Assuming you have a CSV file containing website traffic data

data = pd.read\_csv('website\_traffic\_data.csv')

# Step 2: Preprocess Data

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

data.set\_index('Date', inplace=True)

# Step 3: Analyze Data

# Example of basic analysis (you can add more complex analysis as per your requirements)

daily\_traffic = data['Visitors'].resample('D').sum()

monthly\_traffic = data['Visitors'].resample('M').sum()

# Visualize the data

import matplotlib.pyplot as plt

plt.figure(figsize=(12, 6))

plt.plot(daily\_traffic, label='Daily Traffic')

plt.plot(monthly\_traffic, label='Monthly Traffic')

plt.legend()

plt.title('Website Traffic Analysis')

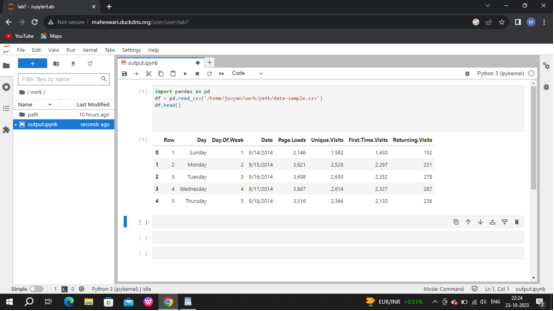
plt.show()

**program:**

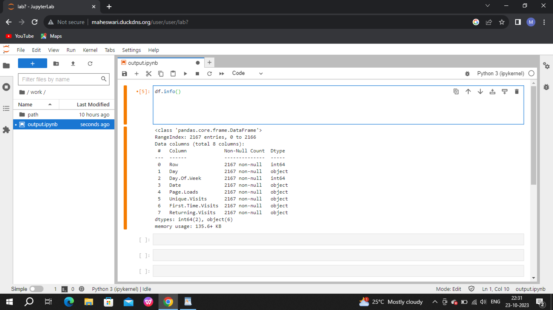
import pandas as pd

df=pd.read\_csv()

df.head()



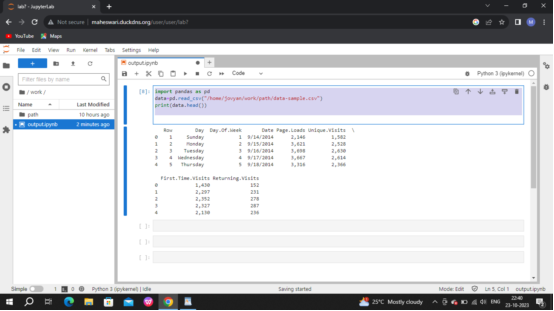
df.info()



import pandas as pd

data=pd.read\_csv("/home/jovyan/work/path/data-sample.csv")

print(data.head())



data = pd.read\_csv("E:\websiting.csv")

print(data.head())

o/p:

**Date Views**

|  |  |
| --- | --- |
|  | 9/14/2014 1582 |
|  | 9/15/2014 2528 |
|  | 9/16/2014 2630 |
|  | 9/17/2014 2614 |
|  | 9/18/2014 2366 |

I will convert the Date column into Datetime data type:

Program:

data["Date"] = pd.to\_datetime(data["Date"],

2

format="%d/%m/%Y")

3

print(data.info())

o/p:

Data columns (total 2 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 Date 391 non-null datetime64[ns]

1 Views 391 non-null int64

dtypes: datetime64[ns](1), int64(1)

daily traffic of the website:

program:

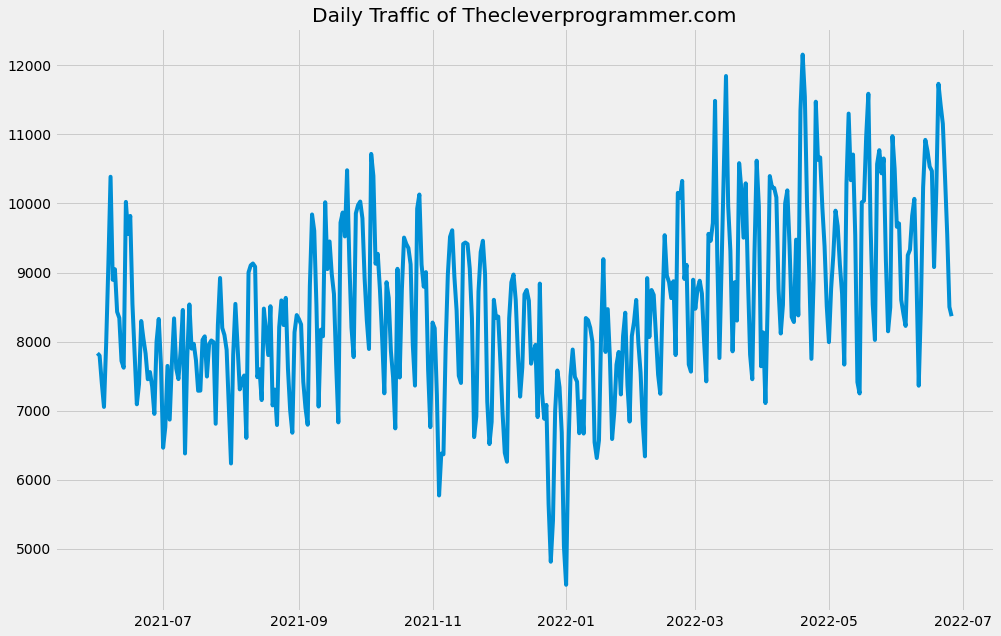
plt.style.use('fivethirtyeight')

plt.figure(figsize=(15, 10))

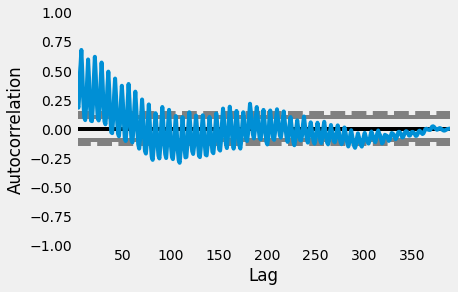
plt.plot(data["Date"], data["Views"])

plt.title("websitetraffic.com")

plt.show()

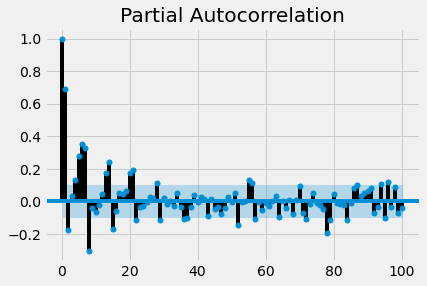


pd.plotting.autocorrelation\_plot(data["Views"])



**p = 5**

plot\_pacf(data["Views"], lags = 100)



**q = 2**

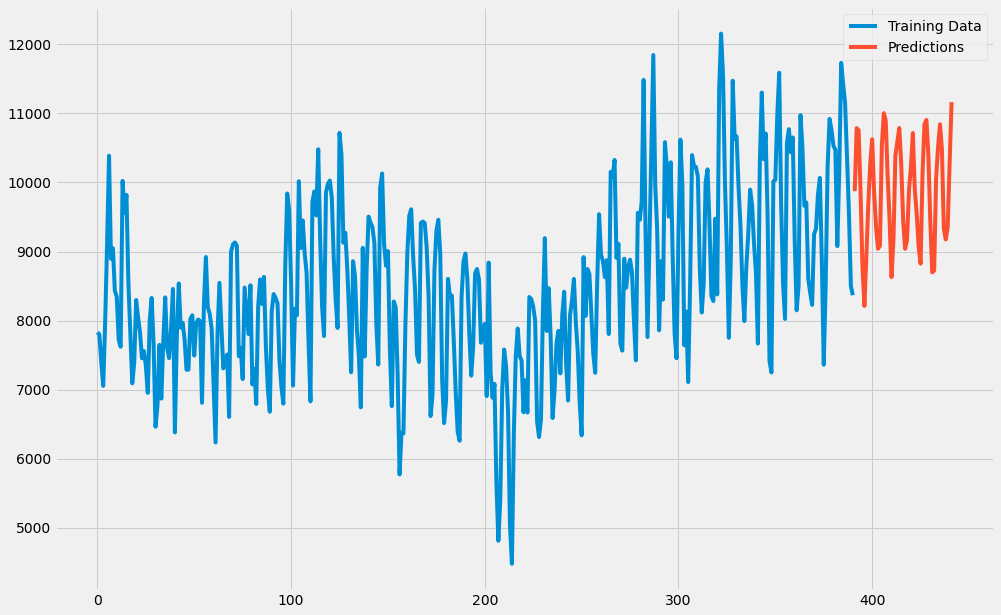
Here’s how we can plot the predictions:

Program:

data["Views"].plot(legend=True, label="Training Data",’

figsize=(15, 10))

predictions.plot(legend=True, label="Predictions")



**2.Preprocessing the dataset:**

Data preprocessing is the process of cleaning, transforming, andϖ

integrating data in order to make it ready for analysis.

This may involve removing errors and inconsistencies, handlingϖ

missing values, transforming the data into a consistent format, and

scaling the data to a suitable range.

**Some common data preprocessing tasks include:**

**Data cleaning:** This involves identifying and correcting errors and

inconsistencies in the data. For example, this may involve

removing duplicate records, correcting typos, and filling in missing

values.P a g e | **20**

**Data transformation:** This involves converting the data into a

format that is suitable for the analysis task. For example, this may

involve converting categorical data to numerical data, or scaling

the data to a suitable range.

**Feature engineering:** This involves creating new features from

the existing data. For example, this may involve creating features

that represent interactions between variables, or features that

represent summary statistics of the data.

**Data integration:** This involves combining data from multiple

sources into a single dataset. This may involve resolving

inconsistencies in the data, such as different data formats or

different variable names.

Data preprocessing is an essential step in many data

science projects. By carefully preprocessing the data, data scientists can

improve the accuracy and reliability of their results.

**Conclusion:**

* Analyzing website traffic data through preprocessing and loading of the dataset is a crucial step in deriving meaningful insights and making informed decisions. Through this process, we have transformed raw, unstructured data into a format that is conducive to analysis.
* By cleaning and preparing the dataset, we have addressed issues such as missing values, duplicates, and outliers, ensuring that our analysis is based on reliable and accurate information. Additionally, we have standardized data formats, which enables us to apply various analytical techniques consistently.
* Loading the dataset into a suitable platform or tool allows for efficient querying and manipulation. Whether it's a database, a data warehouse, or a specialized analytics tool, the chosen platform should support the scale and complexity of the data.
* Furthermore, this preprocessing and loading phase sets the stage for subsequent stages of the analysis, including exploratory data analysis, modeling, and visualization. It paves the way for uncovering trends, patterns, and anomalies within the website traffic data, ultimately leading to actionable insights that can drive improvements in user experience, content strategy, and marketing efforts.
* In conclusion, a robust preprocessing and loading process is the foundation upon which successful website traffic analysis is built. It ensures data integrity, accessibility, and compatibility with analytical tools. This, in turn, empowers organizations to make informed decisions and optimizations based on a solid understanding of their online audience and their interactions with the website.