**PROJECT PHASE 1**

**TOPIC:** **WEBSITE TRAFFIC ANALYSIS**

**INTRODUCTION:**

Website and their actions once they reach it. This information is vital for marketers to implement effective Website traffic analysis is the process of collecting and analyzing data about visitors to your tactics that optimize website traffic and improve conversion rates.

There are a number of different tools and services that can be used to analyze website traffic. One of the most popular is Google Analytics, which is a free tool that provides a wide range of data about your website visitors, including:

* Number of visitors: The number of unique visitors to your website over a given period of time.
* Page views: The total number of pages viewed on your website over a given period of time.
* Average session duration: The average amount of time that visitors spend on your website.
* Bounce rate: The percentage of visitors who leave your website after viewing only one page.
* Traffic sources: The different sources of traffic to your website, such as search engines, social media, and referral links.

There are a number of benefits to using machine learning for website traffic analysis. Some of the key benefits include:

* *Accuracy*: ML algorithms can be very accurate at predicting website traffic trends and patterns. This can help businesses to make better decisions about their marketing and website optimization strategies.
* *Scalability*: ML algorithms can be scaled to handle large datasets. This is important for businesses with high-traffic websites.
* *Automation*: ML algorithms can automate many of the tasks involved in website traffic analysis, such as data collection, cleaning, and analysis. This can free up businesses to focus on other tasks.
* *Insights*: ML algorithms can identify insights and patterns in website traffic data that would be difficult or impossible to find manually. This can help businesses to better understand their visitors and how they are interacting with their website.

**STEPS FOR PROBLEM SOLVING:**

Here is a step-by-step problem-solving guide for website traffic analysis:

1. **Identify the problem**: What specific problem are you trying to solve with website traffic analysis? Are you trying to increase website traffic, improve conversion rates, or identify new traffic sources? Once you know what problem you are trying to solve, you can focus your analysis on the relevant data.
2. **Collect data**: The first step to solving any problem is to collect data. There are a number of different tools and services that you can use to collect website traffic data, such as Google Analytics, Semrush, and Ahrefs.
3. **Clean the data**: Once you have collected your data, you need to clean it to ensure that it is accurate and complete. This may involve removing incomplete or inaccurate data, and correcting any errors.
4. **Analyze the data**: Once you have cleaned your data, you can start to analyze it to identify patterns and trends. You can use a variety of different tools and techniques to analyze your data, such as statistical analysis, data visualization, and machine learning.
5. **Identify the root cause of the problem**: Once you have identified patterns and trends in your data, you need to identify the root cause of the problem. This may involve looking at your website content, design, marketing campaigns, or other factors.
6. **Develop a solution**: Once you have identified the root cause of the problem, you need to develop a solution. This may involve making changes to your website content, design, marketing campaigns, or other factors.
7. **Implement the solution**: Once you have developed a solution, you need to implement it. This may involve making changes to your website, creating new marketing campaigns, or other tasks.
8. **Evaluate the results**: Once you have implemented the solution, you need to evaluate the results to see if it has solved the problem. If not, you may need to make further changes.

**PROBLEM SOLVING VIA DESIGN THINKING:**

Design thinking can be used to solve a wide range of problems, including problems related to website traffic analysis. Here is a step-by-step guide to using design thinking to solve problems with website traffic analysis:

1. *Empathize with your users*: The first step in design thinking is to understand the people you are designing for. In the context of website traffic analysis, this means understanding your website visitors. What are their needs and wants? What are their pain points? You can learn more about your users by conducting user research, such as interviews, surveys, and observation.
2. *Define the problem*: Once you have a good understanding of your users, you can start to define the problem you are trying to solve. Be as specific as possible and focus on the user's needs, rather than your own assumptions. For example, instead of saying "I need to increase website traffic," you could say "I need to help users find the information they need on my website more easily."
3. *Ideate*: Once you have defined the problem, it's time to start generating ideas for solutions. This is where you can get creative and come up with as many ideas as possible, no matter how crazy they may seem. Some examples of ideas for solutions to website traffic analysis problems might include:
   * Improving the website's search function
   * Creating more engaging content
   * Promoting the website on social media
   * Running paid advertising campaigns
4. *Prototype*: Once you have a few ideas for solutions, it's time to start prototyping them. This means creating rough models or sketches of your ideas so that you can test them out and get feedback from users. For example, you could create a prototype of a new search function by creating a simple web page with a search bar and some sample results.
5. *Test:* The final step in design thinking is to test your prototypes with users and get their feedback. This will help you to refine your ideas and make sure that you are creating solutions that meet the needs of your users.

**IMPORTING LIBRARY PACKAGES:**

import pandas as pd

import matplotlib.pyplot as plt

**LOAD THE DATA IN DATASET:**

website\_traffic\_df = pd.read\_csv('websitetrafficanalysis.csv')

**PRINT THE STATEMENT:**

*# Print the first 5 rows of the DataFrame*

print(website\_traffic\_df.head())

*# Print the DataFrame information*

print(website\_traffic\_df.info())

**PLOT THE STATEMENT:**

# Create a line graph to visualize the website traffic trend over time

plt.plot(website\_traffic\_df['date'], website\_traffic\_df['pageviews'])

# Set the title and labels of the graph

plt.title('Website Traffic Trend')

plt.xlabel('Date')

plt.ylabel('Pageviews')

# Show the graph

plt.show()

**OVERVIEW OF OUR FINAL PROGRAMING CODE IN PYTHON:**

**import pandas as pd**

**import matplotlib.pyplot as plt**

**# Load the website traffic data from a CSV file**

**website\_traffic\_df = pd.read\_csv('websitetrafficanalysis.csv')**

**# Print the first 5 rows of the DataFrame**

**print(website\_traffic\_df.head())**

**# Print the DataFrame information**

**print(website\_traffic\_df.info())**

**# Create a line graph to visualize the website traffic trend over time**

**plt.plot(website\_traffic\_df['date'], website\_traffic\_df['pageviews'])**

**# Set the title and labels of the graph**

**plt.title('Website Traffic Trend')**

**plt.xlabel('Date')**

**plt.ylabel('Pageviews')**

**# Show the graph**

**Plt.show()**

**OUTPUT:**

Row Day Day.Of.Week Date Page.Loads Unique.Visits \

0 1 Sunday 1 9/14/2014 2,146 1,582

1 2 Monday 2 9/15/2014 3,621 2,528

2 3 Tuesday 3 9/16/2014 3,698 2,630

3 4 Wednesday 4 9/17/2014 3,667 2,614

4 5 Thursday 5 9/18/2014 3,316 2,366

First.Time.Visits Returning.Visits

0 1,430 152

1 2,297 231

2 2,352 278

3 2,327 287

4 2,130 236

<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 object

5 Unique.Visits 2167 non-null object

6 First.Time.Visits 2167 non-null object

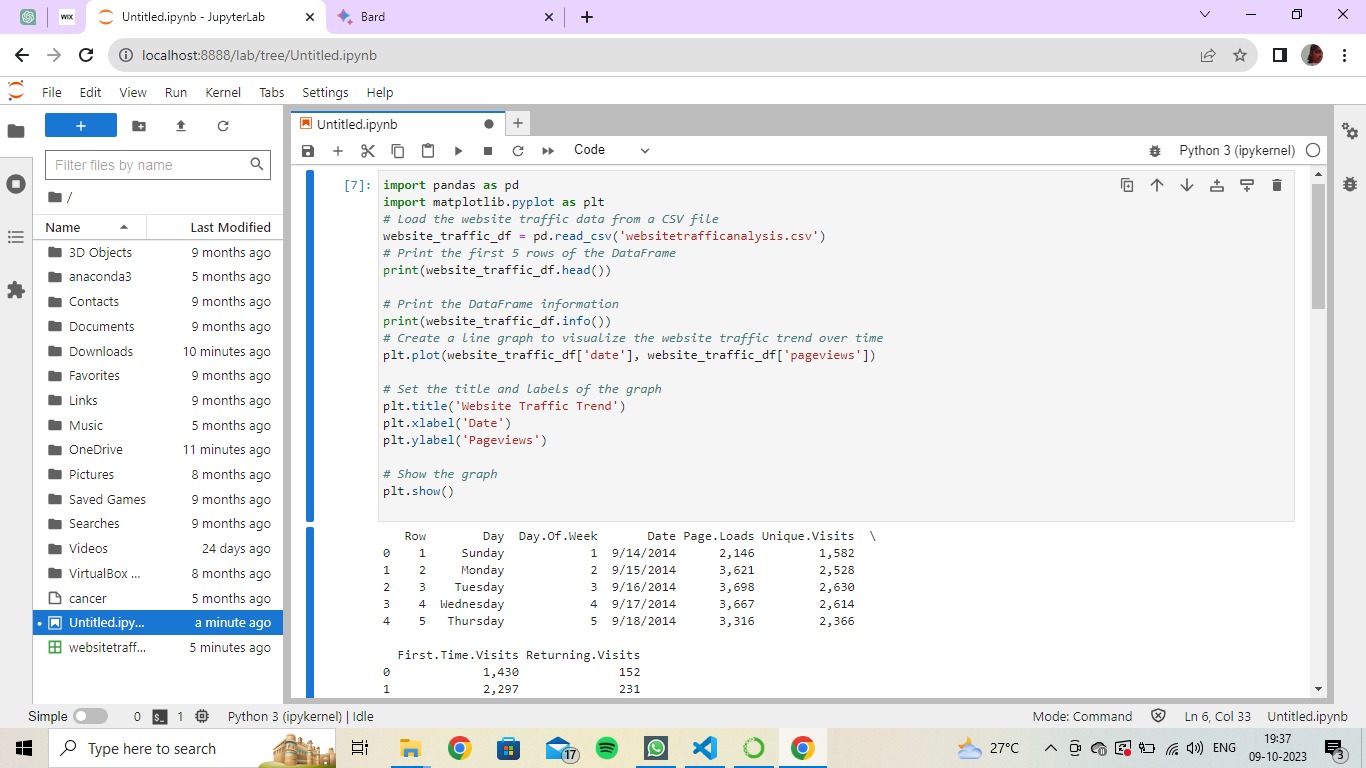
7 Returning.Visits 2167 non-null object

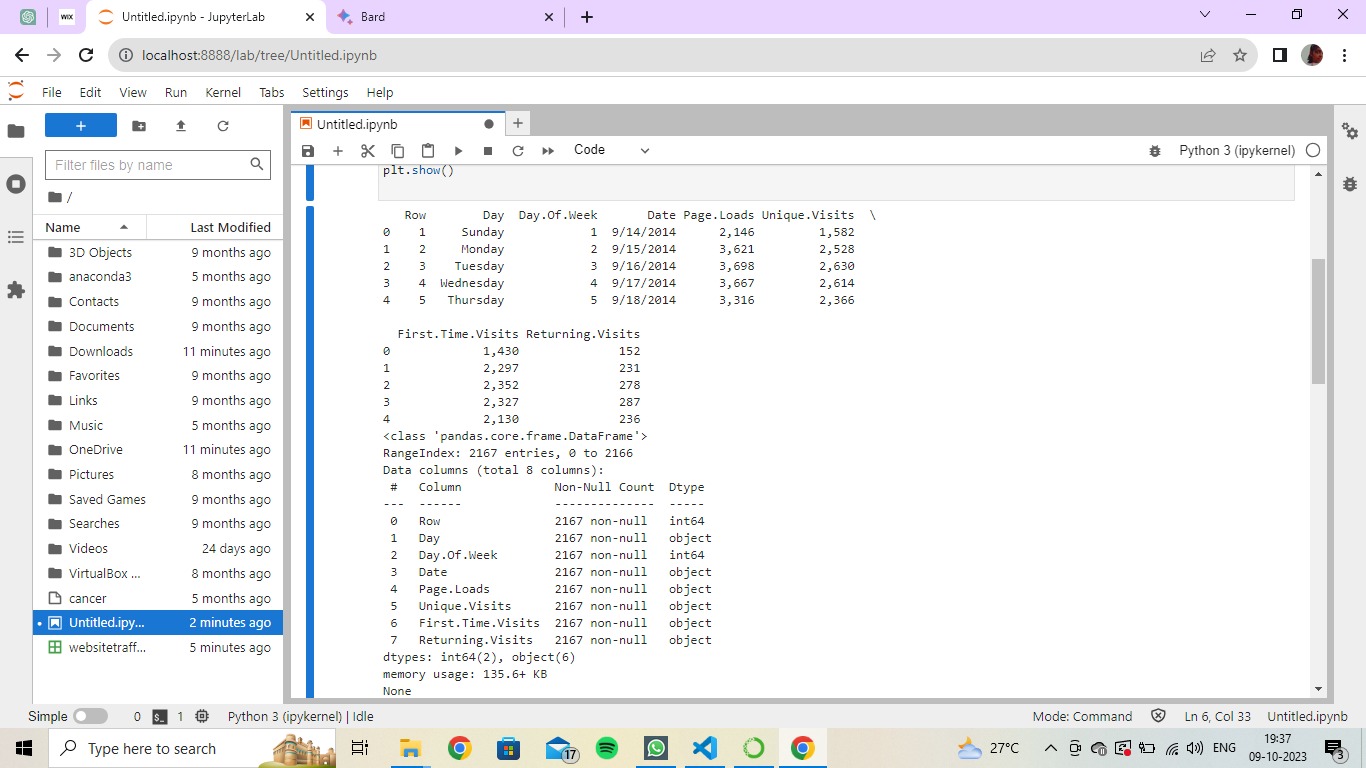
dtypes: int64(2), object(6)

memory usage: 135.6+ KB

None

**DEMONSTRATION OF OUR 1ST PROJECT:**

****

****