WEBSITE TRAFFIC ANALYSIS

To execute a Python program like the one I provided in a datasheet or spreadsheet, you typically cannot directly run Python code within the datasheet itself.

1. Create the Python Script:

Use a code editor, such as Visual Studio Code, PyCharm, or a simple text editor, to create a Python script. Copy the code I provided earlier and save it with a `.py` file extension, for example, `website\_traffic\_analysis.py`.

2. Prepare Your Data:

You can input your website traffic data into a spreadsheet or a separate data file, such as a CSV file. Make sure your data file follows the structure expected by the Python script.

3. Run the Python Script:

Open a command prompt or terminal, navigate to the directory where your Python script is saved, and execute it using the Python interpreter. For example:

```bash

python website\_traffic\_analysis.py

```

This will run the Python script, which will load and analyze the data you prepared.

4. Display and Export Results:

We Python script can print or save the results to another file. we can then copy the results back into your datasheet or spreadsheet manually, or we can automate this step by writing code to update the spreadsheet programmatically using libraries like `openpyxl` (for Excel) or `pandas` for data manipulation.

Here's a simplified example of how to use the `pandas` library to load data from a CSV file and save the results to a new CSV file:

```python

import pandas as pd

# Load data

data = pd.read\_csv('website\_traffic\_data.csv')

# Perform analysis (modify this part as needed)

# Save results to a new CSV file

results.to\_csv('analysis\_results.csv', index=False)

```

Keep in mind that integrating Python code directly into a datasheet or spreadsheet application is not a common practice. Instead, Python scripts are typically used to process and analyze data, and the results are imported or exported to and from spreadsheets as needed.

# Import necessary libraries

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score

# Load your website traffic data

data = pd.read\_csv('website\_traffic\_data.csv')

# Preprocess the data (e.g., clean, transform, and feature engineering)

# ...

# Define features and target variable

X = data.drop('target\_variable', axis=1)

y = data['target\_variable']

# 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 a machine learning model (e.g., Random Forest Classifier)

model = RandomForestClassifier()

# Train the model on the training data

model.fit(X\_train, y\_train)

# Make predictions on the testing data

predictions = model.predict(X\_test)

# Evaluate the model's performance

accuracy = accuracy\_score(y\_test, predictions)

print(f"Model Accuracy: {accuracy}")

we will show the output screenshot for this project in next phase