Project Structure: Twitter/X Sockpuppet Detection System

Overview

This document outlines the file and directory hierarchy, along with filename conventions, for a modular Python project designed to detect sockpuppets on Twitter/X using Playwright, stylometry, and behavioral analysis.

Directory Structure

```
sockpuppet_detector/
|
|-- src/
|   |-- __init__.py
|   |-- main.py                  # Entry point
|   |-- config.py                # Configuration constants
|   |-- utils.py                 # Helper functions
|
|   |-- input/
|   |   |-- __init__.py
|   |   \-- user_input.py        # CLI parser, validation
|
|   |-- browser/
|   |   |-- __init__.py
|   |   |-- controller.py        # Playwright launcher
|   |   \-- page_loader.py       # Navigation & scrolling
|
|   |-- scraping/
|   |   |-- __init__.py
|   |   \-- tweet_extractor.py   # Tweet data extraction
|
|   |-- processing/
|   |   |-- __init__.py
|   |   |-- normalizer.py        # Clean/normalize tweet data
|   |   \-- feature_extractor.py # Stylometry & behavior vectors
|
|   |-- analysis/
|   |   |-- __init__.py
```

```
|   |   |-- similarity_engine.py   # Feature comparison
|   |   \-- detector.py            # Sockpuppet detection logic
|
|   |-- output/
|   |   |-- __init__.py
|   |   \-- reporter.py            # Results & reporting
|
|-- tests/
|   |-- test_input.py
|   |-- test_extractor.py
|   \-- test_detector.py
|
|-- requirements.txt
|-- README.md
\-- .env or .env.example           # (Optional) Credentials
```

Filename Conventions

- All files use `snake_case.py` naming.
- Keep names short and descriptive: `tweet_extractor.py`, not `extract_all_tweets_module.py`.
- Test files follow the format: `test_module.py`.
- Separate files by purpose, not chronology or dev stage.

Development Tips

- Build incrementally from `main.py`.
- Treat each folder as a module by using `__init__.py`.
- Maintain modular separation: scraping, processing, analysis.
- Write unit tests for each module.