

## Звіт з лабораторної роботи №3

Кишинський Олександр

КНС-21

### Завдання

Побудувати рекомендаційну модель на основі колаборативної фільтрації

### Хід виконання:

Колаборативна фільтрація ґрунтується на спільних інтересах користувачів, аналізуючи подібність їх вподобань для генерації певних прогнозів. У цій лабораторній роботі реалізована рекомендаційна система, що є заснована на 50-ти об'єднаннях (певних групах) користувачів.

Оскільки готових даних з якими ми будемо працювати немає, їх можна згенерувати безпосередньо в самому коді. Уявимо що ми створюємо рекомендації для перегляду фільмів:

```
num_users_per_group = 2
num_movies = 50

user_ratings = np.random.randint(1, 11, size=(num_users_per_group *
num_movies, num_movies))
```

Було згенеровано 50 фільмів, на кожен з яких було виділено одну певну групу користувачів, генеруючи таким чином 50 об'єднань. Також були випадково згенеровані рейтинги для цих фільмів - від 1 до 10.

Тому як дані є випадково згенерованими – нам буде важко отримати дійсні рекомендації, але в якості прикладу це припустимо,

адже навіть в рандомно згенерованих даних можна відшукати ту чи іншу закономірність.

Щоб відокремити певні складові, на яких можна буде робити рекомендації і заодно зменшити розмірності даних з яким ми працюємо можна скористатися SVD (Singular Value Decompression) методом, який потрібно заздалегідь імпортувати з бібліотеки sklearn.

```
svd = TruncatedSVD(n_components=10, random_state=42)
svd.fit(user_ratings)
```

Ми навчаємо нашу модель цим методом, зберігаючи лише перші 10 складових (цими складовими є три види матриць з якими SVD метод працює за лаштунками). Ці матриці найкраще пояснюють змінність в наших даних.

Тепер для кожної групи користувачів можна зробити власну рекомендацію і отримати прогнозовані оцінки:

```
selected_group_index = 0
start_index = selected_group_index * num_users_per_group
end_index = (selected_group_index + 1) * num_users_per_group

selected_group_ratings = user_ratings[start_index:end_index]

user_ratings_predicted
= svd.inverse_transform(svd.transform(selected_group_ratings))
user_ratings_predicted = np.clip(user_ratings_predicted, 1, 10)

recommendations = []

for user_id, predicted_ratings in enumerate(user_ratings_predicted):
    recommendations.extend([(i+1, predicted_ratings[i]) for i in
range(num_movies)])

print("Рекомендовані фільми для перегляду:")
print("-----")
for movie_id, predicted_rating in sorted(recommendations, key=lambda x: x[1],
reverse=True):
    print(f"Movie ID: {movie_id}, Передбачена оцінка: {predicted_rating}")
    if predicted_rating >= 9:
        print("Максимальна рекомендація до перегляду!")
    elif predicted_rating >= 7:
        print("Рекомендовано до перегляду.")
    elif predicted_rating >= 5:
        print("Може вас зацікавити.")
    else:
        print("Можливо це не найкраща ідея.")
print("-----")
```

1. Вибираємо групу користувачів за індексом `selected_group_index`.
2. Виділяємо рейтинги цієї групи користувачів з загальної матриці рейтингів за допомогою змінних `start_index` і `end_index`.
3. Застосовуємо модель SVD для прогнозування оцінок для цієї групи користувачів.
4. Обмежуємо прогнозовані оцінки, щоб вони знаходилися у межах від 1 до 10.
5. Створюємо список рекомендацій для кожного фільму на основі прогнозованих оцінок.
6. Виводимо рекомендації з можливими описами відповідно до рівня оцінок.