SYNOPSIS

Report on

Analyzing Website Traffic Data

By

Dhruv Bathla -- 202410116100062
Chetanya Bedi– 202410116100053
Deepak Sharma -- 202410116100055
Devansh Kumar – 202410116100059

Session:2024-2026 (II Semester)

Under the supervision of

MR. APPORAV JAIN (Assistant Professor)

KIET Group of Institutions, Delhi-NCR, Ghaziabad

DEPARTMENT OF COMPUTER APPLICATIONS
KIET GROUP OF INSTITUTIONS, DELHI-NCR,
GHAZIABAD-201206

INTRODUCTION

In today’s digital landscape, understanding website traffic is crucial for optimizing user experience, improving marketing strategies, and driving business growth. Website traffic analysis involves collecting, processing, and interpreting data about visitors, their behavior, and interactions on a website.

By analyzing website traffic, businesses and website owners can answer critical questions such as:

- Where are visitors coming from?
- Which pages are the most popular?
- How long do users stay on the site?
- What devices and browsers are they using?
- What factors contribute to conversions or drop-offs?

Using tools like Google Analytics, server logs, and custom tracking scripts, organizations can gain actionable insights into user behavior. These insights help in enhancing website performance, optimizing content strategy, and making data-driven decisions to boost engagement and revenue.

This analysis is particularly important for businesses relying on digital marketing, as it enables them to track campaign effectiveness, understand customer demographics, and refine their targeting strategies. By leveraging data analytics techniques such as segmentation, trend analysis, and predictive modeling, organizations can stay ahead in the competitive online space.

In this document, we will explore various aspects of website traffic analysis, the key metrics to track, the tools used, and best practices for leveraging this data effectively.

.

Literature Review

Literature Review on Analyzing Website Traffic Data

Website traffic analysis has been widely studied in the fields of digital marketing, web analytics, and data science. Various research studies and industry reports highlight the importance of tracking and analyzing visitor behavior to improve website performance and user experience.

1. Importance of Website Traffic Analysis

Several studies emphasize that website traffic data is a key determinant of online business success. According to Kotler & Keller (2016), businesses that leverage web analytics can enhance their marketing efforts by identifying user preferences and optimizing content accordingly. Similarly, Chaffey & Smith (2017) discuss how businesses can use traffic insights to improve conversion rates and customer engagement.

2. Website Traffic Metrics and Their Impact

Research by Kaushik (2019) highlights essential web traffic metrics such as page views, bounce rate, session duration, and conversion rates. These metrics provide a comprehensive understanding of user behavior and help in making data-driven decisions. Another study by Sharma & Gupta (2020) explores how analyzing click-through rates (CTR) and heatmaps can improve website design and usability.

3. Tools and Techniques for Traffic Analysis

Several tools, including Google Analytics, Matomo, and Adobe Analytics, have been studied for their effectiveness in website traffic analysis. A study by Jansen et al. (2018) compared different analytics tools and found that Google Analytics remains the most widely used platform due to its advanced tracking capabilities and real-time data processing. Moreover, machine learning techniques, such as clustering and predictive modeling, have been explored for trend prediction and segmentation in web traffic analysis (Li et al., 2021).

4. SEO and Traffic Optimization Strategies

Search Engine Optimization (SEO) plays a critical role in increasing website traffic. Moz's (2020) industry report outlines best practices for SEO, such as keyword optimization, backlinking, and mobile-friendliness, which contribute to higher search engine rankings. Additionally, a study by Patel (2021) emphasizes the role of content marketing and social media in driving organic and referral traffic.

5. Challenges in Website Traffic Analysis

Despite the benefits, analyzing website traffic comes with challenges, such as data privacy concerns, bot traffic, and data accuracy issues. Research by Smith & Brown (2022) highlights the increasing need for compliance with data protection regulations like GDPR and CCPA when collecting user data.

Methodology

Methodology for Analyzing Website Traffic Data

The methodology for analyzing website traffic data involves a structured approach to collecting, processing, and interpreting visitor interactions. This section outlines the key steps in the analysis, including data collection, preprocessing, analysis techniques, and interpretation.

---

1. Data Collection

Website traffic data is collected from various sources to ensure comprehensive insights into user behavior. The main data sources include:

- **Google Analytics & Other Web Analytics Tools** – Provides insights on visitor count, session duration, bounce rates, and user demographics.
- **Server Log Files** – Captures raw data on page requests, IP addresses, timestamps, and referrer URLs.
- **Heatmaps & Click Tracking Tools** – Records user interactions such as clicks, scrolls, and mouse movements.
- **User Surveys & Feedback Forms** – Collects qualitative data on user satisfaction and experience.
- **Social Media & Referral Traffic Data** – Identifies external sources driving visitors to the website.

2. Data Preprocessing

Raw website traffic data may contain inconsistencies and irrelevant information. The preprocessing stage ensures data accuracy and quality through:

- **Data Cleaning:** Removing bot traffic, duplicate records, and irrelevant entries.
- **Data Formatting:** Standardizing timestamps, session durations, and device categories.
- **Handling Missing Values:** Using interpolation techniques to fill missing data points.
- **Data Aggregation:** Grouping traffic data by date, source, or user type for meaningful insights.

3. Data Analysis Techniques

Once the data is preprocessed, various analytical techniques are applied:

a. Descriptive Analytics

- Identifies trends in traffic volume, user demographics, and engagement metrics.
- Uses visualizations like line charts, bar graphs, and heatmaps to depict key insights.

b. Comparative Analysis

- Compares traffic before and after implementing marketing strategies or website changes.
- Analyzes differences in visitor behavior across different time frames or user segments.

c. Predictive Analytics

- Uses machine learning models like regression analysis and time series forecasting to predict future traffic trends.
- Identifies factors that contribute to conversion rates and user retention.

d. Segmentation & Clustering

- Groups visitors based on location, behavior, device type, or acquisition source.
- Uses clustering algorithms (e.g., K-Means, DBSCAN) to identify hidden patterns in user behavior.

4. Interpretation & Decision-Making

After the analysis, the insights are interpreted to guide decision-making:

- **SEO & Content Optimization:** Enhancing keywords, backlinks, and site structure based on traffic sources.
- **Marketing Strategy Improvements:** Refining ad campaigns and targeting based on user engagement data.
- **Website Performance Enhancements:** Identifying slow-loading pages and optimizing UI/UX.
- **Conversion Rate Optimization:** Adjusting CTAs, landing pages, and sales funnels based on visitor interactions.

5. Tools & Technologies Used

Several tools are employed in different stages of website traffic analysis:

- **Data Collection:** Google Analytics, Matomo, Adobe Analytics, server logs.
- **Data Preprocessing & Storage:** Python (Pandas, NumPy), SQL, MongoDB.
- **Data Analysis & Visualization:** Tableau, Power BI, Python (Matplotlib, Seaborn, Scikit-learn).
- **Predictive Modeling:** Machine learning frameworks (TensorFlow, Scikit-learn).

CODE:

Untitled12.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text

IMPORTING BASIC LIBRARIES

```
[1] import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    import seaborn as sns
```

LOADING THE DATASET USING PANDAS PD.READ_CSV() FUNCTION

```
[5] traffic=pd.read_csv("website_trafficdata.csv")
    print("Dataset Loaded Successfully")
```

```
Dataset Loaded Successfully
```

Untitled12.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text

DISPLAYING TOP 5 ENTRIES OF THE DATASET USING HEAD() FUNCTION

```
[6] # going through data's top rows
    traffic.head()
```

	Page Views	Session Duration	Bounce Rate	Traffic Source	Time on Page	Previous Visits	Conversion Rate
0	5	11.051381	0.230652	Organic	3.890460	3	1.0
1	4	3.429316	0.391001	Social	8.478174	0	1.0
2	4	1.621052	0.397986	Organic	9.636170	2	1.0
3	5	3.629279	0.180458	Organic	2.071925	3	1.0
4	5	4.235843	0.291541	Paid	1.960654	5	1.0

Next steps: Generate code with traffic | View recommended plots | New interactive sheet

DISPLAYING LAST 5 ENTRIES OF THE DATASET USING TAIL() FUNCTION

```
[7] # going through data's top rows
    traffic.tail()
```

	Page Views	Session Duration	Bounce Rate	Traffic Source	Time on Page	Previous Visits	Conversion Rate
1995	1	2.724513	0.207187	Referral	1.324206	2	1.0
1996	3	0.392856	0.095559	Organic	3.824416	1	1.0
1997	4	9.899823	0.446622	Organic	1.288675	1	1.0
1998	3	0.393319	0.278340	Paid	5.037584	2	1.0
1999	3	0.882638	0.338026	Direct	5.186908	3	1.0

Untitled12.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text

Descriptive Analysis

```
[8] # statistical skimming of dataset
traffic.describe()
```

	Page Views	Session Duration	Bounce Rate	Time on Page	Previous Visits	Conversion Rate
count	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000
mean	4.950500	3.022045	0.284767	4.027439	1.978500	0.982065
std	2.183903	3.104518	0.159781	2.887422	1.432852	0.065680
min	0.000000	0.003613	0.007868	0.068515	0.000000	0.343665
25%	3.000000	0.815828	0.161986	1.935037	1.000000	1.000000
50%	5.000000	1.993983	0.266375	3.315316	2.000000	1.000000
75%	6.000000	4.197569	0.388551	5.414627	3.000000	1.000000
max	14.000000	20.290516	0.844939	24.796182	9.000000	1.000000

Untitled12.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text

Data Exploration

```
[9] # examining dataset's shape
traffic.shape
```

```
(2000, 7)
```

```
[10] # going through data's basic information
traffic.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 7 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Page Views        2000 non-null   int64
 1   Session Duration  2000 non-null   float64
 2   Bounce Rate       2000 non-null   float64
 3   Traffic Source    2000 non-null   object
 4   Time on Page      2000 non-null   float64
 5   Previous Visits   2000 non-null   int64
 6   Conversion Rate   2000 non-null   float64
dtypes: float64(4), int64(2), object(1)
memory usage: 109.5+ KB
```

Untitled12.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text

[11]
```
# finding null values
traffic.isnull().sum()
```

	0
Page Views	0
Session Duration	0
Bounce Rate	0
Traffic Source	0
Time on Page	0
Previous Visits	0
Conversion Rate	0

dtype: int64

[12]
```
# examining unique values of dataset
traffic.nunique()
```

	0
Page Views	15
Session Duration	2000
Bounce Rate	2000
Traffic Source	5
Time on Page	2000
Previous Visits	10
Conversion Rate	228

dtype: int64

Untitled12.ipynb

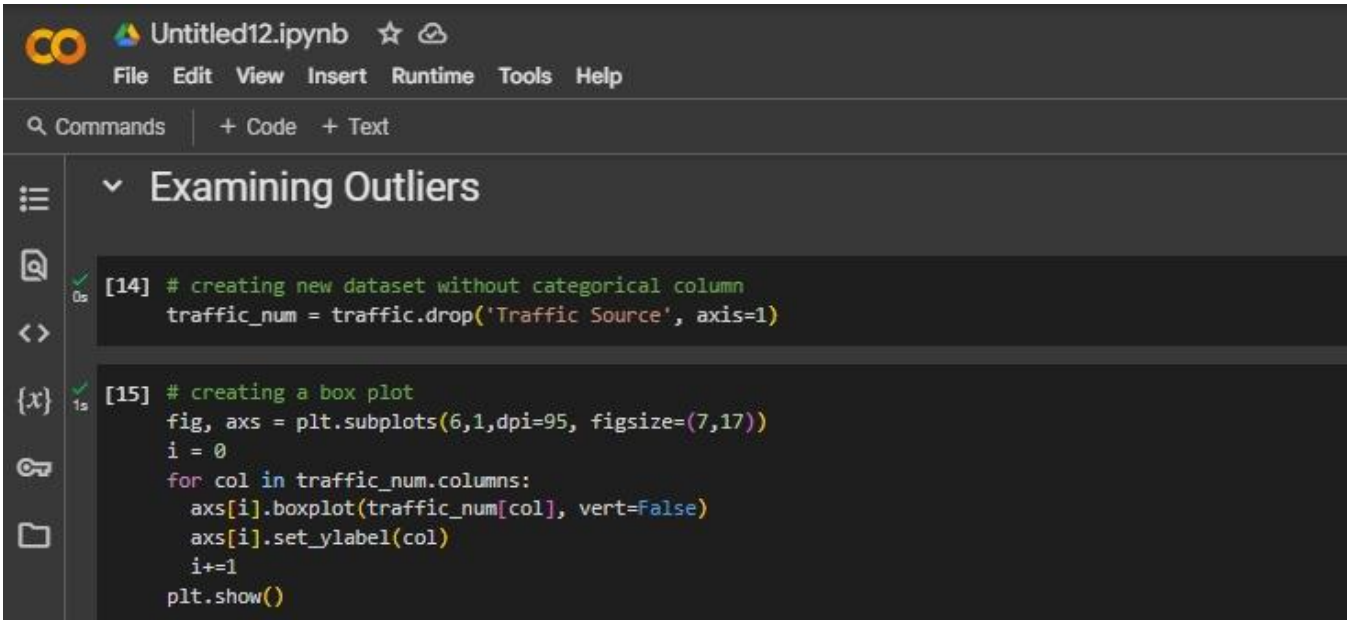
File Edit View Insert Runtime Tools Help

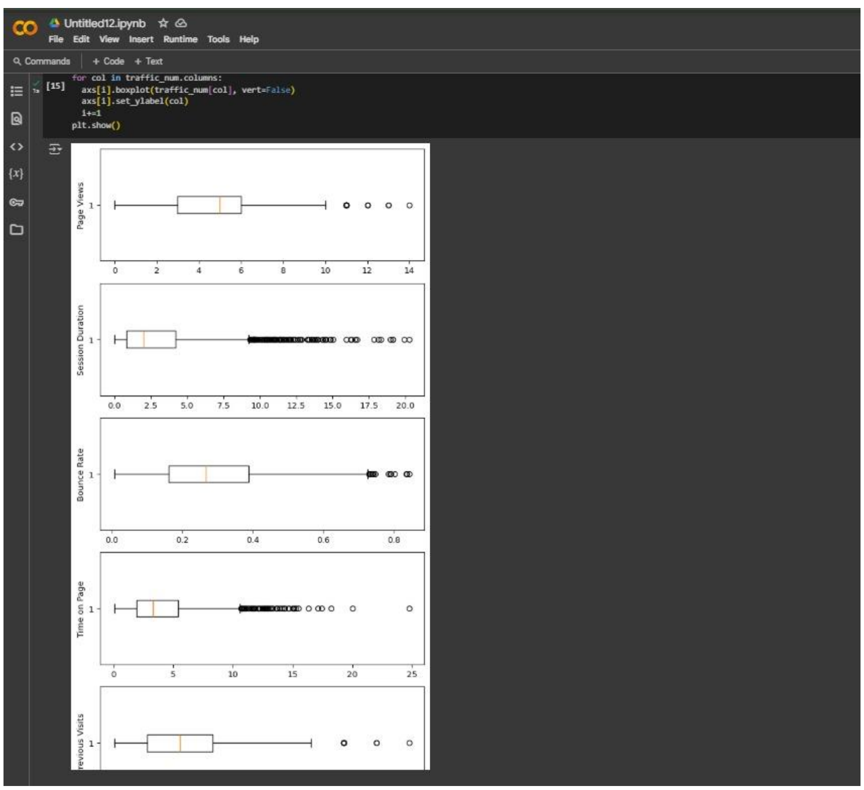
Commands + Code + Text

[13]
```
# finding value count of '1.0' in CONVERSION RATE column
con_count = traffic['Conversion Rate'].value_counts().get(1.0, 0)

# printing value count
con_count
```

```
np.int64(1773)
```


Untitled12.ipynb
File Edit View Insert Runtime Tools Help
Commands
+ Code + Text
Examining Outliers
[14] # creating new dataset without categorical column
traffic_num = traffic.drop('Traffic Source', axis=1)
[15] # creating a box plot
fig, axs = plt.subplots(6,1,dpi=95, figsize=(7,17))
i = 0
for col in traffic_num.columns:
axs[i].boxplot(traffic_num[col], vert=False)
axs[i].set_ylabel(col)
i+=1
plt.show()


Untitled12.ipynb
File Edit View Insert Runtime Tools Help
Commands
+ Code + Text
[15]
for col in traffic_num.columns:
axs[i].boxplot(traffic_num[col], vert=False)
axs[i].set_ylabel(col)
i+=1
plt.show()
Page Views
Session Duration
Bounce Rate
Time on Page

Untitled12.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text

Dropping Outliers

[16]

```
# COLUMN: Page Views

# identify the quartiles
q1, q3 = np.percentile(traffic['Page Views'], [25, 75])

# calculate the interquartile range
iqr = q3 - q1

# calculate the lower and upper bounds
lower_bound = q1 - (1.5 * iqr)
upper_bound = q3 + (1.5 * iqr)

# drop the outliers
traffic_clean = traffic[(traffic['Page Views'] >= lower_bound) & (traffic['Page Views'] <= upper_bound)]


# COLUMN: Session Duration

# identify the quartiles
q1, q3 = np.percentile(traffic['Session Duration'], [25, 75])

# calculate the interquartile range
iqr = q3 - q1

# calculate the lower and upper bounds
lower_bound = q1 - (1.5 * iqr)
upper_bound = q3 + (1.5 * iqr)

# drop the outliers
traffic_clean = traffic[(traffic['Session Duration'] >= lower_bound) & (traffic['Session Duration'] <= upper_bound)]


# COLUMN: Bounce Rate

# identify the quartiles
q1, q3 = np.percentile(traffic['Bounce Rate'], [25, 75])

# calculate the interquartile range
iqr = q3 - q1
```

Untitled12.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text

[16]
```python
# calculate the lower and upper bounds
lower_bound = q1 - (1.5 * iqr)
upper_bound = q3 + (1.5 * iqr)

# drop the outliers
traffic_clean = traffic[(traffic['Bounce Rate'] >= lower_bound) & (traffic['Bounce Rate'] <= upper_bound)]


# COLUMN: Time on Page

# identify the quartiles
q1, q3 = np.percentile(traffic['Time on Page'], [25, 75])

# calculate the interquartile range
iqr = q3 - q1

# calculate the lower and upper bounds
lower_bound = q1 - (1.5 * iqr)
upper_bound = q3 + (1.5 * iqr)

# drop the outliers
traffic_clean = traffic[(traffic['Time on Page'] >= lower_bound) & (traffic['Time on Page'] <= upper_bound)]


# COLUMN: Previous Visits

# identify the quartiles
q1, q3 = np.percentile(traffic['Previous Visits'], [25, 75])

# calculate the interquartile range
iqr = q3 - q1

# calculate the lower and upper bounds
lower_bound = q1 - (1.5 * iqr)
upper_bound = q3 + (1.5 * iqr)

# drop the outliers
traffic_clean = traffic[(traffic['Previous Visits'] >= lower_bound) & (traffic['Previous Visits'] <= upper_bound)]



# COLUMN: Conversion Rate

# identify the quartiles
q1, q3 = np.percentile(traffic['Conversion Rate'], [25, 75])

# calculate the interquartile range
iqr = q3 - q1

# calculate the lower and upper bounds
lower_bound = q1 - (1.5 * iqr)
upper_bound = q3 + (1.5 * iqr)

# drop the outliers
traffic_clean = traffic[(traffic['Conversion Rate'] >= lower_bound) & (traffic['Conversion Rate'] <= upper_bound)]

# checking if outliershave been dropped successfully
print("Dropped outliers successfully!")
```

```
Dropped outliers successfully!
```

Untitled12.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text

Examining Correlations

[17]
```python
# dropping categorical column from outlier-free dataset
traffic_clean_num = traffic_clean.drop('Traffic Source', axis=1)
```

[18]
```python
# creating a heatmap and dropping CONVERSION RATE column in it as it can't be visualized
corr = traffic_clean_num.drop('Conversion Rate', axis=1).corr()
plt.figure(dpi=100)
sns.heatmap(corr, annot=True, fmt= '.2f')
plt.show()
```

Untitled12.ipynb

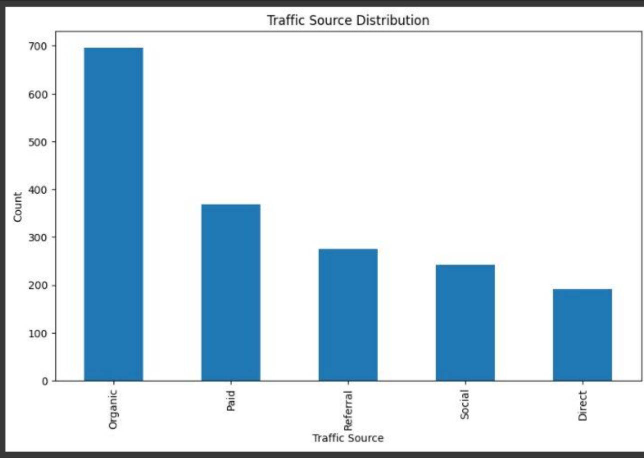
File Edit View Insert Runtime Tools Help

Commands + Code + Text

Data Visualization & Traffic Source Distribution

```
[19] # examining TRAFFIC SOURCE value counts
     traffic_counts = traffic_clean['Traffic Source'].value_counts()
```

```
[20] # creating a bar graph
     traffic_counts.plot(kind='bar', figsize=(10, 6))
     plt.title('Traffic Source Distribution')
     plt.xlabel('Traffic Source')
     plt.ylabel('Count')
     plt.show()
```



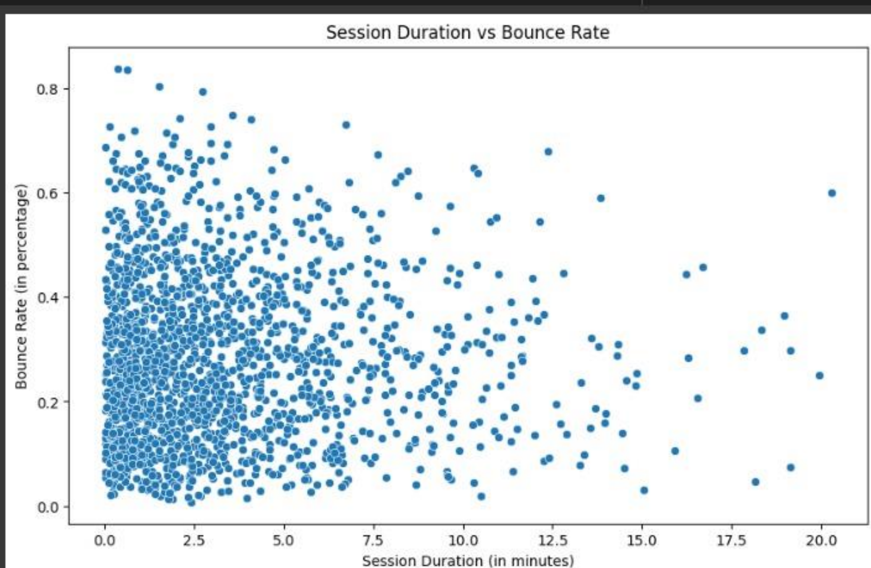
Untitled12.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text

Relationship Between Session Duration And Bounce Rate

```
[21] # creating a scatter plot
     plt.figure(figsize=(10, 6))
     sns.scatterplot(x='Session Duration', y='Bounce Rate', data=traffic_clean_num)
     plt.title('Session Duration vs Bounce Rate')
     plt.xlabel('Session Duration (in minutes)')
     plt.ylabel('Bounce Rate (in percentage)')
     plt.show()
```



Untitled12.ipynb

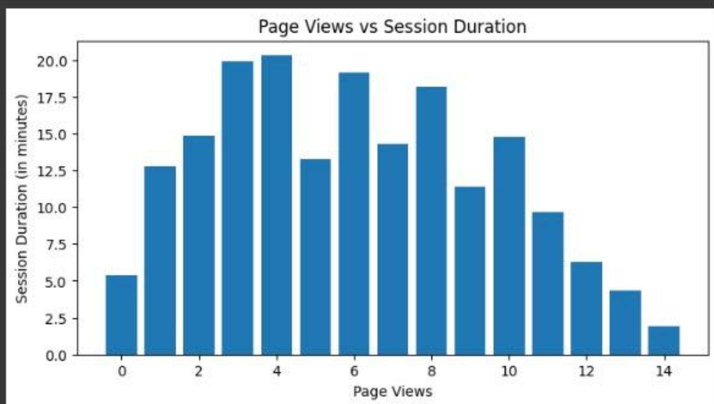
File Edit View Insert Runtime Tools Help

Commands + Code + Text

Relationship Between Page Views And Session Duration

[22]
```
# creating new variables for data visualization
page_views = traffic_clean_num['Page Views']
session_duration = traffic_clean_num['Session Duration']
```

[23]
```
# creating a bargraph
plt.figure(figsize=(8, 4))
plt.bar(page_views, session_duration)
plt.title('Page Views vs Session Duration')
plt.xlabel('Page Views')
plt.ylabel('Session Duration (in minutes)')
plt.show()
```



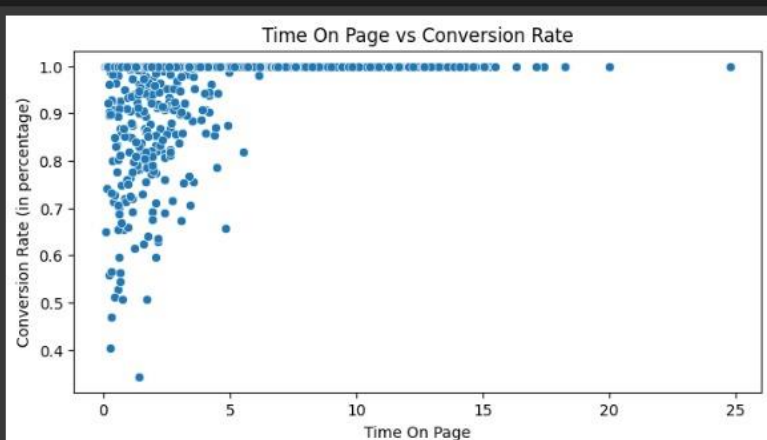
Untitled12.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text

Relationship Between Time On Page And Conversion Rate

[24]
```
# creating a scatter plot
plt.figure(figsize=(8,4))
sns.scatterplot(x='Time on Page', y='Conversion Rate', data=traffic)
plt.title('Time On Page vs Conversion Rate')
plt.xlabel('Time On Page')
plt.ylabel('Conversion Rate (in percentage)')
plt.show()
```



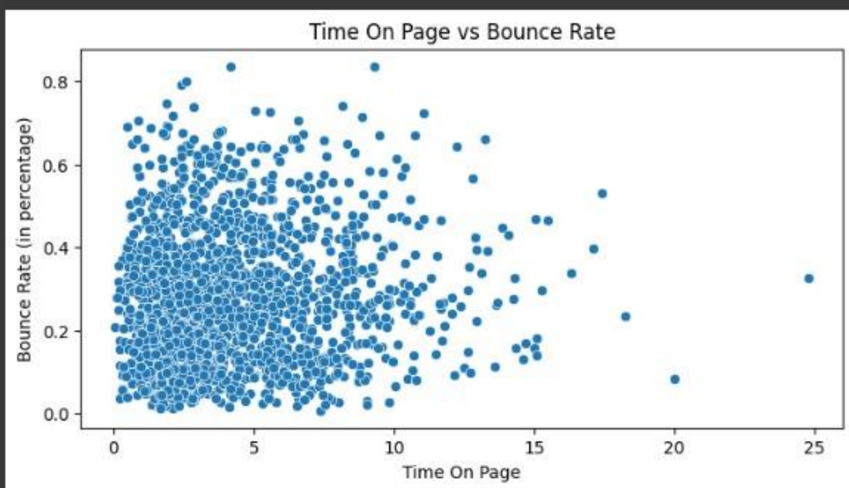
Untitled12.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text

Relationship Between Time On Page And Bounce Rate

[25]
```
# creating a scatter plot
plt.figure(figsize=(8,4))
sns.scatterplot(x='Time on Page', y='Bounce Rate', data=traffic_clean_num)
plt.title('Time On Page vs Bounce Rate')
plt.xlabel('Time On Page')
plt.ylabel('Bounce Rate (in percentage)')
plt.show()
```



Untitled12.ipynb

File Edit View Insert Runtime Tools Help

Commands + Code + Text

Conclusion

The dataset 'Website Traffic' is about 'Website Traffic and User Engagement Metrics'. As per the data source, the data is genrated and is not from a real website. The dataset have: Total Rows: 2000 Total Columns: 7 Categorical Column(s): 1 Numerical Column(s): 6 There are user engagement metrices of 1999 users. The dataset doesn't have any null values. The dataset have following datatypes respectively: Page Views int64 Session Duration float64 Bounce Rate float64 Traffic Source object Time on Page float64 Previous Visits int64 Conversion Rate float64 The dataset is a well-maintained and clean, thus, it doesn't require much data cleaning. The maximum number of pages viewd during a session are 14, whereas the least is 0 pages. There are 5 types of Traffic Sources in the dataset. 1773 users, out of 1999, who converted from a vistor to a buyer. There were several outliers in all the numerical columns, which were removed. The status of correlations between various variables is as follows: There is zero correlation between Session Duration and Bounce Rate. There is no very strong positive or very strong negative correlation in the entire dataset. Most of the traffic on website is 'Organic' and least traffic is 'Direct'. The 'Relationship Between Session Duration And Bounce Rate' data visualization doesn't convey any insights as there is no correlation between the two at all. The 'Relationship Between Time On Page And Bounce Rate' data visualization doesn't convey any insights as there is no important correlation between the two. The 'Relationship Between Page Views and Session Duration' data visualization doesn't convey any insights as there is a very random pattern of trends in it. As per the 'Relationship Between Time On Page And Conversion Rate' dataviz, users who are on a specific website page for 5 to 15 minutes are most likely to be converted from a visitor to a buyer. Next Steps The EDA of 'Website Traffic' suggested that subsequent data analysis or model development should consider:

Investigating dataset integrity for causations and duplicates. Assessing column proportionality. Identifying and handling unwanted observations. Selecting appropriate features and target variables. Choosing suitable modeling algorithms. Scaling features (if necessary).