WEBSITE TRAFFIC ANALYSIS
DEVELOPMENT PART-2

Table of Contents:

1. INTRODUCTION

Website traffic analysis is an essential component of any online business strategy. It provides valuable insights into how users interact with a website, helping organizations make data-driven decisions.

In this document, we will discuss three key aspects of website analytics: Time Series Analysis, User Segmentation, and Machine Learning (ML)-based Prediction.

2. TIME SERIES ANALYSIS

What is Time Series Analysis?

Time Series Analysis involves examining and interpreting data points collected over time.

In the context of website traffic analysis, it focuses on understanding the patterns, trends, and seasonality of website visitors and their behaviour.

Why is Time Series Analysis Important?

- Identifying long-term trends: It helps in understanding whether your website's traffic is growing, declining, or stable over time.

- Seasonal trends: Recognizing periodic patterns, such as increased traffic holidays or weekends, can guide content and marketing strategies.

- Daily and weekly patterns: Studying how traffic fluctuates within a day or week can optimize ad placement and content scheduling.

Tools and Techniques:

Common tools for time series analysis include Python libraries like Pandas, Matplotlib, and Seaborn.

PYTHON CODE

```
import pandas as pd
import matplotlib.pyplot as plt
# Load our website traffic analysis dataset
data = pd.read_csv('/content/daily-website-visitors.csv')
# Convert the 'Date' column to a datetime object
data['Date'] = pd.to_datetime(data['Date'])
# Time Series Analysis
plt.figure(figsize=(12, 6))
plt.plot(data['Date'], data['Page.Loads'], label='Page
Views')
plt.plot(data['Date'], data['Unique.Visits'],
label='Unique Visitors')
plt.xlabel('Date')
plt.ylabel('Count')
plt.title('Website Traffic Over Time')
plt.legend()
plt.show()
```

OUTPUT

3. USER SEGMENTATION

What is User Segmentation?

User segmentation divides website visitors into distinct groups based on specific characteristics or behaviours. These segments enable more personalized marketing and content strategies.

Why is User Segmentation Important?

Personalization: Tailoring content and offers to different user groups increases engagement and conversions.

Targeted marketing: Segmented users allow for more efficient ad targeting, reducing ad spend.

Improved user experience: Knowing your audience better helps in creating a website that caters to their needs.

PYTHON CODE

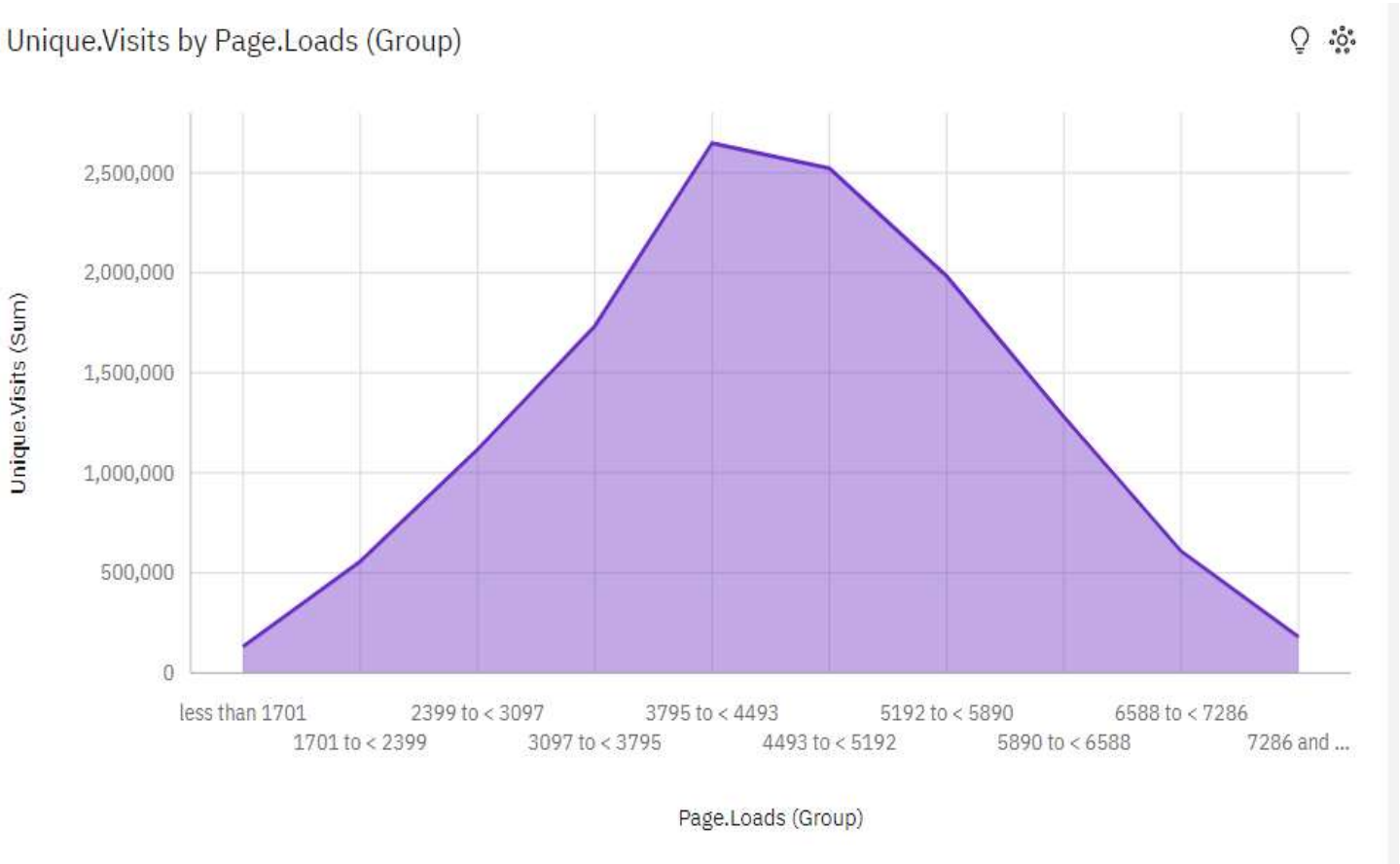
```
user_segments = pd.cut(data['Page.Loads'], bins=[0, 100,
1000, 10000, float('inf')],
                       labels=['Low', 'Moderate', 'High',
'Very High'])
data['user_segments'] = user_segments

# Grouping the dataset for user segment
segmented_data = data.groupby('user_segments').agg({
    'Page.Loads': 'sum',
    'Unique.Visits': 'sum'
})
```

Types of User Segmentation:

- Unique visitors
- First time visitors
- Returning visitors

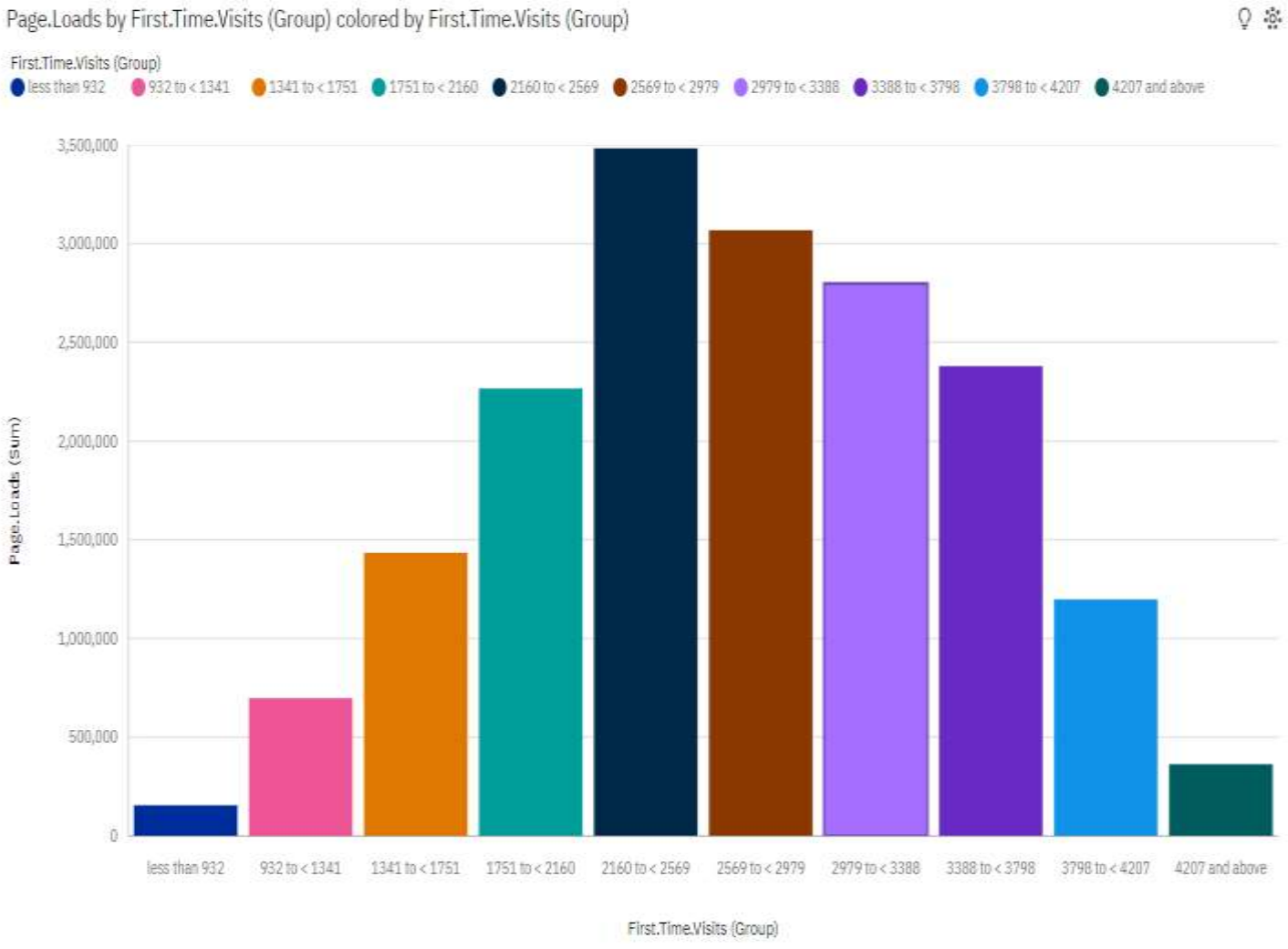
PAGE LOADS BY UNIQUE VISITORS



INSIGHTS:

- From **2020-05-04** to **2020-05-05**, **5890 to < 6588**'s **Unique.Visits** increased by **115**%.
- Over all values of **Page.Loads (Group)**, the sum of **Unique.Visits** is **almost thirteen million**.
- **Unique.Visits** ranges from **almost 132 thousand**, when **Page.Loads (Group)** is **less than 1701**, to **over 2.6 million**, when **Page.Loads (Group)** is **3795 to < 4493**
- For **Unique.Visits**, the most significant values of **Page.Loads (Group)** are **3795 to < 4493** and **4493 to < 5192**, whose respective **Unique.Visits** values add up to **nearly 5.2 million**, or **40.5** % of the total.

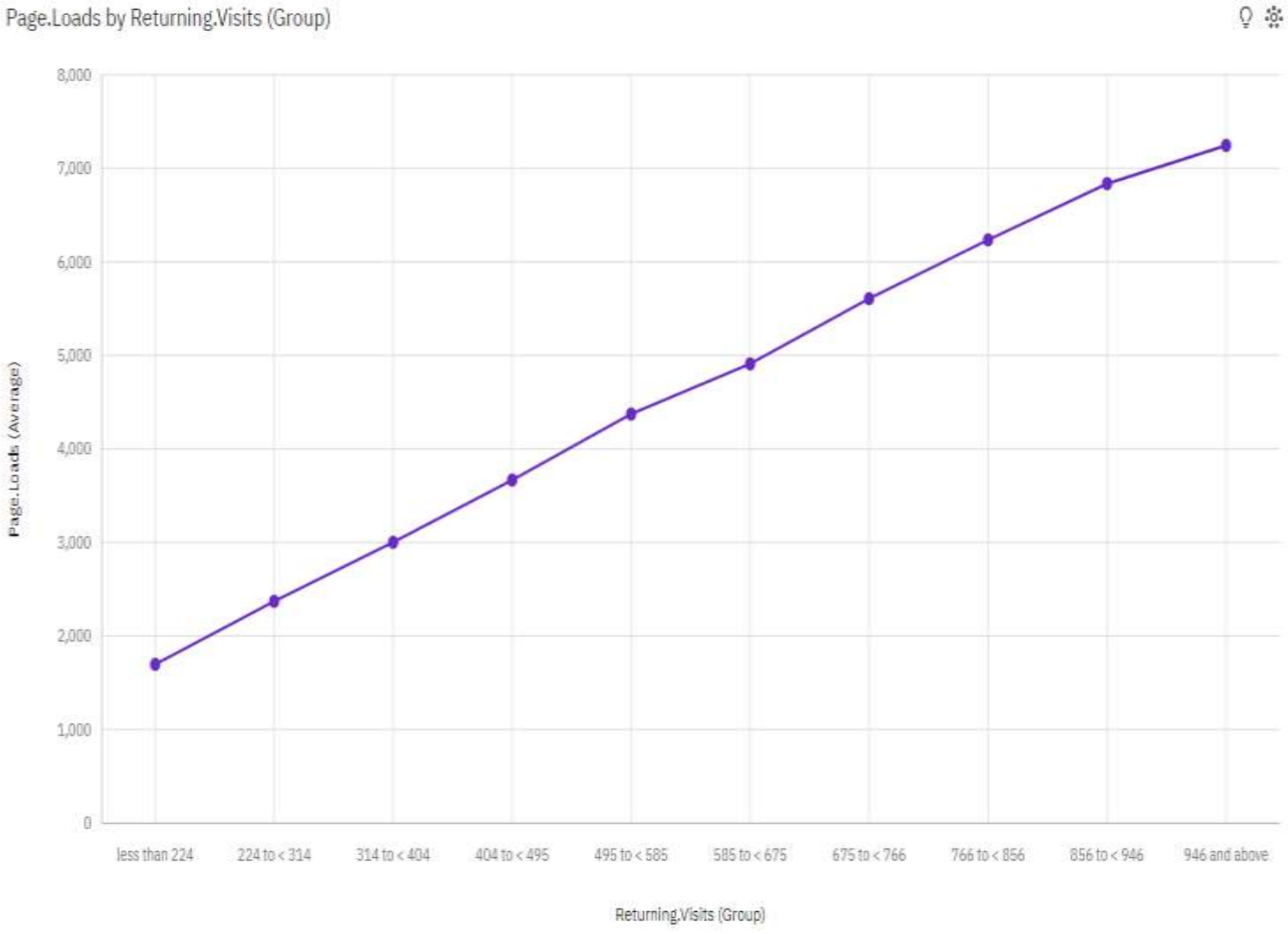
PAGE LOAD BY FIRST TIME VISITORS



INSIGHTS:

- From **2020-05-04** to **2020-05-05**, **3798 to < 4207**'s **Page.Loads** increased by **114**%.
- Over all values of **First.Time.Visits (Group)** and **First.Time.Visits (Group)**, the sum of **Page.Loads** is **nearly eighteen million**.
- The summed values of **Page.Loads** range from **over 155 thousand** to **nearly 3.5 million**
- For **Page.Loads**, the most significant values of **First.Time.Visits (Group)** are **2160 to < 2569**, **2569 to < 2979**, **2979 to < 3388**, **3388 to < 3798**, and **1751 to < 2160**, whose respective **Page.Loads** values add up to **nearly fourteen million**, or **78.4** % of the total.

PAGE LOAD BY RETURNING VISITORS



INSIGHTS:

- From **2018-10-07** to **2018-10-08**, **585 to < 675**'s **Page.Loads** increased by **48**%.
- **Returning.Visits (Group)** strongly affects **Page.Loads** (**81%**)
- **495 to < 585** (**19.9** %), **404 to < 495** (**19.1** %), **585 to < 675** (**15.9** %), **314 to < 404** (**13.5** %), and **675 to < 766** (**10.8** %) are the most frequently occurring categories of **Returning.Visits (Group)** with a combined count of **3432** items with **Page.Loads** values (**79.2** % of the total) .
- Across all values of **Returning.Visits (Group)**, the average of **Page.Loads** is **over four thousand**.
- The average values of **Page.Loads** range from **over 1500**, occurring when **Returning.Visits (Group)** is **less than 224**, to **over seven thousand**, when **Returning.Visits (Group)** is **946 and above**.

4. MACHINE LEARNING-BASED PREDICTION

What is ML-Based Prediction?

Machine learning models can forecast website traffic and user behaviour based on historical data.

Predictions can range from daily page views to conversion rates.

Why is ML-Based Prediction Important?

Data-driven decisions: Predictive models assist in optimizing marketing campaigns and resource allocation.

Anticipating traffic spikes: Knowing when traffic will peak allows for server scaling and infrastructure planning.

Personalized recommendations: Predictive models can suggest content or products to users, increasing engagement.

Types of Predictive Models:

- **Regression:** For predicting numerical values like page views or revenue.
- **Classification:** For categorizing users into segments or predicting their likelihood to convert.
- **Time Series Forecasting:** Specialized models for predicting time-related data.

PYTHON CODE

```
import pandas as pd
import matplotlib.pyplot as plt

# Load the website traffic analysis dataset
data = pd.read_csv('/content/daily-website-visitors.csv')

# Convert the 'date' column to a datetime object
data['Date'] = pd.to_datetime(data['Date'])

# Remove commas and convert to integers
data['Unique.Visits'] =
data['Unique.Visits'].str.replace(',', '').astype(int)
data['Page.Loads'] = data['Page.Loads'].str.replace(',',
'').astype(int)

# Machine Learning-based Predictions (Regression Example)
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

# Prepare features and target
X = data[['Unique.Visits']]
y = data['Page.Loads']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Create and train a linear regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions
```

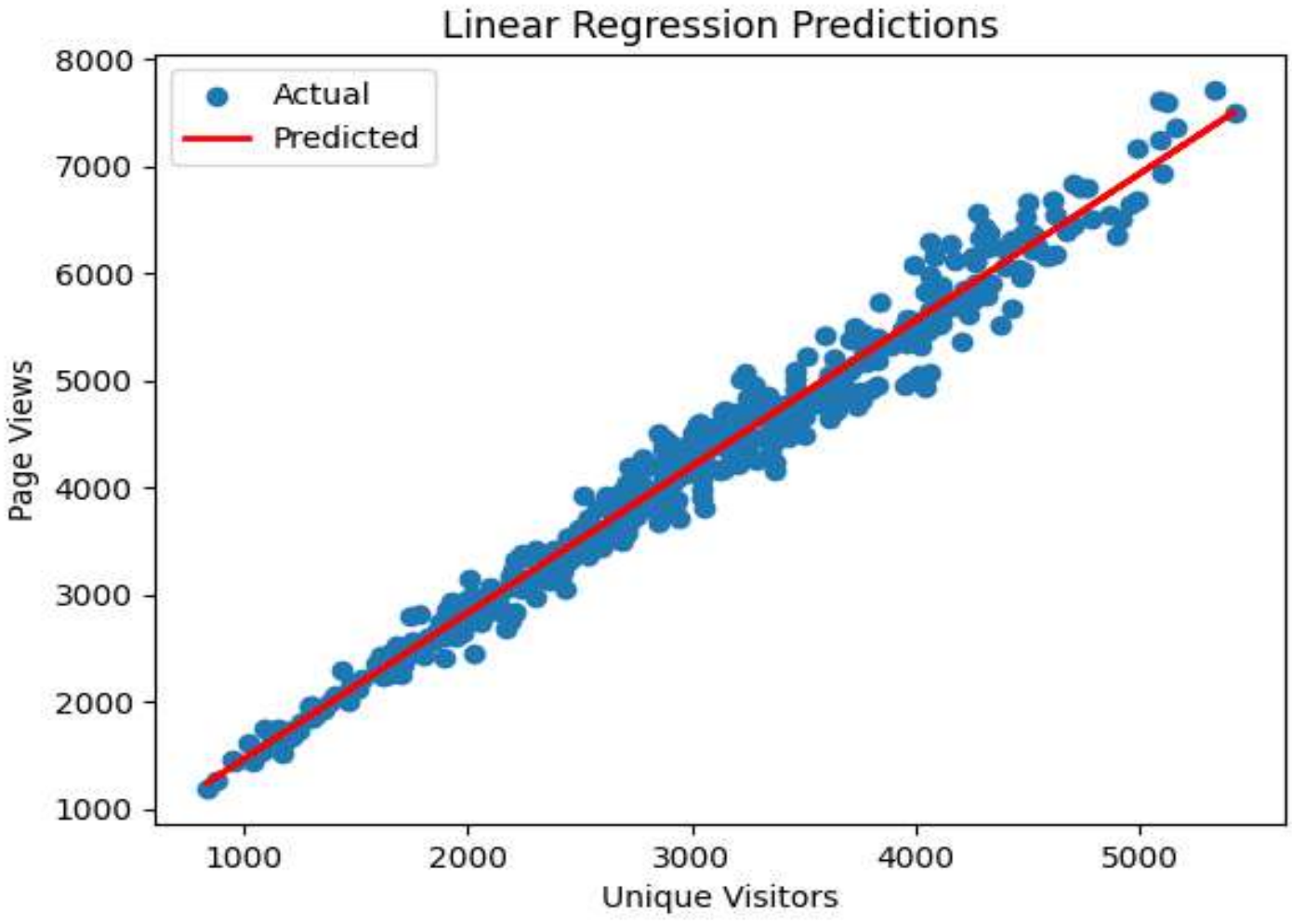
```
y_pred = model.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
print(f'Mean Squared Error: {mse}')

# Visualize the regression results
plt.scatter(X_test, y_test, label='Actual')
plt.plot(X_test, y_pred, color='red', linewidth=2,
label='Predicted')
plt.xlabel('Unique Visitors')
plt.ylabel('Page Views')
plt.legend()
plt.title('Linear Regression Predictions')
plt.show()
```

OUTPUT

```
Mean Squared Error: 45275.177119579836
```



PYTHON CODE

```
def prob(t, n, lmbda):
    return math.pow(lmbda * t,
n)/math.factorial(n)*math.exp(-lmbda*t)

mean = df['Page.Loads'].mean()
print( "mean loads per day:", mean)

std = df['Page.Loads'].std()
print( "std deviation of loads per day:", std)

n = 1
px = np.linspace(1, 8000, 50)
py = np.zeros(50)
for i in range(0, 50):
    x = (px[i]-mean)/std
    p = norm.pdf(x)
    py[i] = 1000*p
```

OUTPUT

```
mean loads per day: 4116.9893862482695
std deviation of loads per day: 1350.9778426999621
```

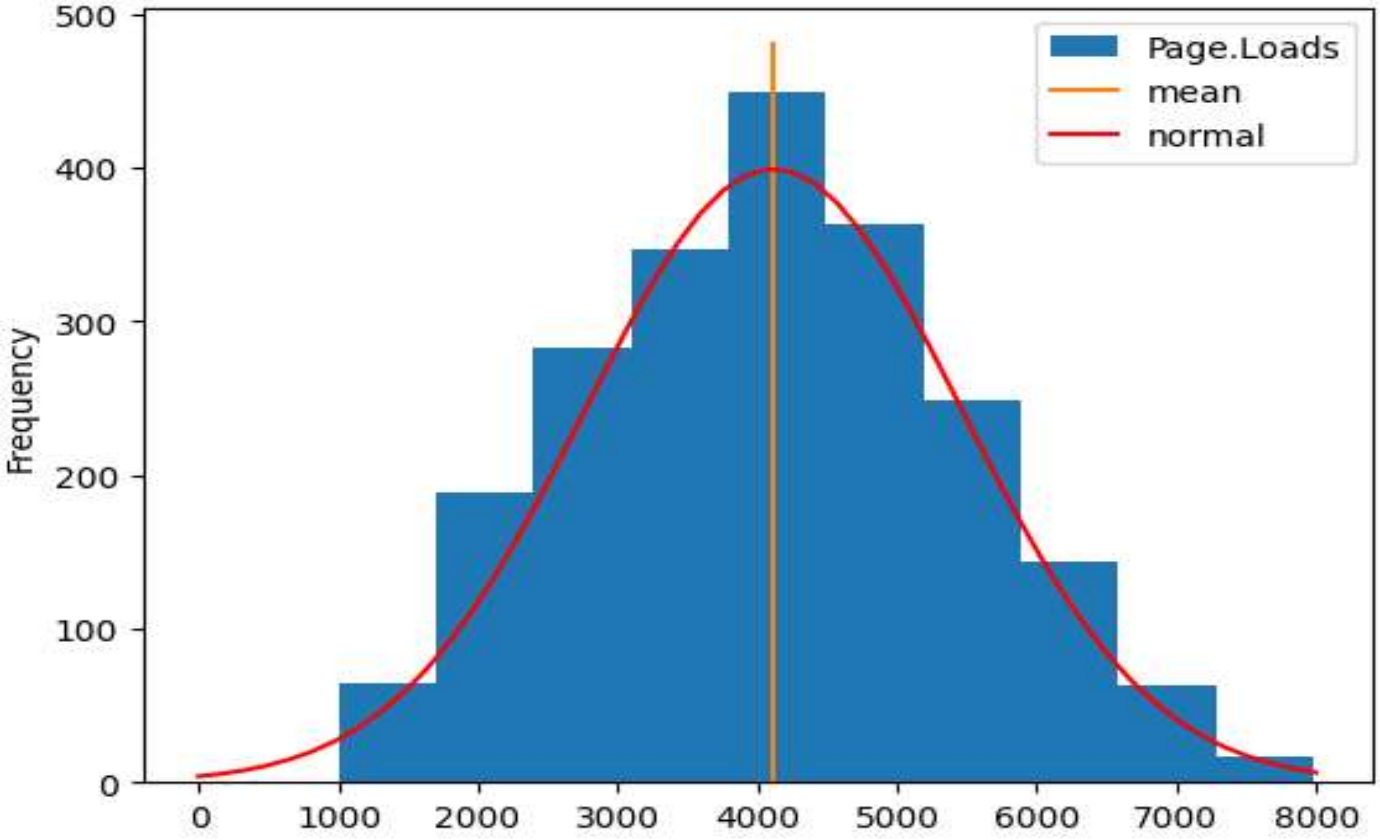
PYTHON CODE

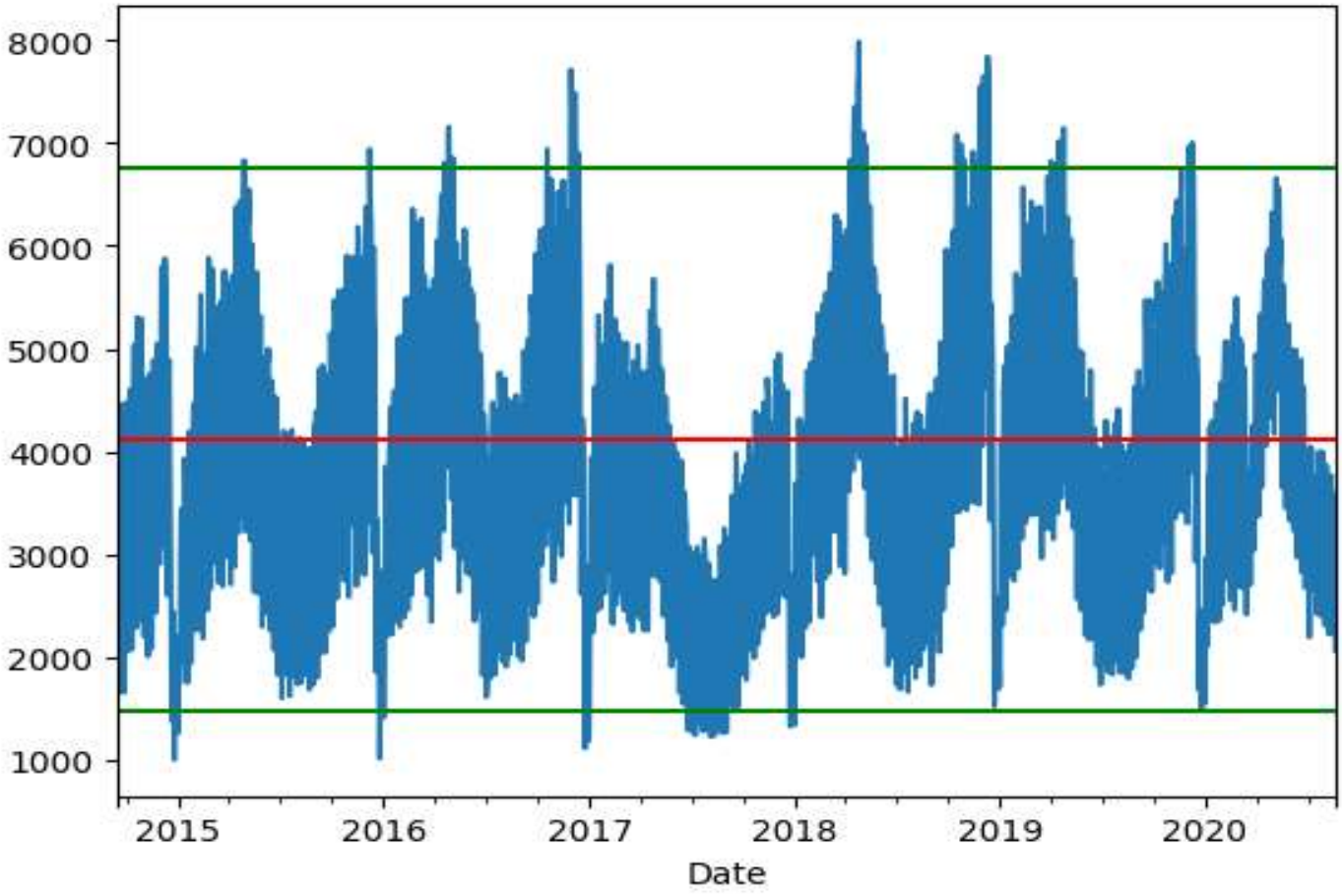
```
fig, ax1 = plt.subplots()
df['Page.Loads'].plot.hist(ax = ax1, label='Page.Loads')

plt.plot([mean, mean], [0, 480], label='mean')

plt.plot(px, py, label='normal', color='red')
plt.legend()
plt.show()
fig, ax1 = plt.subplots()
df['Page.Loads'].plot(ax = ax1, label='Page.Loads')
plt.plot([df.index[0], df.index[-1]], [mean, mean],
color='red')
upper = mean + 1.96*std
lower = mean - 1.96*std
plt.plot([df.index[0], df.index[-1]], [upper, upper],
color='green')
plt.plot([df.index[0], df.index[-1]], [lower, lower],
color='green')
plt.show()
```

OUTPUT


Page.Loads
mean
normal
Frequency
500
400
300
200
100
0
0
1000
2000
3000
4000
5000
6000
7000
8000


8000
7000
6000
5000
4000
3000
2000
1000
2015
2016
2017
2018
2019
2020
Date

PYTHON CODE

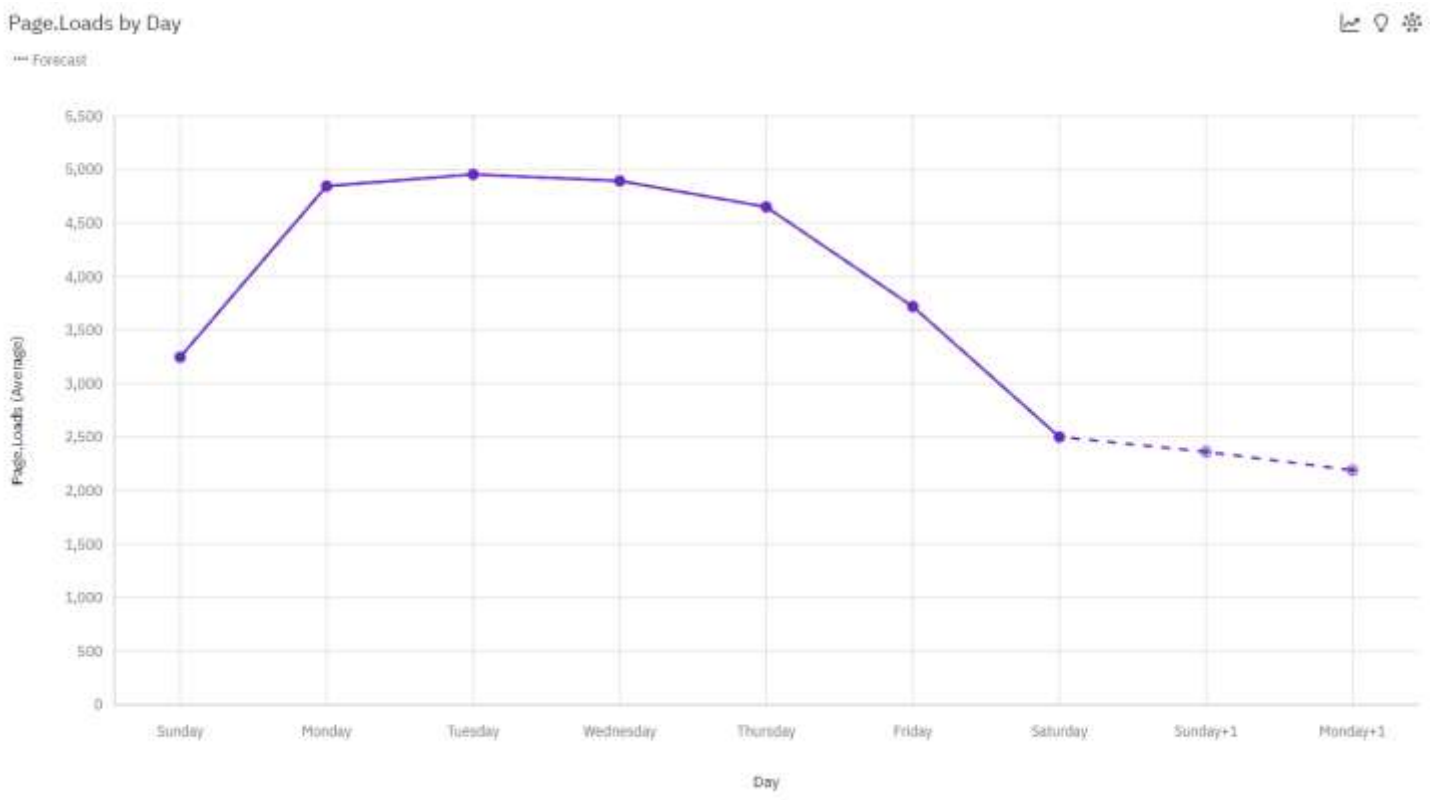
```
y_pred = model.predict([[5758]])
y_pred
```

OUTPUT

```
array([7963.52118177])
```

IBM COGNOS ANALYTICS

PAGE LOADS BY DAY

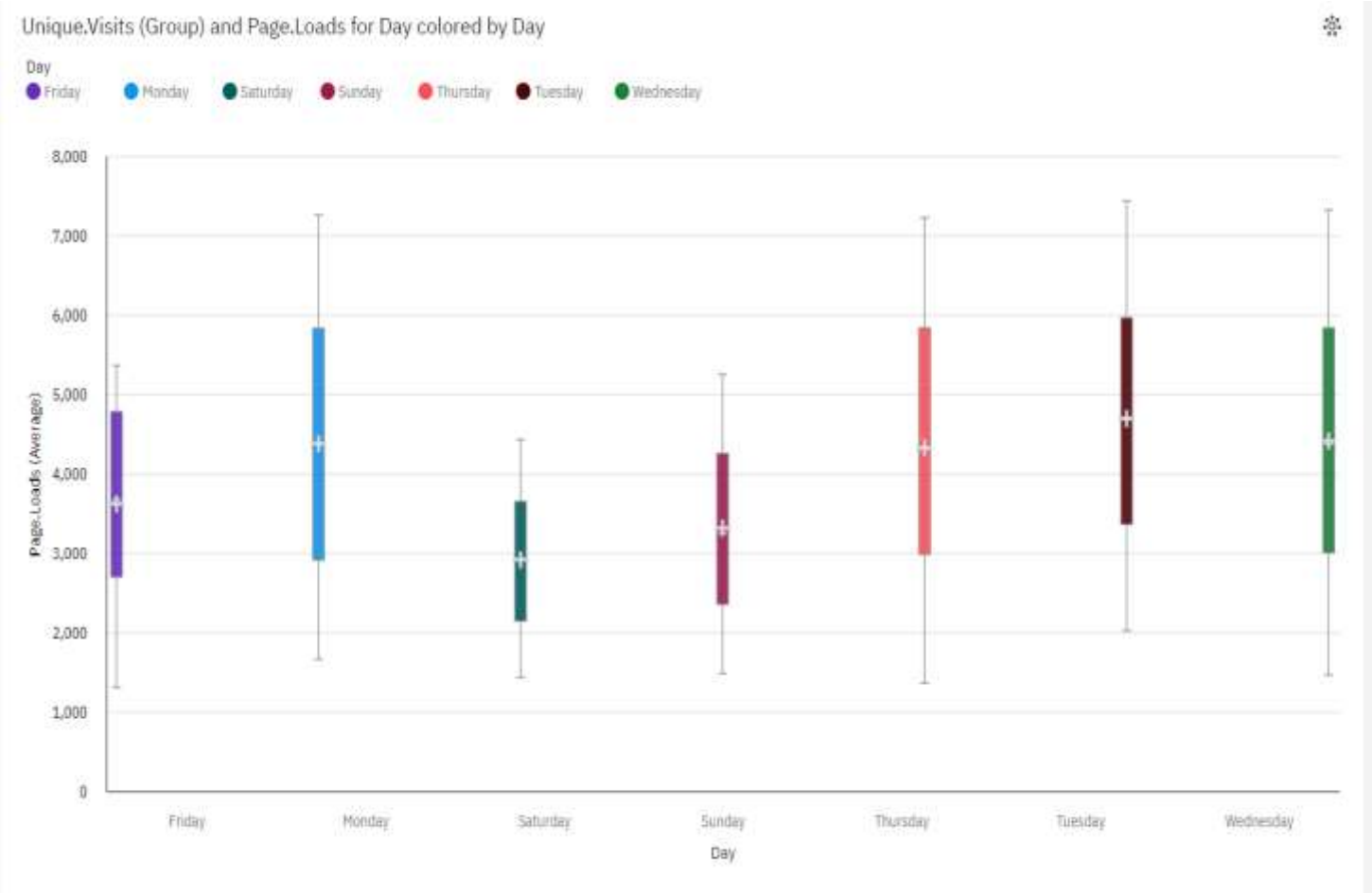


INSIGHTS:

- **Page.Loads** is unusually low when **Day** is **Saturday**.
- Based on the current forecasting, **Page.Loads** may reach **over two thousand** by **Day Monday+1**.
- **Day** moderately affects **Page.Loads** (**44%**)
- **Monday** (**14.3** %), **Sunday** (**14.3** %), **Wednesday** (**14.3** %), and **Tuesday** (**14.3** %) are the most frequently occurring categories of **Day** with a combined count of **2480** items with **Page.Loads** values (**57.2** % of the total).
- Across all **days**, the average of **Page.Loads** is **over four thousand**.

- The average values of **Page.Loads** range from **over 2500**, occurring when **Day** is **Saturday**, to **nearly five thousand**, when **Day** is **Tuesday**.

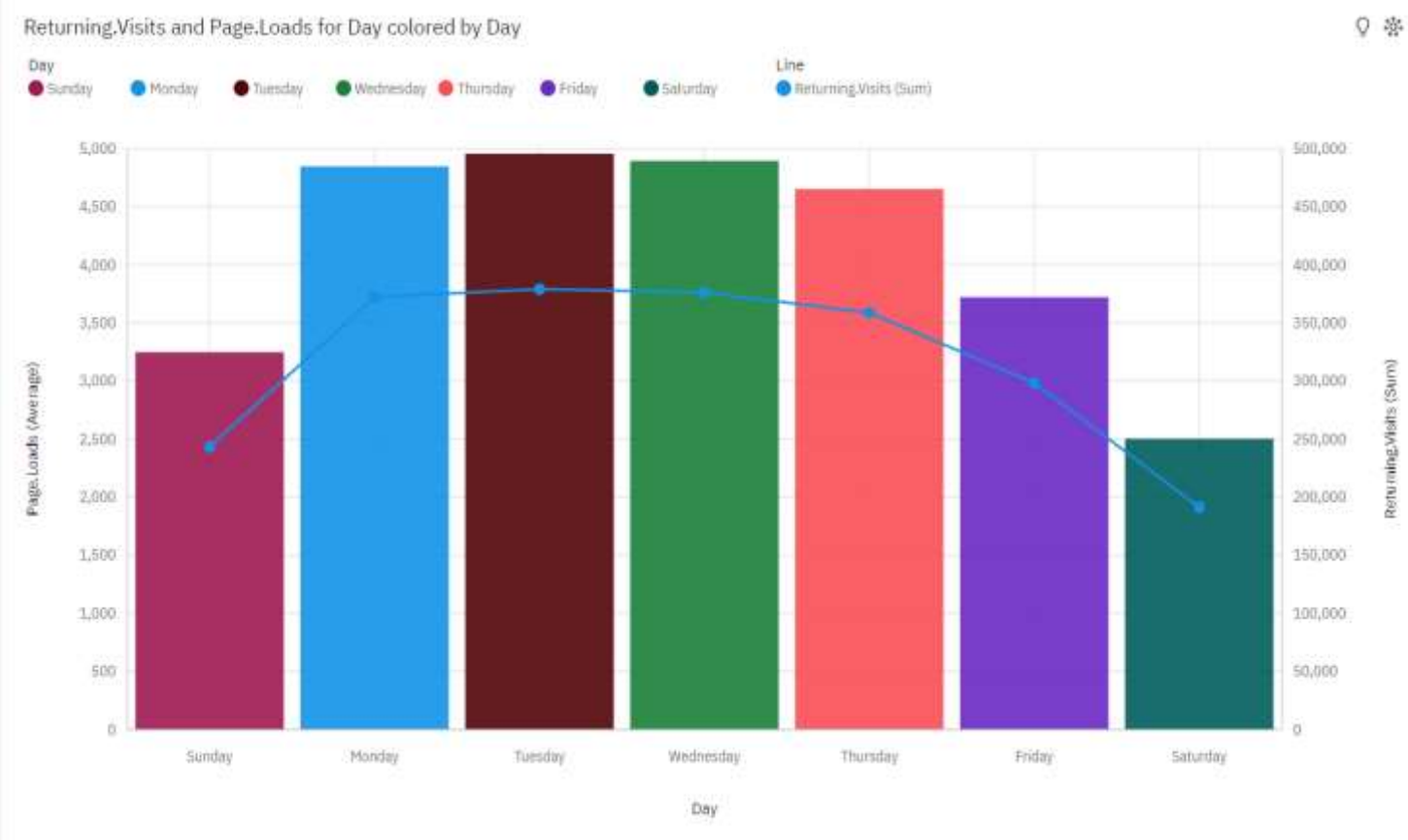
UNIQUE VISITOR AND PAGE LOADS BY DAY IN BOX PLOT



INSIGHTS:

- Based on the current forecasting, **Page.Loads** may reach **over two thousand** by **Day Monday+1**.
- Based on the current forecasting, **Page.Loads** may reach **over two thousand** by **Day Monday+1**.
- The overall number of results for **Page.Loads** is **nearly 4500**.

RETURNING VISITORS AND PAGE LOADS BY DAYS IN LINE AND COLUMN CHAT



INSIGHTS:

- **Page.Loads** is most unusual when the combinations of **Day** and **Day** are **Saturday and Saturday, Sunday and Sunday, Tuesday and Tuesday, Wednesday and Wednesday, Monday and Monday** and more.
- Based on the current forecasting, **Page.Loads** may reach **over two thousand** by **Day Monday+1**.
- **Day** moderately affects **Page.Loads** (**44%**).
- **Monday** (**14.3** %), **Sunday** (**14.3** %), **Wednesday** (**14.3** %) and **Tuesday** (**14.3** %) are the most frequently occurring categories of **Day** with a combined count of **2480** items with **Page.Loads** values (**57.2** % of the total).
- Across all **days** and **days**, the average of **Page.Loads** is **over four thousand**.
- The average values of **Page.Loads** range from **over 2500** to **nearly five thousand**.
- Based on the current forecasting, **Returning.Visits** may reach **over 174 ousand** by **Day Monday+1**.

- **Returning.Visits** ranges from **over 191 thousand**, when **Day** is **Saturday**, to **almost 379 thousand**, when **Day** is **Tuesday**.
- For **Returning.Visits**, the most significant values of **Day** are **Tuesday**, **Wednesday**, **Monday**, **Thursday**, and **Friday**, whose respective **Returning.Visits** values add up to **almost 1.8 million**, or **80.4** % of the total.

5. CONCLUSION

Website traffic analysis is a multifaceted endeavour, involving time series analysis, user segmentation, and machine learning-based prediction. When combined, these techniques can provide a comprehensive understanding of user behaviours, enabling organizations to make informed decisions, improve user experiences, and drive better results. Incorporating these insights into your website management and marketing strategies will help you stay competitive in the ever-evolving digital landscape.