Traffic Sources:

Utilize pie charts or stacked bar charts to show the breakdown of traffic sources (e.g., direct, organic search, social, referral).

Python code:

```
import pandas as pd

import matplotlib.pyplot as plt

data = pd.read_csv('web_traffic_data.csv')

data['timestamp'] =    pd.to_datetime(data['timestamp']) data.set_index('timestamp', inplace=Truedaily_traffic = data.resample('D').agg({'page_views': 'sum', 'unique_visitors': 'sum'})

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

plt.plot(daily_traffic.index, daily_traffic['page_views'], label='Page Views')

plt.title('Daily Page Views Over Time')

plt.xlabel('Date')

plt.ylabel('Page Views')

plt.legend()

plt.grid(True)

rolling_avg = daily_traffic['page_views'].rolling(window=7).mean()

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

plt.plot(daily_traffic.index, daily_traffic['page_views'], label='Page Views')

plt.plot(daily_traffic.index, rolling_avg, label='7-Day Rolling Average', color='orange')

plt.title('Daily Page Views and Rolling Average')

plt.xlabel('Date')

plt.ylabel('Page Views')

plt.legend()

plt.grid(True)

high_engagement_users = data[data['time_on_page'] > 300] low_bounce_rate_users = data[data['bounce_rate'] < 40]

from sklearn.linear_model import LinearRegression

X=daily_traffic.index.to_julian_date().values.reshape(-1, 1)
```

```
y = daily_traffic['page_views'].values model = LinearRegression()
model.fit(X, y)
future_dates = pd.date_range(start=data.index.max(), periods=7)
future_dates_julian = future_dates.to_julian_date().values.reshape  -1, 1)
predicted_page_views = model.predict(future_dates_julian)
plt.figure(figsize=(12, 6))
plt.plot(daily_traffic.index, daily_traffic['page_views'], label='Page Views')
plt.plot(future_dates, predicted_page_views, label='Predicted Page Views', color='red', linestyle='--')
plt.title('Web Traffic Prediction with Linear Regression')
plt.xlabel('Date')
plt.ylabel('Page Views')
plt.legend()
plt.grid(True)
plt.show()
```