import numpy as np

import pandas as pd

from sklearn.metrics.pairwise import cosine\_similarity

# Load MovieLens dataset

ratings\_data = pd.read\_csv('ratings.csv')

movies\_data = pd.read\_csv('movies.csv')

# Create user-item matrix

user\_item\_matrix = ratings\_data.pivot\_table(index='userId', columns='movieId', values='rating')

# Compute user similarity matrix (cosine similarity)

user\_similarity = cosine\_similarity(user\_item\_matrix.fillna(0))

# Define function to generate recommendations for a given user

def generate\_recommendations(user\_id, user\_similarity\_matrix, user\_item\_matrix, n=5):

# Get similar users

similar\_users = user\_similarity\_matrix[user\_id]

# Find top N similar users (excluding the user itself)

top\_similar\_users = np.argsort(similar\_users)[::-1][1:n+1]

# Get items rated by similar users but not by the target user

rated\_items = user\_item\_matrix.loc[user\_id]

unrated\_items = user\_item\_matrix.loc[top\_similar\_users].fillna(0)

unrated\_items = unrated\_items.replace(rated\_items, np.nan)

# Calculate mean rating of unrated items by similar users

mean\_ratings = unrated\_items.mean()

# Get top N recommended items

top\_recommendations = mean\_ratings.sort\_values(ascending=False).head(n)

return top\_recommendations

# Example usage

user\_id = 1

recommendations = generate\_recommendations(user\_id, user\_similarity, user\_item\_matrix)

print("Top 5 recommendations for user", user\_id, ":\n", recommendations)