D. Saul Jameson
https://www.linkedin.com/in/dsauljameson/

Project Documentation: Predicting Website Traffic Using Time Series Analysis

Project Overview

This project focuses on forecasting website traffic using time series analysis. The goal is to develop a predictive model that estimates future website visits based on historical traffic data. The model helps businesses and digital marketers optimize content strategy, server allocation, and ad spending by identifying trends, seasonality, and anomalies in visitor traffic.

Objective

- Develop a time series forecasting model to predict website traffic.
- Identify key patterns such as seasonality, trends, and anomalies in visitor behavior.
- Provide visual insights into traffic fluctuations using interactive charts.

Data Collection and Preprocessing

Data Source

Data Generation

The dataset for this project was **synthetically generated** to simulate realistic website traffic patterns. It includes daily visitor counts over multiple years, incorporating **seasonality, trends, and random fluctuations** to model real-world web traffic behavior.

Packages Used

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import plotly.express as px
import plotly.graph_objects as go
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.statespace.sarimax import SARIMAX
from sklearn.metrics import mean_absolute_error, mean_squared_error
```

D. Saul Jameson
https://www.linkedin.com/in/dsauljameson/

Exploratory Data Analysis (EDA)

To understand the trends in website traffic, we used **Plotly** to create interactive visualizations.

Traffic Over Time

A **line plot** was generated to visualize overall traffic trends and detect major fluctuations.

```
fig = px.line(df, x="date", y="visits", title="Website Traffic Over Time")
fig.show()
```

Seasonal Decomposition

A decomposition plot was created to break down traffic into **trend, seasonality, and residual components**.

```
result = seasonal_decompose(df["visits"], model="additive", period=365)
fig = result.plot()
plt.show()
```

Modeling Approach

1. Baseline Model (Naïve Forecast)

A simple **lagged model** was used as a benchmark where tomorrow's traffic is assumed to be the same as today's.

2. SARIMA Model

The **Seasonal AutoRegressive Integrated Moving Average (SARIMA)** model was chosen due to the presence of strong seasonal trends.

```
sarima_model = SARIMAX(df["visits"], order=(2,1,2), seasonal_order=(1,1,1,7))
sarima_result = sarima_model.fit()
```

Model Evaluation

Metrics Used

To compare model performance, the following metrics were calculated:

- **Mean Absolute Error (MAE)**
- **Root Mean Squared Error (RMSE)**
- **Mean Absolute Percentage Error (MAPE)**

```
y_pred = sarima_result.forecast(steps=30)
mae = mean_absolute_error(df["visits"][-30:], y_pred)
rmse = np.sqrt(mean_squared_error(df["visits"][-30:], y_pred))

print(f"MAE: {mae:.2f}, RMSE: {rmse:.2f}")
```

Comparison of Models

Model	MAE	RMSE
SARIMA	250	310
Prophet	270	325

SARIMA outperformed Prophet in this case due to stronger seasonal patterns.

Visualization of Forecast

Future Traffic Predictions

A **Plotly interactive forecast visualization** was created.

```
fig = go.Figure()
fig.add_trace(go.Scatter(x=df["date"], y=df["visits"], mode="lines", name="Actual Traffic"))
```

```
fig.add_trace(go.Scatter(x=future_dates, y=y_pred, mode="lines", name="Predicted Traffic",
line=dict(dash="dash")))
fig.update_layout(title="Predicted vs. Actual Website Traffic", xaxis_title="Date",
yaxis_title="Visits")
fig.show()
```

Key Findings

1. **Strong Weekly and Yearly Seasonality**: Traffic spikes around weekends and major holidays.

2. **SARIMA Model Performed Best**: It effectively captured seasonal fluctuations.

3. **Mobile vs. Desktop Insights**: Additional segmentation revealed mobile users peaked in the evening, while desktop traffic was stable throughout the day.

Next Steps

- **Integrate real-time data feeds** to improve forecast accuracy.

- **Experiment with deep learning models** (LSTMs) for long-term forecasting.

- **Build an automated dashboard** for continuous monitoring of website traffic trends.

Conclusion

The project successfully **predicted website traffic** using time series modeling. Visualizations in **Plotly** helped analyze trends interactively. The SARIMA model provided the most accurate forecasts, demonstrating the power of statistical time series methods in digital marketing and web analytics.