Environment Setup

1. Python Environment Setup
   As of August 2025, Python 3.11.x and 3.12.x have been confirmed to work.
   Download from https://www.python.org/downloads/.
   Note: Python 3.13 is currently not supported due to missing LightGBM DLL files.
   During Python installation:
   1. Optional Features – Check "pip" to install the package manager.
   2. Advanced Options – Check "Add Python to environment variables" so python and pip can be used from the command line.
   Important:
   Use the provided requirements.txt for your Python version (3.11 or 3.12).
   SHAP must be installed separately after the requirements using:
   `> pip install shap --only-binary=:all:`
   This ensures all dependencies (including LightGBM and SHAP) work without needing local compilation.
   Verify installation:
   `> python --version`
   Verify pip installation:
   `> pip --version`
   If missing, install pip:
   `> python -m ensurepip --upgrade`
   Upgrade pip (recommended):
   `> python -m pip install --upgrade pip`

   1.1. Using a Virtual Environment *(Recommended)*
   2. Create and Activate a Virtual Environment
   `$ python -m venv venv`
   Windows:
   `> venv\Scripts\activate`
   Linux / macOS:
   `$ source venv/bin/activate`
   3. Install Required Packages from `requirements.txt`
   Windows:
   `$ py -m pip install -r requirements.txt`
   Linux / macOS:
   `$ python3 -m pip install -r requirements.txt`
   1.2. Without a Virtual Environment
   1. Install Python (same as above).
   2. Install Required Packages Globally
   Windows:
   `> pip install -r requirements.txt`
   Linux / macOS:
   `$ pip3 install -r requirements.txt`
   (If you encounter permission issues, add `--user`)

2. Non-Python Environment Requirements
   1. Operating System
   Windows 10 or later, or Ubuntu 20.04+ (Linux)
   macOS is also supported.
   2. Browser
   Modern browser such as Chrome, Firefox, Edge (latest version recommended).
   Required for accessing the Flask web application.
   3. Port
   The Flask application in this project is configured to run on port 8080 by default.
   This is not a fixed requirement — you may change it to any available port if 8080 is in use or blocked by a firewall.
   To change the port, modify the port parameter in the app.run() call inside webapp.py.

   4. ZIP Extraction Tool
   Windows: Built-in Explorer extractor or 7-Zip
   Linux/macOS: `unzip` command or built-in archive manager.

3. Verifying the Setup
   `$ python --version`
   `$ pip list`
   You should see the Python version and all required packages installed (from `requirements.txt`).

4. Package Version Updates
   The following libraries are used in this system:

flask==3.0.3	Web framework used to build and run the web application interface.
pandas==2.2.2	Data manipulation and analysis library for handling CSV and tabular data.
numpy==1.26.4	Numerical computation library for fast mathematical operations on arrays.
scikit-learn==1.4.2	Machine learning library used for model training, evaluation, and preprocessing.
lightgbm==4.3.0	Gradient boosting framework for high-performance model training and prediction.

   `$ pip install --upgrade PACKAGE_NAME`
   or update all installed packages with:
   `$ pip list --outdated`
   `$ pip install --upgrade PACKAGE_NAME`
   Ensure that the application is tested after any version upgrades to maintain compatibility.

5. Latest Updates & Resources
   The latest updates, revision history, and general-purpose tools are available at the following URL.
   https://github.com/NextGenAI-corder/FastSpring