**Mechanical Shop Recommendation System**

This repository contains code for a recommendation system for a mechanical shop. The system utilizes collaborative filtering and content-based filtering techniques, as well as a neural network model for hybrid filtering. It also includes data preprocessing steps and data visualization.

**Columns Explanation**

**Preprocessing and Data Loading**

* **customer\_data**: A DataFrame that contains historical customer data. It is loaded from a CSV file called "synthetic\_data.csv".

**Collaborative Filtering**

* **user\_item\_matrix**: A pivot table that represents the user-item matrix. It is computed from the **customer\_data** DataFrame and contains user ratings for each item. If a user hasn't rated an item, the value is filled with 0.
* **user\_similarity**: A matrix that represents the similarity between users using cosine similarity. It is computed based on the **user\_item\_matrix**.
* **N**: The number of similar users to consider for each user.
* **top\_similar\_users**: A dictionary that stores the top N similar users for each user. It is computed based on the **user\_similarity** matrix.

**Content-based Filtering**

* **item\_matrix**: A matrix representing the item-item matrix using TF-IDF vectors. It is computed from the item descriptions in the **customer\_data** DataFrame.
* **item\_similarity**: A matrix representing the similarity between items using cosine similarity. It is computed based on the **item\_matrix**.
* **top\_similar\_items**: A dictionary that stores the top N similar items for each item. It is computed based on the **item\_similarity** matrix.

**Neural Network Model for Hybrid Filtering**

* **user\_input**: Input layer for the user ID.
* **item\_input**: Input layer for the item ID.
* **embedding\_dim**: The dimension of the embedding for users and items.
* **num\_items**: The number of unique items in the **customer\_data** DataFrame.
* **user\_embedding**: Embedding layer for the user ID.
* **item\_embedding**: Embedding layer for the item ID.
* **dot\_product**: Dot product between user and item embeddings.
* **output**: Output layer of the model.
* **model**: The compiled neural network model that takes user and item inputs and predicts the purchase probability.

**Data Visualization**

* **user\_item\_matrix**: Heatmap plot of the user-item matrix.
* **item\_similarity**: Heatmap plot of the item-item similarity matrix.

**Instructions**

To use the recommendation system and visualize the data, follow these steps:

1. Load the historical customer data from the "synthetic\_data.csv" file.
2. Compute the user-item matrix using collaborative filtering.
3. Compute the similarity between users using cosine similarity.
4. Get the top N similar users for each user.
5. Compute the item-item matrix using TF-IDF vectors for content-based filtering.
6. Compute the similarity between items using cosine similarity.
7. Get the top N similar items for each item.
8. Define the neural network model for hybrid filtering.
9. Preprocess the data by encoding the user\_id column with unique integer values.
10. Train the model using the user\_id, item\_id, and purchase columns.
11. Visualize the user-item matrix heatmap.
12. Visualize the item-item matrix heatmap.

Note: Make sure to have the required dependencies installed, such as pandas, sklearn, tensorflow, matplotlib, seaborn, and numpy.

Feel free to modify and adapt the code to fit your specific use case.