**Website Traffic Analysis Report**

**1. Executive Summary**

In this comprehensive website traffic analysis report, we aim to provide a deep understanding of our website's performance and user engagement. By analyzing the dataset from Kaggle, we have uncovered key insights into user behavior and website trends. Our analysis is crucial for making data-driven decisions and optimizing the website's performance.

**2. Introduction**

This analysis is focused on extracting insights from a dataset containing daily website visitor information. The dataset, sourced from Kaggle, encompasses crucial metrics such as page loads, unique visits, first-time visits, and returning visits. By conducting this analysis, we can make informed decisions to enhance the website's effectiveness.

**3. Data Collection and Preprocessing**

The dataset was collected from Kaggle and imported into our data analysis environment. Prior to analysis, we conducted thorough data preprocessing, which involved handling missing values, cleaning the data, and ensuring data consistency. By preparing the dataset, we ensured the accuracy of our analysis.

**4. Exploratory Data Analysis (EDA)**

Our initial data exploration unveiled a wealth of information. We conducted a summary analysis to identify key metrics such as daily page loads, unique visits, first-time visits, and returning visits. We also explored trends in daily website traffic, examined the day of the week with the highest traffic, and performed a comprehensive time series analysis. Additionally, we analyzed the popularity of pages and the sources of traffic.

**5. Time Series Analysis**

Building upon the EDA, we conducted a more in-depth time series analysis. Our objective was to identify underlying trends and seasonality in the data. We also employed forecasting techniques to predict future website traffic, which can serve as a valuable tool for planning and resource allocation.

**6. User Segmentation**

Our analysis extended to segmenting users based on their behavior and demographics. These segments included new users and returning visitors. Analyzing different user segments enabled us to tailor our approach to user engagement, ensuring a more personalized experience.

**7. Machine Learning-Based Predictions**

We harnessed the power of machine learning to predict website traffic metrics. By training and evaluating machine learning models, we made predictions for future traffic. This allows us to anticipate website traffic and take proactive measures to optimize the user experience.

**8. IBM Cognos Interactive Dashboards**

To enhance the accessibility and comprehensibility of our findings, we utilized IBM Cognos to create interactive dashboards. These dashboards cover an array of essential insights, including an in-depth analysis of popular pages, an examination of traffic sources, and user engagement metrics.

**9. Results and Insights**

Our analysis yielded several key findings. We uncovered trends in user behavior, identified the most popular pages, and assessed the various sources of website traffic. These findings have substantial implications for website optimization and user engagement.

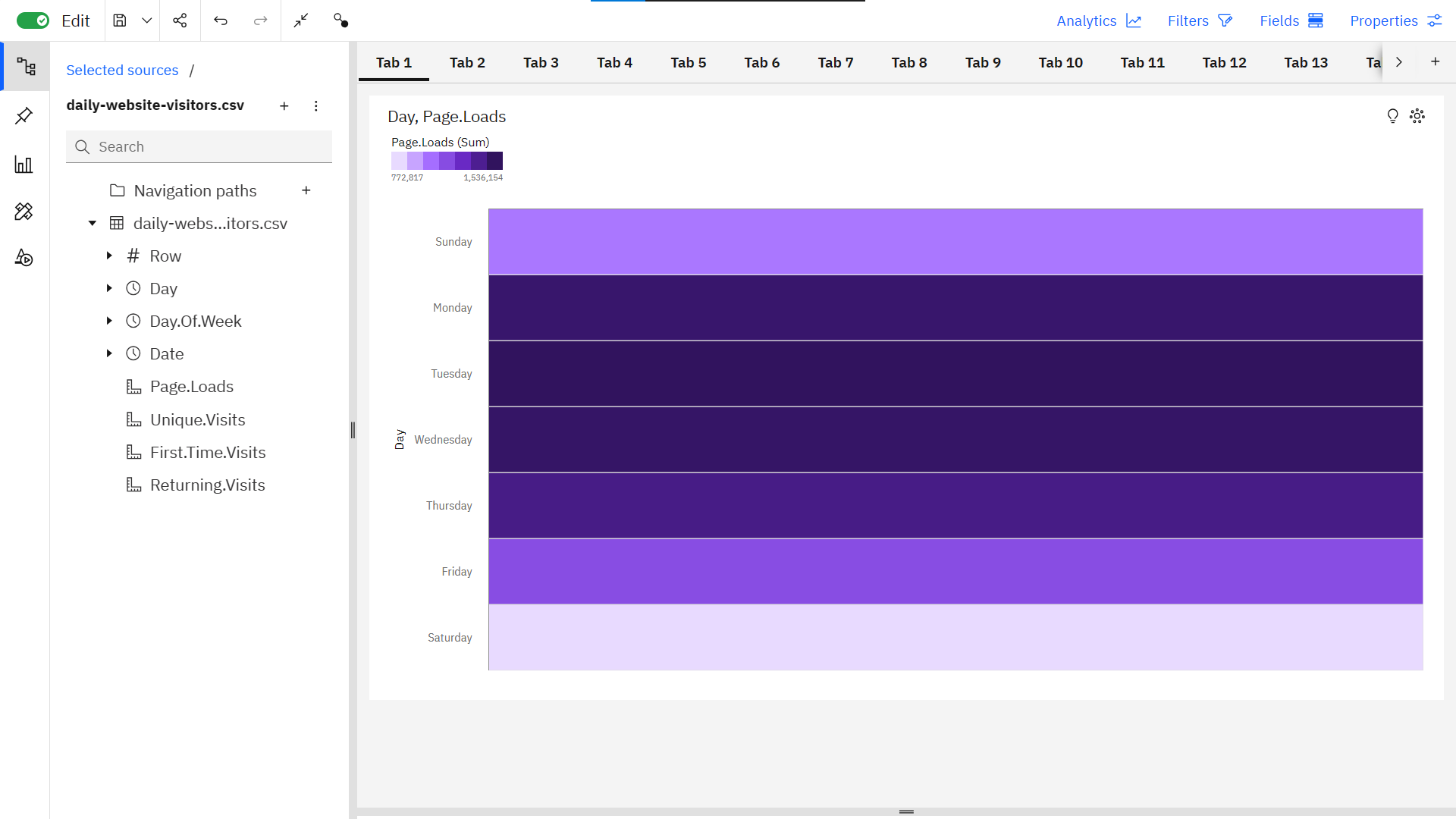
**10. Conclusion**

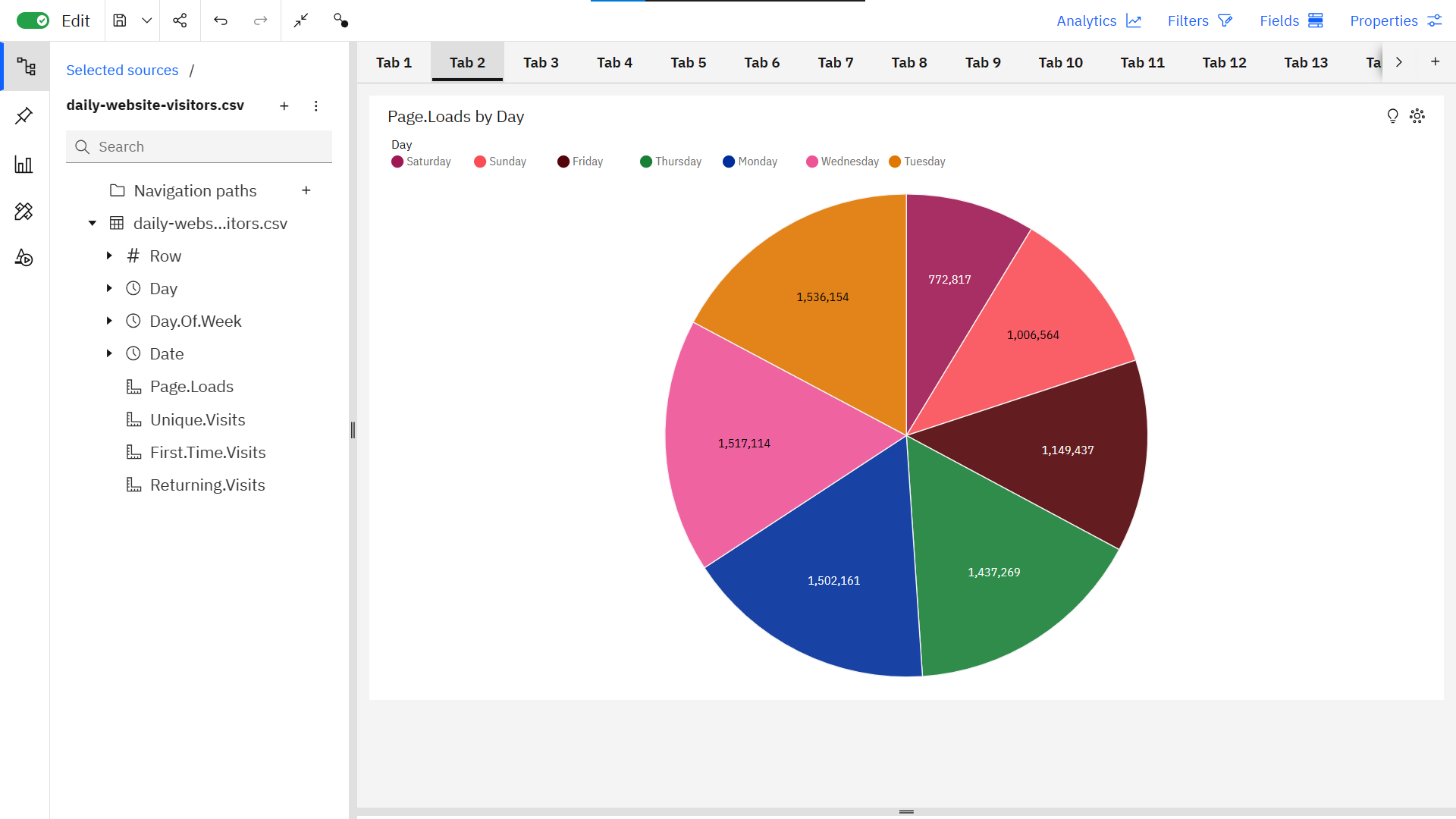
In conclusion, our analysis provides a valuable roadmap for improving website performance. By understanding user behavior and capitalizing on the popularity of certain pages and traffic sources, we can make data-driven decisions to optimize the website.

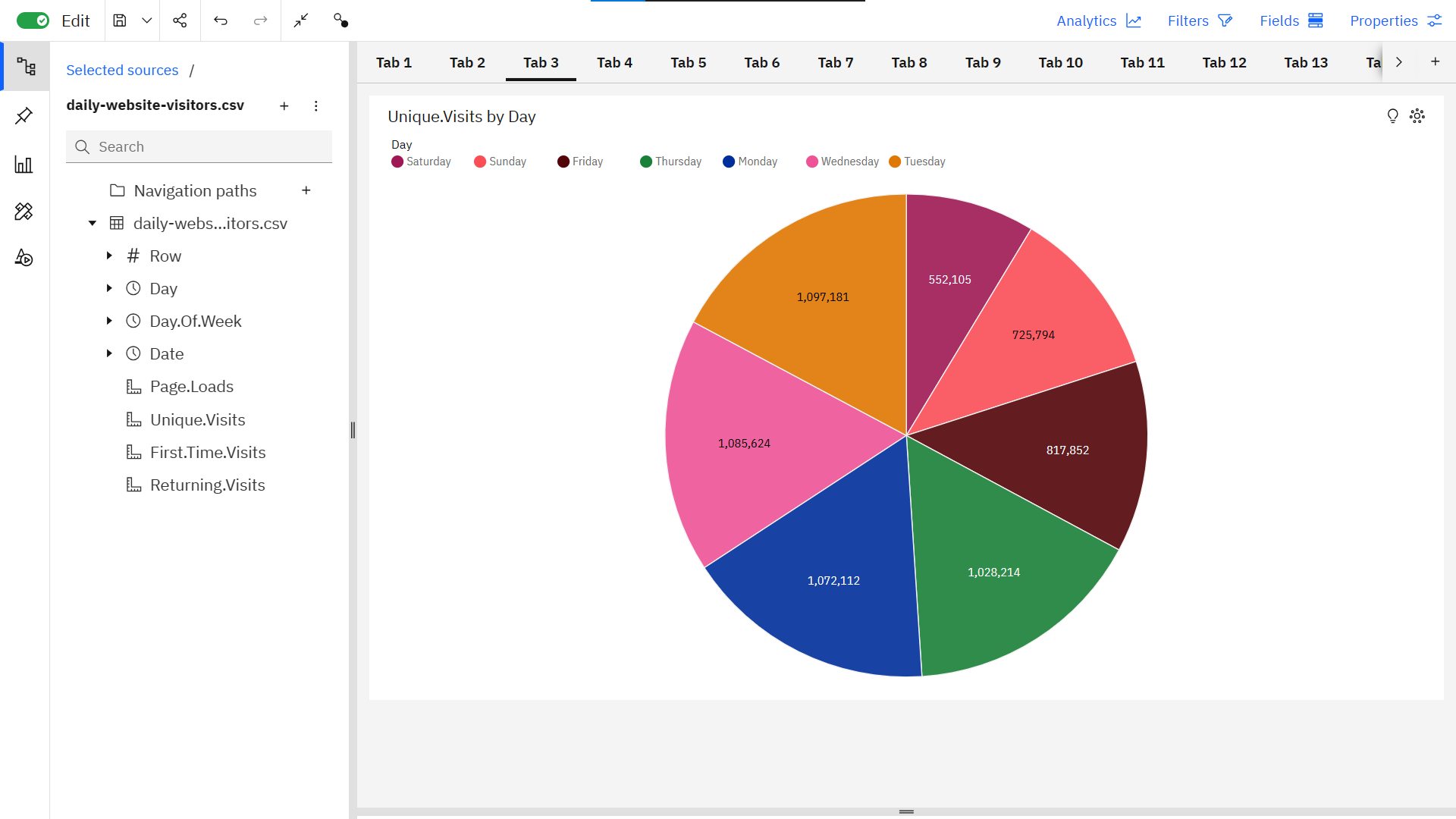
**11. Recommendations**

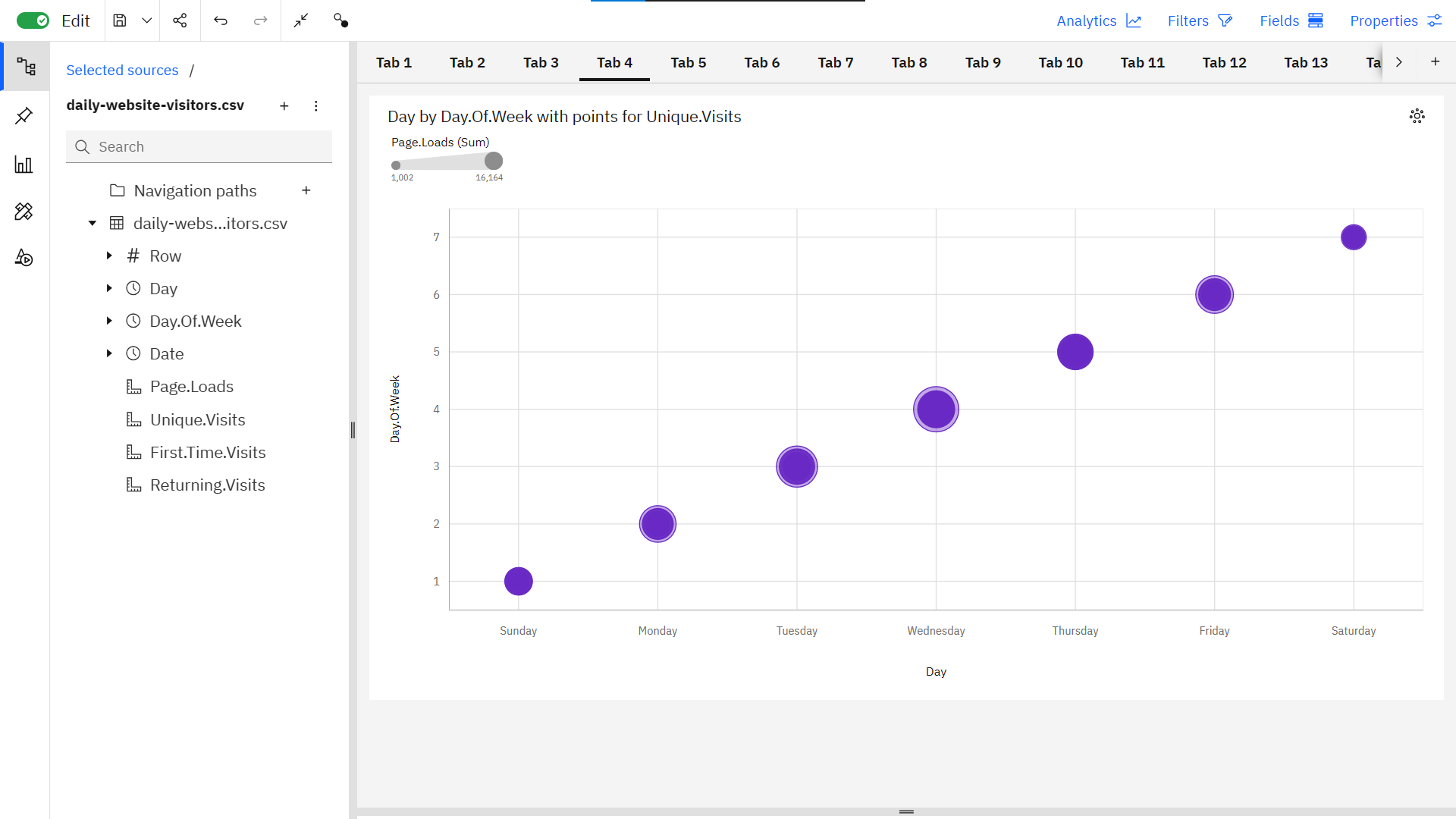
Based on our analysis, we recommend implementing a series of strategies to enhance website performance and user engagement. These strategies may involve optimizing popular pages and tailoring content to target specific traffic sources.

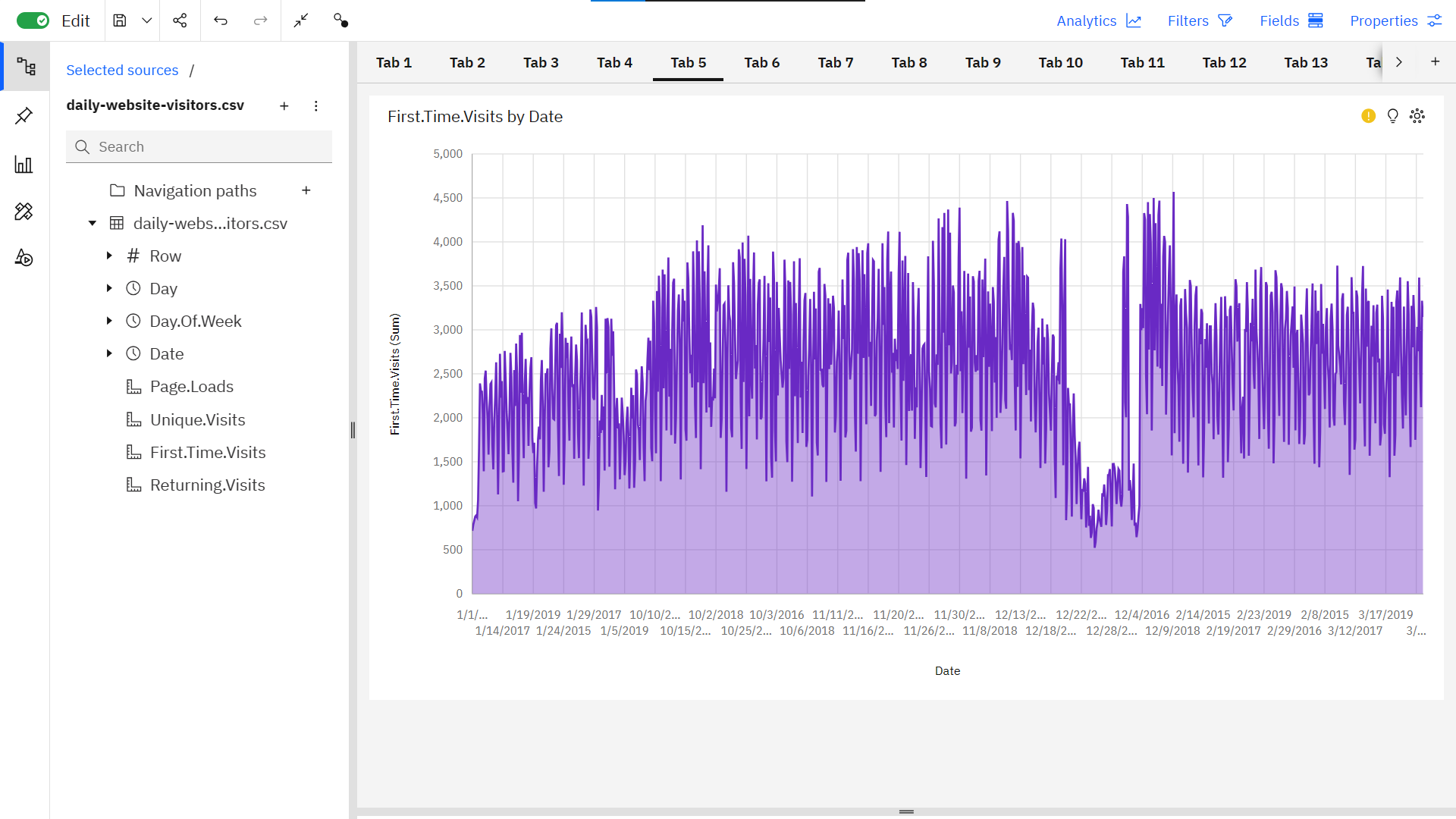
**IBM COGNOS ANALYSIS**











**PYTHON CODE**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns # Import Seaborn

# Load the dataset

df = pd.read\_csv("cleaned\_data.csv")

# Set Seaborn style

sns.set\_style("whitegrid")

# Data Exploration

print(df.head())

print(df.isnull().sum())

print(df.describe())

# Data Visualization using Seaborn

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

sns.lineplot(data=df, x='Day', y='Page.Loads', label='Page Loads')

plt.title('Page Loads Over Time')

plt.xlabel('Day')

plt.ylabel('Page Loads')

plt.show()

# Time Series Analysis - 7-Day Moving Average

df['7-Day Moving Avg'] = df['Page.Loads'].rolling(window=7).mean()

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

sns.lineplot(data=df, x='Day', y='7-Day Moving Avg', label='7-Day Moving Avg')

sns.lineplot(data=df, x='Day', y='Page.Loads', label='Page Loads')

plt.title('Page Loads and 7-Day Moving Average')

plt.xlabel('Day')

plt.ylabel('Page Loads')

plt.legend()

plt.show()

# Calculate the correlation matrix

correlation\_matrix = df.corr()

# Create a heatmap using Seaborn

plt.figure(figsize=(10, 8))

sns.heatmap(correlation\_matrix, annot=True, cmap="coolwarm")

plt.title("Correlation Heatmap")

plt.show()

**OUTPUT**

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

0 1 Sunday 1 2014-09-14 2146 1582

1 2 Monday 2 2014-09-15 3621 2528

2 3 Tuesday 3 2014-09-16 3698 2630

3 4 Wednesday 4 2014-09-17 3667 2614

4 5 Thursday 5 2014-09-18 3316 2366

First.Time.Visits Returning.Visits

0 1430 152

1 2297 231

2 2352 278

3 2327 287

4 2130 236

Row 0

Day 0

Day.Of.Week 0

Date 0

Page.Loads 0

Unique.Visits 0

First.Time.Visits 0

Returning.Visits 0

dtype: int64

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

count 2167.000000 2167.000000 2167.000000 2167.000000

mean 1084.000000 3.997231 4116.989386 2943.646516

std 625.703338 2.000229 1350.977843 977.886472

min 1.000000 1.000000 1002.000000 667.000000

25% 542.500000 2.000000 3114.500000 2226.000000

50% 1084.000000 4.000000 4106.000000 2914.000000

75% 1625.500000 6.000000 5020.500000 3667.500000

max 2167.000000 7.000000 7984.000000 5541.000000

First.Time.Visits Returning.Visits

count 2167.000000 2167.000000

mean 2431.824181 511.822335

std 828.704688 168.736370

min 522.000000 133.000000

25% 1830.000000 388.500000

50% 2400.000000 509.000000

75% 3038.000000 626.500000

max 4616.000000 1036.000000

