 TwiBot-Spotter

**Running TwiBot-Spotter (Twitter Bot Detection): Step-by-Step Guide**

**Requirements:**

- Python

- HTML

- CSS

- Javascript (frontend)

- Flask (backend)

- Machine learning libraries (such as scikit-learn, Random Forest, KNN Decision Tree depending on the bot detection approach)

**Step 1: Install Python:**

1. Download and install Python from the official website (https://www.python.org/downloads/).

2. Follow the installation instructions for your operating system.

**Step 2: Create a new Python script:**

1. Open a text editor or an integrated development environment (IDE).

2. Create a new Python script file with a .py extension (e.g., bot\_detection.py).

**Step 3: Import the required libraries:**

1. In the Python script, import the necessary libraries such as Pandas and Scikit, including the machine learning libraries you plan to use.

**Step 4: Collect Twitter data:**

1. Use the twitter dataset to collect relevant data from Twitter, such as tweets, user profiles, or follower networks.

2. Specify the search criteria or user handles based on your bot detection approach.

**Step 5: Pre-process the data:**

1. Clean and pre-process the collected Twitter data to remove any irrelevant or redundant information.

2. Perform tasks such as removing stop words, stemming, or tokenization, depending on the analysis requirements.

**Step 6: Train a bot detection model:**

1. Use the preprocessed data to train a machine learning model for bot detection.

2. Choose an appropriate algorithm (e.g., Decision Tree, Random Forest, or KNN algorithm) and train the model using the preprocessed data.

**Step 7: Evaluate the model:**

1. Assess the performance of the trained bot detection model using appropriate evaluation metrics (e.g., accuracy, precision, recall, or F1-score).

2. Make necessary adjustments to the model or feature engineering techniques if required.

**Step 8: Apply the model for bot detection:**

1. Use the trained model to classify Twitter accounts or tweets as bots or non-bots.

2. Apply the model to obtain predictions.

**Step 9: Implementation of Front-end using:**

**-** HTML

- CSS

- JAVASCRIPT

**Step 10: Implementation of Backend**

-FLASK

**Step 11: Interpret the results:**

1. Analyse the bot detection results and interpret the findings.

2. Identify potential bot accounts or bot-related patterns based on the model predictions.