Ссылка на датасет для jupyter-nitebook: https://www.kaggle.com/netflix-inc/netflix-prize-data

В следующем задании вам необходимо будет написать свою имплементацию **SVD разложения**.

1. Создадим матрицу 'пользователь-фильм', на которой будем практиковаться с SVD-разложением

   user_movie_matrix = np.array(((1,5,0,5,4), (5,4,4,3,2), (0,4,0,0,5), (4,4,1,4,0), (0,4,3,5,0), (2,4,3,5,3)))

2. Сингулярное разложение матрицы — это некое новое представление исходной матрицы, которое отображено в сингулярных числах и сингулярных векторах. Так выглядит формула сингулярного разложения:
   $M = U\sigma V$,
   где M – исходная матрица 'пользователь-фильм'; σ – матрица, на диагонали которой лежат сингулярные числа, а вне диагонали нули; U – матрица-представление пользователей; V – матрица-представление фильмов.

   Важно понимать, что в лице матриц U и V получается новое представление пользователей и фильмов соответственно. Для расчета сингулярных чисел и сингулярных векторов, используется встроенная функция библиотеки scipy:

   U,s,V=scipy.linalg.svd(M)

   Задание: При помощи библиотеки scipy проведите SVD-разложение матрицы **user_movie_matrix.**

3. На этом этапе получилось новое матричное представление пользователей и фильмов. Размерность разложения в данном случае совпадает с размерностью исходной матрицы. На практике обычно используют сокращенное представление, так как алгоритм SVD обладает свойством уменьшения размерности без значительной потери информации. На больших матрицах 1 000 000 x 1 000 000 это особенно ощутимо. Для того, чтобы сократить размерность, например, до 2х компонент, необходимо взять по 2 первых строки из каждой матрицы-представления.

   Задание: Извлеките из матричных представлений первые 2 строки и сохраните их. Все дальнейшие действия будут производиться с ними.

4. SVD-разложение на 2 компоненты произведено. Теперь необходимо посмотреть какой фильм наше новое представление порекомендует в первую очередь для существующего пользователя.

   Задание: Из исходной матрицы выберем 2-ю строчку (пользователь №2) - ***user_2***

5. Чтобы перевести пользователя в новое представление сниженной размерности, необходимо его исходный вектор умножить на транспонированную матрицу-представление фильмов

   $$lowdim = user\_2V^T$$

   Задание: Произвести вычисление представления пользователя сниженной размерности.

6. Финальный шаг — необходимо произвести обратную трансформацию вектора в вектор оценок фильмов (т.е. в исходное представление фильмов). Для этого необходимо представление пользователя сниженной размерности умножить на матрицу-представление фильмов

   $$inversed\_transformation = lowdimV$$

   Задание: Произвести обратную трансформацию вектора сниженной размерности в исходное пространство оценок.

7. Теперь получен новый вектор оценок для пользователя №2 (обратите внимание на незначительное изменение оценок фильмов, которые этот пользователь действительно посмотрел).

8. **Задание:** Укажите индекс непросмотренного пользователем №2 фильма, который имеет наибольшую оценку.

9. **Задание:** Проверьте SVD-разложение на новом пользователе. Выведите индекс непросмотренного новым пользователем фильма, который имеет наибольшую оценку. При этом новый пользователь что-то успел посмотреть: new_user=np.array((0,0,3,4,0)).

   Обратите внимание: SVD-разложение проводите без нового пользователя! В этот раз используйте 3 компоненты.