# Twitter Scraper and Stock Prediction Model

This project fetches recent tweets related to stock market discussions using Twitter's API and predicts stock movement (up or down) based on sentiment analysis using a \*\*Random Forest Classifier\*\*. The sentiment of each tweet is used to make a prediction about whether a stock will go "up" or "down."

## Table of Contents

- [Setup Requirements](#setup-requirements)

- [Dependencies](#dependencies)

- [Twitter API Setup](#twitter-api-setup)

- [Running the Script](#running-the-script)

- [Project Workflow](#project-workflow)

- [Data Scraping](#data-scraping)

- [Data Preprocessing](#data-preprocessing)

- [Model Training](#model-training)

- [Model Evaluation](#model-evaluation)

- [Stock Movement Prediction](#stock-movement-prediction)

- [Example Output](#example-output)

- [Notes](#notes)

- [License](#license)

## Setup Requirements

1. \*\*Python Version\*\*: 3.7+ (Ensure you're using a Python 3 environment)

2. \*\*Dependencies\*\*:

You can install the required libraries using the following command:

```bash

pip install tweepy scikit-learn

* **Tweepy**: Python library for accessing Twitter API.
* **Scikit-learn**: Machine learning library used for preprocessing, model training, and evaluation.

1. **Twitter API**:
   * You need a **Twitter Developer account** to access the Twitter API.
   * Create a project and generate **API keys and Bearer Token** from the [Twitter Developer Portal](https://developer.twitter.com/en/portal/dashboard).
   * Replace "your\_bearer\_token" in the code with your actual Bearer Token.

**Running the Script**

Once you have the dependencies installed and the API key configured, follow these steps:

1. **Set up the API**: Replace "your\_bearer\_token" in the script with your actual Bearer Token.
2. **Run the script**:
3. python twitter\_stock\_prediction.py

**Project Workflow**

**Data Scraping**

The script uses the **Tweepy** library to fetch recent tweets related to stock market discussions. It searches for tweets that contain the keywords "stock" or "investing."

**Data Preprocessing**

The tweets are preprocessed using **TF-IDF vectorization** to convert the text data into numeric features. Simple sentiment labels are generated based on specific keywords like "up" or "bullish" (for positive sentiment) and others for negative sentiment.

**Model Training**

A **Random Forest Classifier** is trained using the preprocessed tweet data. The dataset is split into training and testing sets to evaluate model performance.

**Model Evaluation**

The model's performance is evaluated using **accuracy** and a **classification report**, which includes precision, recall, and F1-score.

**Stock Movement Prediction**

The model predicts whether the stock will go "Up" or "Down" based on the sentiment of each tweet.

**Example Output**

1. **Fetched Tweets**: The script will print out the tweets related to stocks.
2. **Model Evaluation**: The accuracy and classification report will be displayed to assess the model's performance.
3. **Stock Movement Prediction**: For each tweet, the model will predict whether the stock will go "Up" or "Down" based on its sentiment.

**Notes**

* The sentiment analysis is based on simple keyword matching (e.g., "up" for positive sentiment). For a more advanced model, you could incorporate deep learning or more sophisticated NLP techniques like **BERT** for sentiment analysis.
* The **Random Forest Classifier** used here is a simple model. You could experiment with more complex models to improve accuracy.

**License**

This project is licensed under the MIT License.

---

### \*\*How to Use the README File:\*\*

1. \*\*Dependencies\*\*: This section helps the user install the required libraries, ensuring they have all the necessary tools to run the project.

2. \*\*Twitter API Setup\*\*: A clear instruction on how to set up the Twitter Developer account and get the Bearer Token.

3. \*\*Running the Script\*\*: Guide the user through replacing the Bearer Token in the script and running the Python script.

4. \*\*Project Workflow\*\*: The explanation of the main steps, such as data scraping, preprocessing, model training, evaluation, and prediction, so the user knows how the system works.

5. \*\*Example Output\*\*: Describes what users will see after running the script to give them a sense of what to expect.

6. \*\*Notes\*\*: Additional suggestions and points to consider for improving or expanding the project.

This README will make it easier for anyone (including yourself) to understand how to set up and use your project.