**Задание 1**

**Текстовый отчет “Комедии с рейтингом от всех пользователей”**

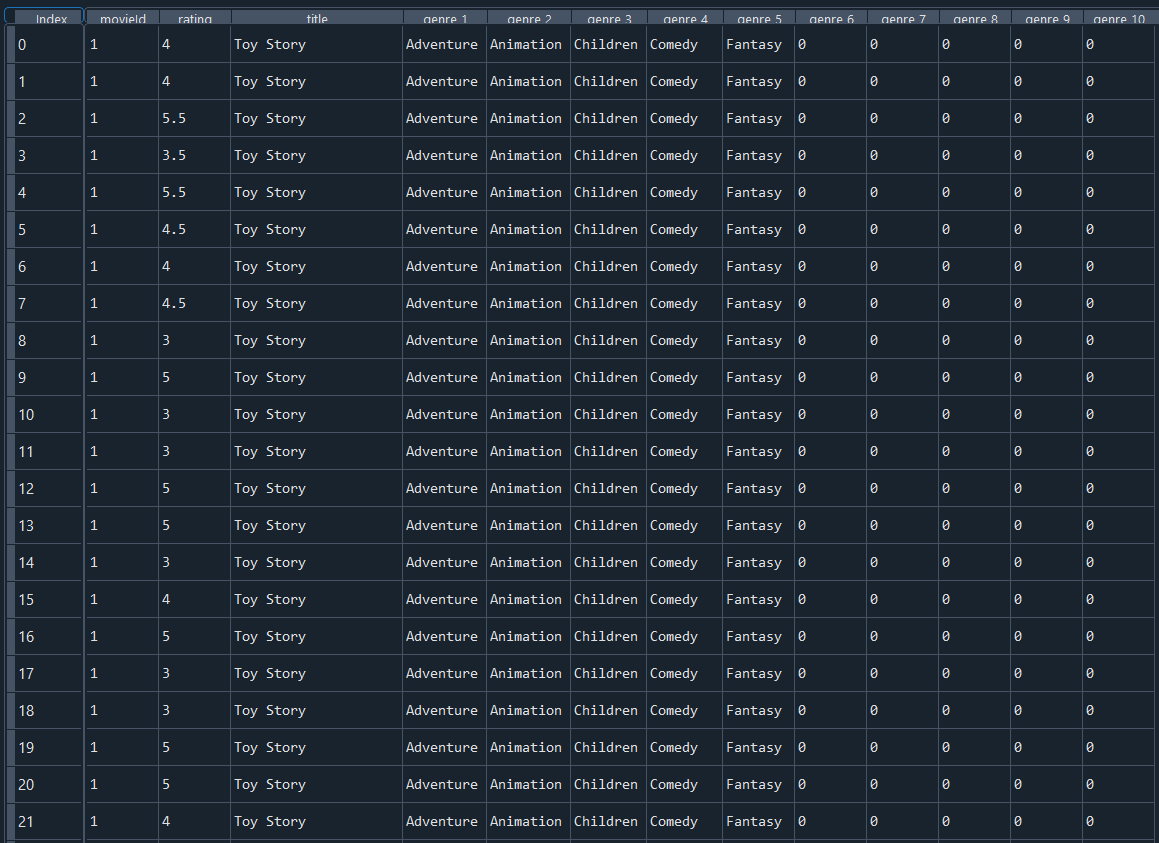
Для кого: для пользователя, который хочет посмотреть оценки комедий от всех пользователей

Названия столбцов: movieId, rating, title, genre 1, genre 2, genre 3, genre 4, genre 5, genre 6, genre 7, genre 8, genre 9, genre 10

Критерии:

Выбираются все фильмы с жаном “Comedy”(перебираются все столбцы genre)

Эти фильмы объединяются с рейтингом каждого пользователя



**def comedies\_with\_ratings\_from\_all\_users(comedy, rating):**

"""

Данная функция показывает рейтинг от каждого пользователя к фильмам жанра комедия.

**• comedy:** таблица с фильмами только определенного жанра

**• rating:** таблица с рейтингами от пользователей

"""

*# comedy\_all\_rating = rating.merge(comedy)*

*# del comedy\_all\_rating['year']*

*# del comedy\_all\_rating['userId']*

**Текстовый отчет “Современные фильмы c большим количеством оценок”**

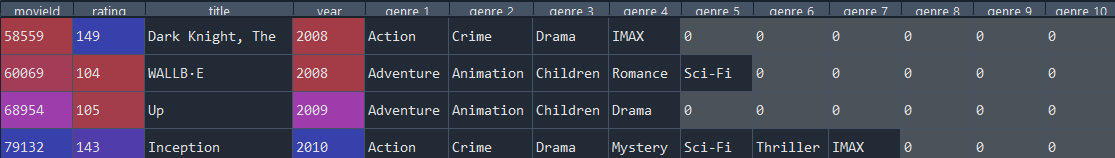
Для кого: Для пользователя, который хочет посмотреть хороший современный фильм

Названия столбцов: movieId, title, year, rating

Критерии:

Выбираются все фильмы с годом выпуска позже 2006 года.

Среди этих фильмов выбираются только те у которых количество оценок от пользователей больше 100



**def modern\_films\_with\_a\_large\_number\_of\_ratings(movies,rating\_count)**

"""

Данная функция показывает фильмы выпущенные после 2006 года с количеством оценок больше 100.

**• movies:** таблица со всеми фильмами

**• rating\_count:** таблица с количеством отзывов

"""

*# new\_movies=movies[movies["year"]>2006]*

*# rating\_count\_100 = rating\_count[rating\_count["rating"]>100]*

*# new\_movies\_100=rating\_count\_100.merge(new\_movies, how = 'inner', left\_on='movieId', right\_on='movieId')*

**Текстовый отчет “Новинки с рейтингом 5,5”**

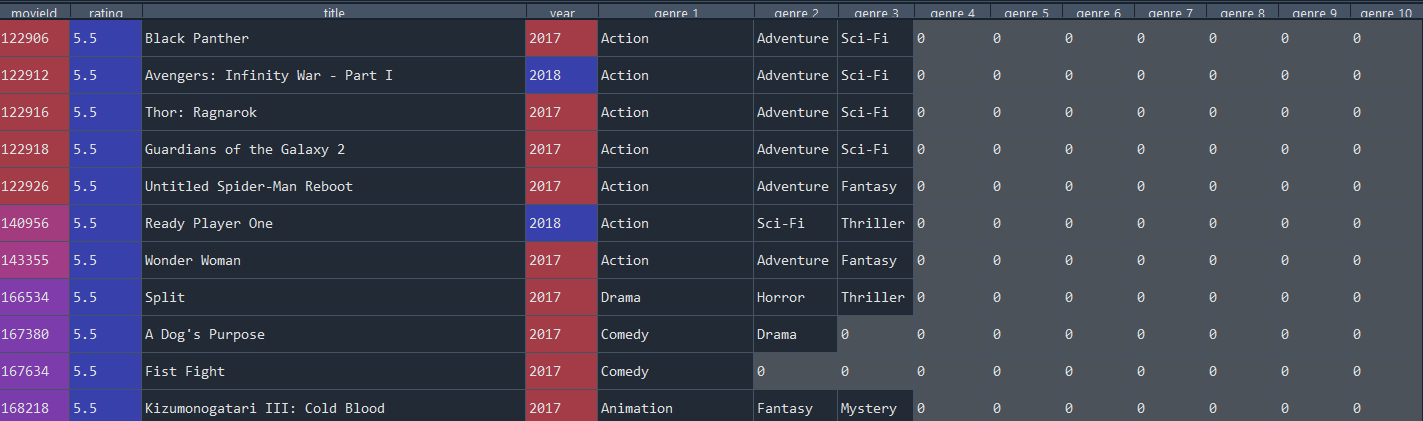
Для кого: для списка бестселлеров

Названия столбцов: movieId,rating, title, year, genre 1, genre 2, genre 3, genre 4, genre 5, genre 6, genre 7, genre 8, genre 9, genre 10

Критерии:

Выбираются все фильмы с годом выпуска не раньше 2017 года.

Среди этих фильмов выбираются только те, у которых есть оценка 5,5



**def new\_items\_rated\_5\_5(movies, rating\_max):**

"""

Данная функция показывает новые фильмы, которые хотя бы один пользователь оценил на 5.5.

**• movies:** таблица со всеми фильмами

**• rating\_max:** таблица с максимальными рейтингами

"""

*# supernew\_movies=movies[movies["year"]>=2017]*

*# rating\_max\_5\_5 = rating\_max[rating\_max["rating"]==5.5]*

*# supernew\_movies\_5\_5=rating\_max\_5\_5.merge(supernew\_movies, left\_on='movieId', right\_on='movieId')*

**Задание 2**

**Средний рейтинг каждого фильма с помощью функции *Pandas.pivot\_table***

*rating\_mean = pd.pivot\_table(rating, values=['rating'], index=['movieId'], aggfunc= {'rating': np.mean})*

**Текстовый отчет “Комедии с рейтингом 5”**

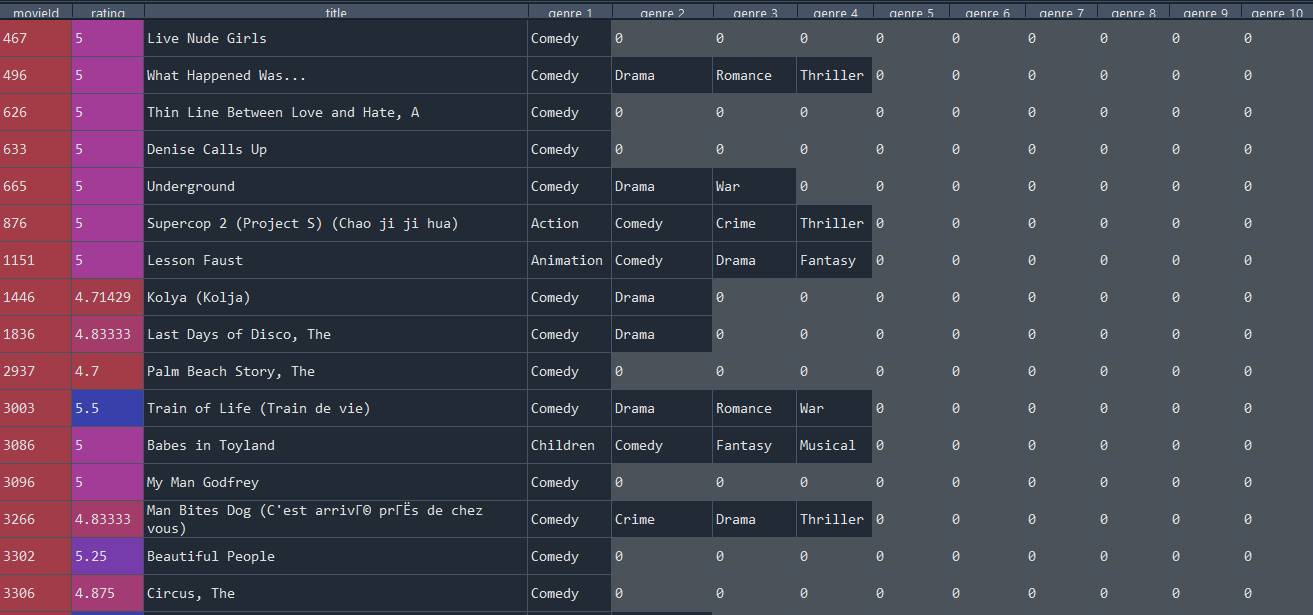
Для кого: для пользователя, который хочет посмотреть хорошую комедию

Названия столбцов: movieId, rating, title, genre 1, genre 2, genre 3, genre 4, genre 5, genre 6, genre 7, genre 8, genre 9, genre 10

Критерии:

Выбираются все фильмы с жаном “Comedy”(перебираются все столбцы genre)

Среди этих фильмов выбираются только те у которых средний рейтинг больше чем 4,7)



**def сomedies\_with\_a\_rating\_of\_5(comedy, rating\_mean):**

"""

Данная функция показывает комедии с средним рейтингом 4.7 и более.

**• comedy:** таблица с фильмами только определенного жанра

**• rating\_mean:** таблица с средним рейтингом всех фильмов

"""

*# rating\_mean\_4\_7 = rating\_mean[rating\_mean["rating"]>=4.7]*

*# comedy\_with\_4\_7 = rating\_mean\_4\_7.merge(comedy, how = 'inner', left\_on='movieId', right\_on='movieId')*

*# del comedy\_with\_4\_7['year']*

**Текстовый отчет “Старые фильмы с низким рейтингом”**

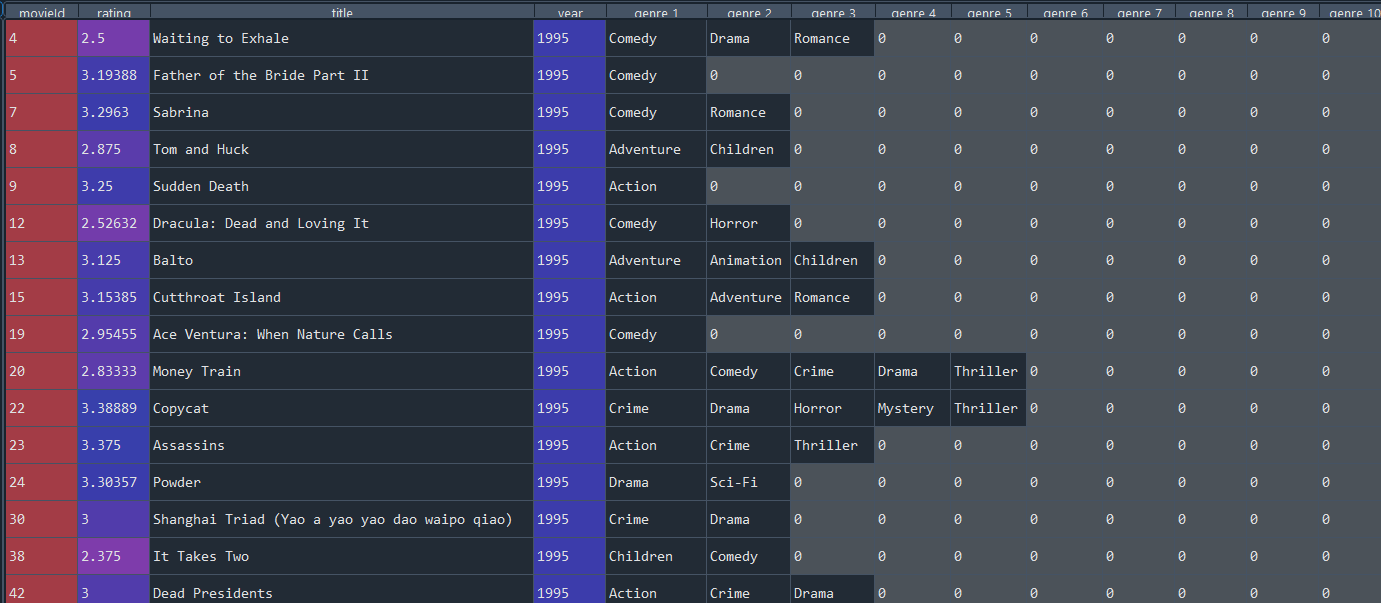
Для кого: для режиссеров фильма в стиле нулевых и девяностых, чтобы понять какие фильмы зрителям не нравятся

Названия столбцов: movieId, rating, title, year, genre 1, genre 2, genre 3, genre 4, genre 5, genre 6, genre 7, genre 8, genre 9, genre 10

Критерии:

Выбираются все фильмы с годом выпуска раньше 2000

Среди этих фильмов выбираются только те у которых средний рейтинг меньше, чем 3,4)

****

**def old movies with low ratings(movies, rating\_mean):**

"""

Данная функция показывает фильмы выпущенные раньше 2000 с средним рейтингом 3.4 и более.

**• movies:** таблица со всеми фильмами

**• rating\_mean:** таблица с средним рейтингом всех фильмов

"""

*# old\_movies=movies[movies["year"]<2000]*

*# rating\_mean\_3\_4=rating\_mean[rating\_mean["rating"]<3.4]*

*# old\_movies\_3\_4=rating\_mean\_3\_4.merge(old\_movies,how = 'inner', left\_on='movieId', right\_on='movieId')*

**Задание 3**

**def \_\_init\_\_(self, parent, controller):**

"""

@author: Алина Пахомова

Данная функция создает 3 окна: 'Что посмотреть', 'Подборки', 'Статистика'

"""

**def podborki(year, genre):**

"""

@author: Алиса Алешина

Данная функция принимает год и жанр, выбранные пользователем, и возвращает список названий нужных фильмов

"""

**def click4():**

**'''**

@author: Алиса Алешина

функция суммирует все рейтинги по данному жанру и фильмам которые понравились

пользователю, затем выдает список фильмов, предложенных пользователю,

по определенному критерию

**'''**

**def click3():**

**'''**

@author: Алиса Алешина

функция обрабатывает кнопки, на которые нажал пользователь по фильмам

**'''**

**def click2():**

**'''**

@author: Алиса Алешина

функция генерирует список фильмов которые предлагаются пользователю на выбор

**'''**

**def click1():**

**'''**

@author: Алиса Алешина

функция обрабатывает год и жанр которые выбрал пользователь

**'''**

**def podborki(year, genre, sort):**

**"""**

@author: Катя Агафонова

Функция принимает значения годов выпуска фильмов, их жанр и тип

сортировки, выбранные пользователем, и возвращает таблицу DataFrame

с названием, годом и средним рейтингом подходящих фильмов

**"""**

**def clck():**

**"""**

@author: Катя Агафонова

Функция получает данные, выбранные пользователем, вызывает функцию

преобразования этих данных в таблицу и выводит на экран год,

рейтинг и название подходящих фильмов

**"""**

**def graphics(xcrit, ycrit):**

**"""**

@author: Катя Агафонова

Функция принимает критерии по x(год или жанры) и критерии по y

(средний рейтинг или количество оценок), выбранные пользователем,

и возвращает таблицу DataFrame с нужными данными

**"""**

**def clck2():**

**"""**

@author: Катя Агафонова

Функция получает данные, выбранные пользователем, вызывает функцию

преобразования этих данных в таблицу и выводит на экран график

зависимости этих данных друг от друга

**"""**

**def close():**

**"""**

@author: Катя Агафонова

Функция удаляет текущий график с экрана. Обратите внимание, что

очищение экрана необходимо проводить после каждого нового вывода

графика!

**"""**