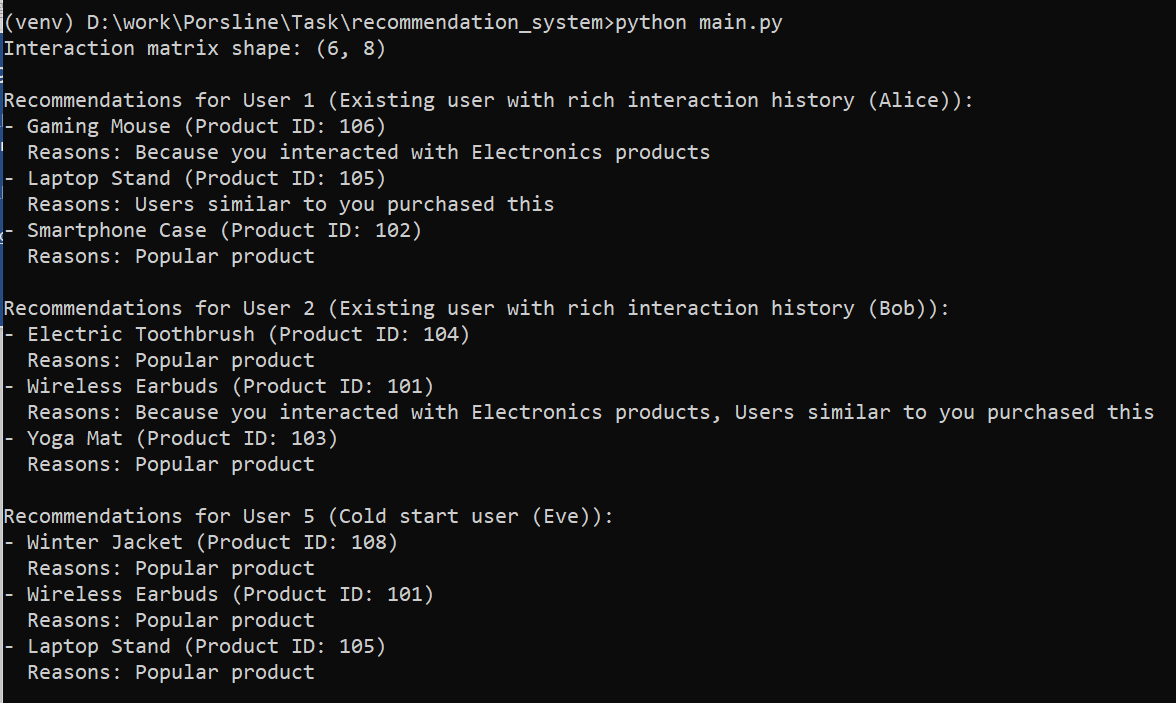
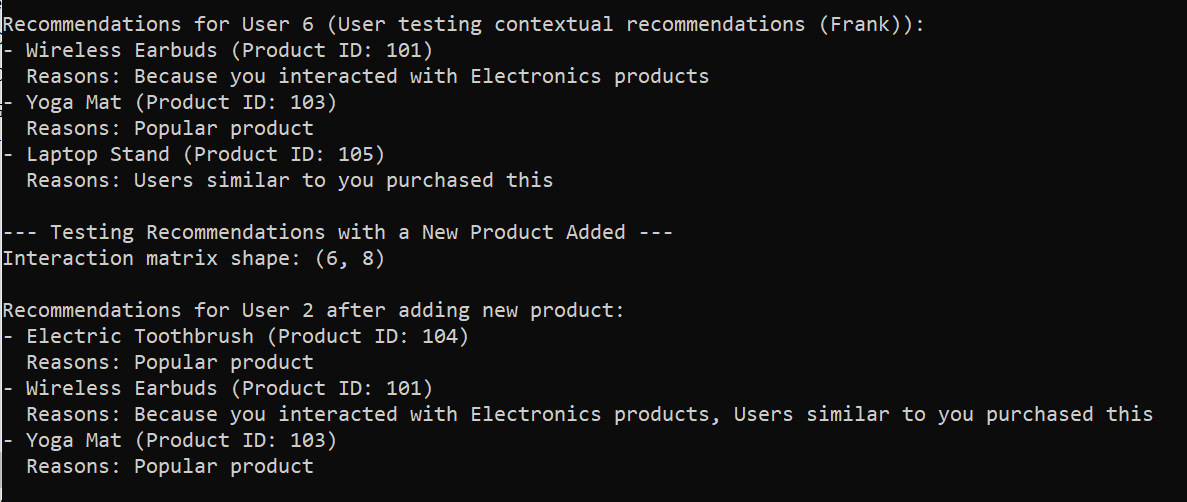
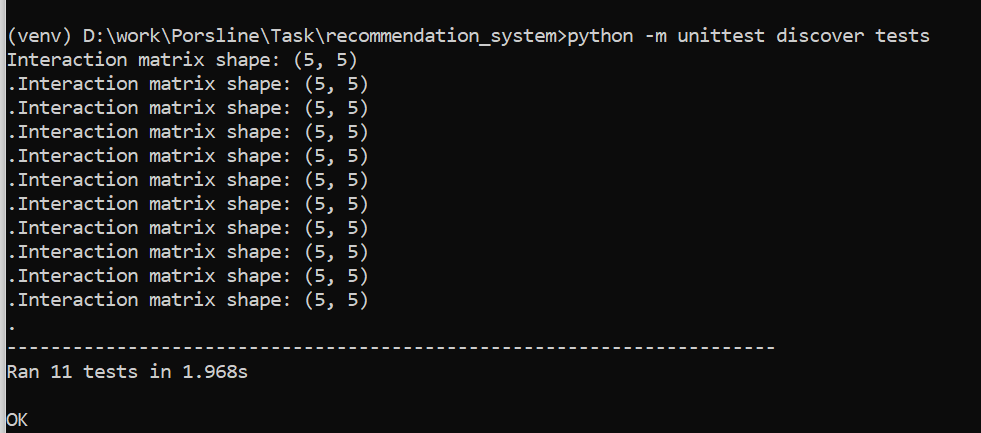
**Design Document**

**Ghazal Bakhshande**

**Output:**

**** ****

**Tests Output:**

****

**Data Structures**

* **df\_users:** Contains user profiles with fields like user\_id, name, and device.
* **df\_products:** Holds product details such as product\_id, name, category, tags, and rating.
* **df\_interactions:** Records user interactions with products, including user\_id, product\_id, and weight (e.g., ratings).
* **df\_purchase\_history:** Logs users' purchase transactions, capturing user\_id, product\_id, purchase\_date, and quantity. This helps in understanding users' buying patterns and preferences.
* **df\_browsing\_history:** Tracks users' browsing activities, including user\_id, product\_id, and timestamp. This provides insights into users' interests and potential future purchases.
* **df\_context:** Stores contextual information like peak days and seasons for product categories.
* **df\_products\_encoded:** An encoded version of df\_products, where categorical features are transformed into numerical formats suitable for clustering and modeling.
* **User-Item Interaction Matrix (user\_item\_matrix):** A pivot table of user interactions used for collaborative filtering.
* **TF-IDF Matrix:** Used in content-based filtering to represent product metadata numerically.

**Algorithm Choices and Reasoning**

1. **Content-Based Filtering**

* **Algorithm:** Uses TF-IDF vectorization of product metadata (tags, categories) and computes cosine similarity between products.
* **Reasoning:** Recommends products similar to those a user has previously interacted with, focusing on product attributes.

1. **Collaborative Filtering**

* **Algorithm:** Calculates user-user similarity based on interaction patterns using cosine similarity.
* **Reasoning:** Identifies users with similar preferences to recommend products that similar users have liked.

1. **Matrix Factorization**

* **Algorithm:** Employs Singular Value Decomposition (SVD) to factorize the user-item interaction matrix.
* **Reasoning:** Uncovers latent factors representing underlying preferences, improving recommendation accuracy.

1. **Clustering**

* **Algorithm:** Applies K-Means clustering to group users and products based on interaction data and features.
* **Reasoning:** Enhances collaborative filtering by considering cluster-based similarities.

1. **Context-Aware Adjustments**

* **Algorithm:** Modifies recommendations based on contextual factors (e.g., device type, time, season).
* **Reasoning:** Improves relevance by incorporating situational variables affecting user behavior.

1. **Hybrid Recommendation**

* **Algorithm:** Combines outputs from all models using a weighted scoring system to generate final recommendations.
* **Reasoning:** Leverages the strengths of each method to create a more robust and personalized recommendation list.

**Optimization Techniques and Trade-Offs**

1. **Data Encoding and Preprocessing**

* **Technique:** One-hot encoding of categorical features (e.g., product tags) for numerical processing.
* **Trade-Off:** Increases dimensionality, which can impact computational efficiency but is necessary for algorithms requiring numerical input.

1. **Dimensionality Reduction**

* **Technique:** Uses PCA to reduce the dimensionality of high-dimensional data (e.g., interaction matrix) before clustering.
* **Trade-Off:** Balances computational efficiency with potential loss of information. A reduced number of components may speed up processing but could omit important variance.

1. **Weighted Hybrid Model**

* **Technique:** Assigns weights to different recommendation sources to control their influence.
* **Trade-Off:** Requires careful tuning; inappropriate weights can skew recommendations away from personalization.

1. **Handling Cold Start Problem**

* **Technique:** For new users with no interaction history, the system recommends popular products or products popular within clusters.
* **Trade-Off:** Recommendations may be less personalized but ensure that new users receive relevant suggestions.

1. **Diversity Enhancement**

* **Technique:** Incorporates diversity in recommendations by ensuring a mix of product categories.
* **Trade-Off:** May slightly reduce relevance if less-preferred categories are included but increases the chance of user engagement with diverse options.

1. **Model Performance Optimization**

* **Technique:** Uses efficient algorithms and limits model complexity (e.g., limiting the number of latent factors in SVD).
* **Trade-Off:** Improves speed and scalability at the potential cost of reduced model expressiveness.