به نام خدا
دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر

درس مبانی رایانش ابری

موضوع: تمرین امتیازی

هستی السادات جلالی چیمه – ۹۸۳۱۰۱۸

هلیا سادات هاشمی‌پور - ۹۸۳۱۱۰۶

نیم‌سال اول ۱۴۰۲-۱۴۰۱

در این بخش با استفاده از شباهت کسینوسی اقدام به ایجاد یک سیستم توصیه‌گر کرده‌ایم.

ابتدا دیتاست‌ها را لود می‌کنیم.
در مرحله بعد به ازای آی‌دی کاربر تشابه کسینوسی وی با سایر کاربران محاسبه می‌شود:

```
def cosine_similarity(v1, v2, norm1, norm2 ):
    return np.dot(v1, v2) / (norm1 * norm2)

# calc cosine similarity of specific user
def calc_similarity(user_id):
    similarity = np.zeros((n, 1))
    if norm[user_id - 1] != 0:
        for i in range(n):
            if norm[i] == 0:
                similarity[i][0] = 0
            else:
                similarity[i][0] = cosine_similarity(rating[user_id - 1],
rating[i], norm[user_id - 1], norm[i])
    return similarity
```

و آی‌دی کاربران مشابه و میزان تشابه آنها return می‌شود:

```
# find the most similar users to a specific user
def find_similar_users(user_id, k):
    similarity = calc_similarity(user_id)
    similar_users = np.zeros((k, 2))
    for i in range(k):

        # save the most similar user id and similarity
        similar_users[i][0] = int(np.argmax(similarity) + 1)
        # print(similar_users[i][0])
        similar_users[i][1] = similarity[np.argmax(similarity)]
        # print(similar_users[i][1])

        # remove the most similar user
        similarity[np.argmax(similarity)] = -1
    return similar_users
```

در این مرحله ابتدا آی‌دی بازی‌هایی که کاربر به آنها امتیاز نداده را محاسبه می‌کنیم.
سپس برای تمامی بازی‌هایی که کاربر اصلی به آنها امتیاز نداده است یک امتیاز در نظر میگیریم که میانگین امتیازهایی است که کاربران مشابه به آن داده‌اند ضرب در میزان شباهت کاربران، خروجی را به ازای امتیازهای تخصیص داده شده به بازی‌ها مرتب می‌کنیم و ۵ بازی را برای کاربر ارسال می‌کنیم:

```
# recommend games to a specific user
# k: number of similar users
# m: number of games to recommend
def recommend_games(user_id, k, m=5):

    # find the most similar users with shape (k, 2)
```

```
    similar_users = find_similar_users(user_id, k)

    # find the games that the user has not rated
    games_not_rated = np.where(rating[user_id - 1] == 0)[0]

    game_numbers = 10000

    # set the number of rated games for each game
    rated_number = np.zeros((game_numbers, 1))

    # set score for each game
    game_score = np.zeros((game_numbers, 1))
    for i in range(k):
        for j in games_not_rated:
            result = rating[int(similar_users[i][0]) - 1][j] * similar_users[i][1]
            game_score[j] += result
            if result != 0:
                rated_number[j] += 1

    # replace the games that have not been rated by any similar users with -1
    rated_number = np.where(rated_number == 0, -1, rated_number)

    # calculate the average score of each game
    game_score = game_score / rated_number

    # find the most similar games by their scores and id
    recommended_games = np.zeros((m, 2))
    for i in range(m):
        recommended_games[i][0] = int(np.argmax(game_score) + 1)
        recommended_games[i][1] = game_score[np.argmax(game_score)]
        game_score[np.argmax(game_score)] = -1
    return recommended_games
```