**Reinforcement Learning for Optimizing Pour-Over Coffee Variables**

This project leverages reinforcement learning (RL) techniques to optimize key variables in the pour-over coffee brewing process. The goal is to help baristas and enthusiasts achieve consistent and optimal coffee extraction by modeling and refining parameters such as grind size, brew ratio, brew time, and water temperature.

**Features**

* Implements reinforcement learning using the **stable-baselines3** library in Python.
* Simulates the pour-over coffee brewing process as a dynamic environment.
* Optimizes brewing parameters for maximum rewards using discrete actions.
* Provides visualizations for training performance and convergence.
* Includes a comparison between RL and a baseline KNN regression model.

**Requirements**

Ensure the following dependencies are installed:

* Python (>=3.8)
* Jupyter Notebook
* Libraries:
	+ numpy
	+ pandas
	+ matplotlib
	+ seaborn
	+ stable-baselines3
	+ gym
	+ scikit-learn
	+ tensorflow (if required by stable-baselines3)

Install the dependencies using:

pip install numpy pandas matplotlib seaborn stable-baselines3 gym scikit-learn tensorflow

**Project Structure**

* **Reinforcement Learning for Pour-Over Coffee.ipynb**: The primary Jupyter Notebook containing:
	+ Environment setup for the coffee brewing process.
	+ RL model training and evaluation.
	+ Visualization of results and comparisons.

**Steps for Implementation**

1. **Clone the Repository** Clone or download this project to your local machine.
2. **Set Up the Environment** Ensure Python and all dependencies listed above are installed. Use virtual environments if necessary:

python -m venv coffee\_env

source coffee\_env/bin/activate # On Windows, use `coffee\_env\Scripts\activate`

pip install -r requirements.txt

1. **Open the Notebook** Launch the Jupyter Notebook server and open the Reinforcement Learning for Pour-Over Coffee.ipynb file:

jupyter notebook

1. **Run the Code**
	* Follow the sequential cells in the notebook.
	* Simulate the coffee brewing environment.
	* Train the RL agent using the **stable-baselines3** library.
	* Evaluate model performance and compare with the baseline KNN model.
2. **Visualize Results** Analyze the provided visualizations, such as:
	* Rewards per episode.
	* Learning curve comparison between RL and KNN models.
	* Predictions for optimal brewing parameters.

**Reproducibility**

To ensure consistent results:

1. Use the provided random seed in the notebook to maintain consistent RL training and model evaluations.
2. Ensure your environment matches the dependency versions specified in the **Requirements** section.
3. Save outputs and models periodically for reference and reuse.

For additional questions or support, feel free to contact arif.bramantoro@utb.edu.bn. Enjoy optimizing your pour-over coffee brewing experience!