Instructions

The code for a simple collaborative filtering algorithm with state space integration has been provided. Below are the instructions and documentation:

Instructions for Running the Code

1. **Environment Setup**:
    - Install Python (version 3.7 or higher recommended).

Install required libraries: `numpy` and `scikit-learn`. You can do this by running:

```
pip install numpy scikit-learn
```

    -
2. **Save the Script**:
    - Save the provided code in a file named `state_space_cf.py`.
3. **Execute the Script**:

Run the script using Python:

```
python state_space_cf.py
```

    -
4. **Expected Output**:
    - Initial state vectors for users and items.
    - Updated state vectors after training on interactions.
    - Top recommended items for a sample user.

Code Documentation

Class: `StateSpaceCollaborativeFilter`

- **Attributes**:
    1. `n_users`: Number of users.

    2. `n_items`: Number of items.
    3. `user_states`: Matrix of user state vectors, initialized randomly.
    4. `item_states`: Matrix of item state vectors, initialized randomly.
- **Methods**:

    1. **`__init__(self, n_users, n_items)`**:
        - Initializes user and item states.
    2. **`update_states(self, interactions, learning_rate=0.01)`**:
        - Updates user and item states based on interactions.
        - Parameters:
            - `interactions`: List of tuples `(user_id, item_id, rating)`.
            - `learning_rate`: Step size for updates.
    3. **`predict(self, user, item)`**:
        - Predicts the rating for a user-item pair.
    4. **`recommend(self, user, top_k=5)`**:
        - Recommends the top-k items for a given user.
        - Parameters:
            - `user`: User ID.
            - `top_k`: Number of items to recommend.

Main Function:

- Sets up a small example with users, items, and interactions.
- Demonstrates training and recommendations.