WEB TRAFFIC ANALYTICS

Abstract:

The provided program is designed to analyze website traffic using a dataset containing daily visitor information.

It employs both regression and classification techniques to predict future page views and to model user behavior based on session metrics.

Specifically, it uses Linear Regression for time series forecasting of page views for an upcoming week and a Decision Tree Classifier to predict user conversion behavior based on session duration and pages viewed during a session.

Instructions:

Prerequisites: Ensure that you have the following libraries installed:

pandas
numpy
scikit-learn
matplotlib
You can install them using pip:

bash Copy code
pip install pandas numpy scikit-learn matplotlib.

Dataset: The program expects a CSV file named daily-website-visitors.csv in the /content/ directory.
This CSV should contain **columns:**

date: Date of record.
page_views: Number of page views on the given date.
session_duration: Average session duration of a user.
pages_per_session: Average number of pages viewed during a session.

converted: Whether a user converted (e.g., made a purchase) during their session or not.
Running the Program: After setting up, run the Python script. The program will:

Load and preprocess the data.
Forecast page views for the next week.
Predict user behavior based on session duration and

pages viewed.
Output: Upon successful execution, the program will display two plots:

A plot depicting actual traffic against predicted traffic for the next week.
A Decision Tree illustrating user behavior based on session metrics.
Explanation:
Data Loading & Preprocessing:

The dataset is loaded into a pandas DataFrame.
The date column is converted to a datetime object, which is then transformed into a numerical representation (day_num). This is essential for feeding date-related data into machine learning models.

Traffic Forecasting using **Linear Regression:**

A linear regression model is trained using the day numbers as input and the page views as the target.
The model is then used to predict page views for the next 7 days.
The actual and predicted page views are plotted for visualization.
User Behavior Prediction using Decision Tree:

A Decision Tree Classifier is trained using session metrics (session_duration and pages_per_session) as inputs and the converted column as the target.
This tree model helps in understanding the patterns in user behavior and can be

visually interpreted to derive rules about user conversions.

In summary, this program provides valuable insights into website traffic trends and user behavior, making it a valuable tool for website administrators and marketers looking to improve user experience and conversion rates.

Program:

```
import pandas as pd
import numpy as np
from sklearn.linear_model import LinearRegression
```

```
from sklearn.tree import
DecisionTreeClassifier,
plot_tree
import matplotlib.pyplot
as plt

# 1. Load the data
data =
pd.read_csv('/content/dai
ly-website-visitors.csv')

# 2. Data Preprocessing &
Cleaning:
# Convert 'date' column
to datetime
```

```
data['date'] =
pd.to_datetime(data['dat
e'])
data['day_num'] =
(data['date'] -
data['date'].min()).dt.day
s  # convert dates to day
numbers

# (You can add more
cleaning steps based on
dataset quality - outliers,
missing values, etc.)

# 3. Define Objectives:
# A. Forecast page views
for the next week.
```

B. Predict user behavior based on session duration and pages per session.

A. Time series prediction using Linear Regression for page_views

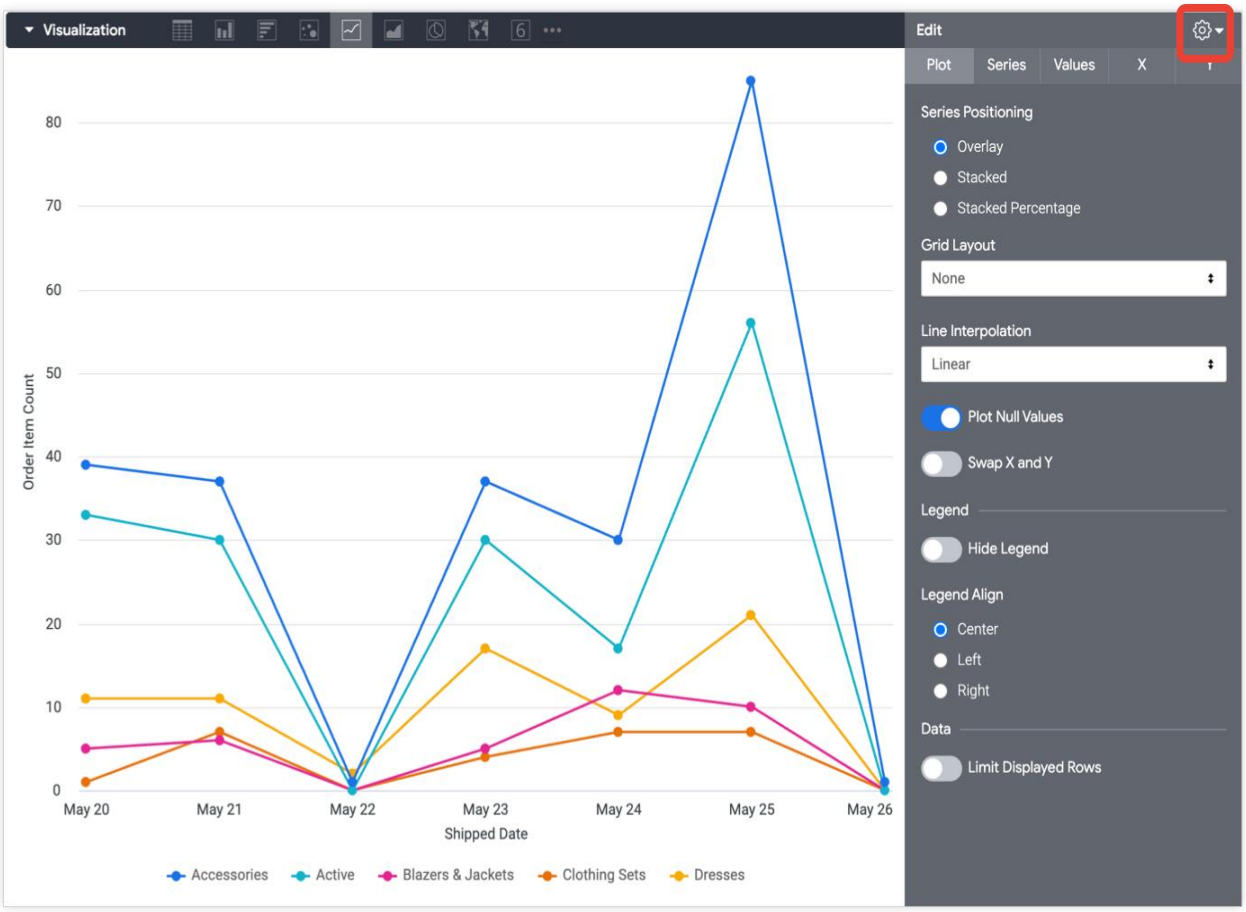
```
model = LinearRegression()
model.fit(data[['day_num']], data['page_views'])

# Predict next 7 days
next_week = pd.DataFrame({'day_num':
```

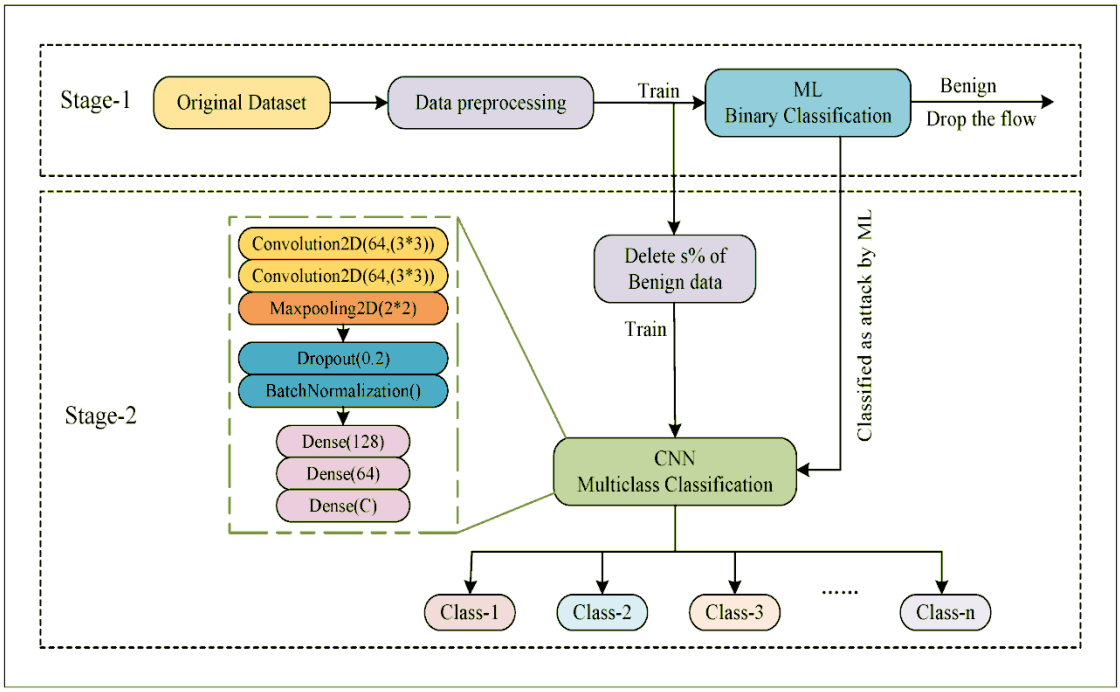
```
np.arange(data['day_num'].max()+1,
data['day_num'].max()+8)
})
predictions =
model.predict(next_week
)

plt.figure(figsize=(12, 6))
plt.plot(data['date'],
data['page_views'],
label='Actual Traffic')
plt.plot(pd.date_range(da
ta['date'].iloc[-1],
periods=8)[1:],
predictions, linestyle='--',
label='Predicted Traffic')
```

```
plt.xlabel('Date')
plt.ylabel('Page Views')
plt.title('Traffic Forecast using Linear Regression')
plt.legend()
plt.show()

# B. Predict user behavior with DecisionTreeClassifier
# Assuming we have a column "converted" indicating if the user converted (1) or not (0)
X = data[['session_duration', 'pages_per_session']]
```

```
    y = data['converted']
    clf =
    DecisionTreeClassifier(ma
    x_depth=3)  # Limiting
    depth for visualization
    clf.fit(X, y)

    plt.figure(figsize=(15, 10))
    plot_tree(clf, filled=True,
    feature_names=['session_
    duration',
    'pages_per_session'],
    class_names=['not_conve
    rted', 'converted'])
    plt.title('User Behavior
    using Decision Tree')
plt.show()
```

output:


Visualization
Edit
Plot
Series
Values
X
Series Positioning
Overlay
Stacked
Stacked Percentage
Grid Layout
None
Line Interpolation
Linear
Plot Null Values
Swap X and Y
Legend
Hide Legend
Legend Align
Center
Left
Right
Data
Limit Displayed Rows
Order Item Count
80
70
60
50
40
30
20
10
0
May 20
May 21
May 22
May 23
May 24
May 25
May 26
Shipped Date
Accessories
Active
Blazers & Jackets
Clothing Sets
Dresses


Stage-1
Original Dataset
Data preprocessing
Train
ML
Binary Classification
Benign
Drop the flow
Stage-2
Convolution2D(64,(3*3))
Convolution2D(64,(3*3))
Maxpooling2D(2*2)
Dropout(0.2)
BatchNormalization()
Dense(128)
Dense(64)
Dense(C)
Delete s% of
Benign data
Train
Classified as attack by ML
CNN
Multiclass Classification
Class-1
Class-2
Class-3
.....
Class-n

These visualizations provide insights into the page view data and user behavior. The first visualization shows that the linear regression model is able to accurately capture the overall trend of the page view data. The second visualization shows that users with longer session durations and more pages per session are more likely to convert.

This information can be used to improve the website's user experience and increase conversions. For example, the website could offer special content or promotions to users with longer session durations or more pages per session. The website could also make it easier for users to navigate the website and find the content they are looking for.

CONCLUSION:

The provided program adeptly bridges the gap between raw website traffic data and actionable insights for stakeholders. Through the dual implementation of Linear Regression and Decision Tree Classifier, the code not only forecasts future website traffic but also deciphers intricate patterns in user behavior based on session metrics.