

Question1.1

Here is the code

```
import pandas as pd
import numpy as np
from sklearn.metrics.pairwise import cosine_similarity
import os

data_path = 'E:/ml-100k/u.data' # Your file path
try:
    ratings = pd.read_csv(data_path, sep='\t', names=['user_id', 'item_id', 'rating', 'timestamp'], encoding='utf-8')
except FileNotFoundError:
    print(f"Error: File {data_path} not found. Ensure the MovieLens 100k dataset is downloaded and extracted to the correct path.")
    raise

utility_matrix = ratings.pivot(index='user_id', columns='item_id', values='rating')
user_means = utility_matrix.mean(axis=1)
centralized_matrix = utility_matrix.sub(user_means, axis=0).fillna(0)

cosine_sim = cosine_similarity(centralized_matrix)
cosine_sim_df = pd.DataFrame(cosine_sim, index=utility_matrix.index, columns=utility_matrix.index)
user_1_sim = cosine_sim_df.loc[1].sort_values(ascending=False)[1:11] # Exclude user 1
top_10_users = user_1_sim.index
print("Top 10 users most similar to user 1:", top_10_users.tolist())

similar_users_ratings = utility_matrix.loc[top_10_users, 508].dropna()
if similar_users_ratings.empty:
    print("No similar users have rated item 508. Prediction cannot be made.")
else:
    weights = user_1_sim.loc[similar_users_ratings.index]
    weighted_sum = np.sum(similar_users_ratings * weights)
    weight_sum = np.sum(weights)
    predicted_rating = weighted_sum / weight_sum if weight_sum != 0 else 0
    user_1_mean = user_means.loc[1]
    final_predicted_rating = predicted_rating + user_1_mean
    print(f"Predicted rating for user 1 on item 508: {final_predicted_rating:.2f}")
```

Here is the output

```
Top 10 users most similar to user 1: [773, 868, 592, 880, 429, 276, 916, 222, 457, 8]
Predicted rating for user 1 on item 508: 7.82
```

We identified the 10 users most similar to user 1 from the dataset, specifically users 773, 868, 592, 880, 429, 276, 916, 222, 457, and 8. Based on these users' ratings for item 508, we calculated the expected rating for user 1 on item 508 to be 7.82.

Question1.2

Here is the code

```
import pandas as pd
import numpy as np
from sklearn.metrics.pairwise import cosine_similarity
import os

data_path = 'E:/ml-100k/u.data' # Your file path
try:
    ratings = pd.read_csv(data_path, sep='\t', names=['user_id', 'item_id', 'rating', 'timestamp'], encoding='utf-8')
except FileNotFoundError:
    print(f"Error: File {data_path} not found. Ensure the MovieLens 100k dataset is downloaded and extracted to the correct path.")
    raise

utility_matrix = ratings.pivot(index='user_id', columns='item_id', values='rating')
user_means = utility_matrix.mean(axis=1)
centralized_matrix = utility_matrix.sub(user_means, axis=0).fillna(0)

centralized_matrix_T = centralized_matrix.T
item_95_profile = centralized_matrix[95].values.reshape(1, -1)
item_similarities = cosine_similarity(centralized_matrix_T)
item_95_sim = item_similarities[94]
similar_items = np.argsort(item_95_sim)[::-1][1:11]

user_200_ratings = utility_matrix.loc[200].fillna(0)
user_15_ratings = utility_matrix.loc[15].fillna(0)

def predict_rating(user_ratings, similar_items, item_similarities):
    weights = item_similarities[similar_items]
    ratings = user_ratings.iloc[similar_items + 1]
    valid = ratings != 0
    if valid.sum() == 0:
        return 0
    weighted_sum = np.sum(ratings[valid] * weights[valid])
    weight_sum = np.sum(weights[valid])
    return weighted_sum / weight_sum if weight_sum != 0 else 0

pred_200_95 = predict_rating(user_200_ratings, similar_items, item_95_sim)
pred_15_95 = predict_rating(user_15_ratings, similar_items, item_95_sim)
user_200_profile_user_space = centralized_matrix.loc[200].values.reshape(1, -1)
user_15_profile_user_space = centralized_matrix.loc[15].values.reshape(1, -1)
print(f"Predicted rating for user 200 on item 95: {pred_200_95:.4f}")
print(f"Predicted rating for user 15 on item 95: {pred_15_95:.4f}")

recommended_user = 200 if pred_200_95 > pred_15_95 else 15
print(f"The recommender system will recommend movie 95 to user {recommended_user} due to higher predicted rating.")
```

Here is the out put

Predicted rating for user 200 on item 95: 4.1569

Predicted rating for user 15 on item 95: 0.0000

The recommender system will recommend movie 95 to user 200 due to higher predicted rating.

We used the following formula to calculate cosine similarity.

$$\text{cosine_similarity}(u, v) = \frac{u \cdot v}{\|u\| \|v\|}$$

After computing the similarity scores, the predicted rating for item 95 by user 200 was 4.1569, while the predicted rating by user 15 was 0. Since user 200 had a higher predicted rating for item 95, the recommendation system would suggest movie 95 to user 200.