Untitled Notebook

Anonymous

2025 年 6 月 20 日

MovieLens 数据分析（SQL in Python 环境）

除了第一个任务 剩下任务仅显示前 20 行，表格其余信息另存在相应的 csv 文件中

[21]:
```python
import pandas as pd
import sqlite3
# # 设置 pandas 显示所有行
# pd.set_option('display.max_rows', None)
# pd.set_option('display.max_columns', None)
```

[22]:
```python
# 加载数据
movies_df = pd.read_csv("movies.csv")
ratings_df = pd.read_csv("ratings.csv")

# 创建内存数据库
conn = sqlite3.connect(":memory:")

# 写入数据库
movies_df.to_sql("movies", conn, index=False, if_exists="replace")
ratings_df.to_sql("ratings", conn, index=False, if_exists="replace")
```

```
105339
```

任务一：平均得分前 10 的电影

[23]:
```python
query1 = '''
SELECT
    m.title,
    AVG(r.rating) AS avg_rating
FROM
    ratings r
JOIN
    movies m ON r.movieId = m.movieId
GROUP BY
    m.title
ORDER BY
    avg_rating DESC
LIMIT 10;
'''
pd.read_sql_query(query1, conn)
```

```
                                              title  avg_rating
0          Young at Heart (a.k.a. Young@Heart) (2007)         5.0
1  Women on the 6th Floor, The (Les Femmes du 6èm...         5.0
2                                      Wings (1927)         5.0
3  Werckmeister Harmonies (Werckmeister harmóniák...         5.0
4                           War Photographer (2001)         5.0
5                      Waiting for 'Superman' (2010)         5.0
6                                Traviata, La (1982)         5.0
```

```
7                                Topkapi (1964)        5.0
8        Time of the Gypsies (Dom za vesanje) (1989)        5.0
9                            Three Ages (1923)        5.0
```

任务二：每个类型的平均得分前 10 的电影（Python 拆分类型）

```
merged_df = pd.merge(ratings_df, movies_df, on="movieId")
merged_df['genres'] = merged_df['genres'].str.split('|')
genre_df = merged_df.explode('genres')

genre_df.to_sql("genre_expanded", conn, index=False, if_exists="replace")

query2 = '''
WITH genre_avg AS (
    SELECT
        genres AS genre,
        title,
        AVG(rating) AS avg_rating
    FROM
        genre_expanded
    GROUP BY
        genre, title
),
ranked AS (
    SELECT *,
           ROW_NUMBER() OVER (PARTITION BY genre ORDER BY avg_rating DESC) AS rank
    FROM genre_avg
)
SELECT genre, title, avg_rating
FROM ranked
WHERE rank ⩽ 10;
'''

task2=pd.read_sql_query(query2, conn)

# 显示前 20 行
display(task2.head(20))

# 导出完整结果为 CSV 文件
[31]: task2.to_csv("task2_top10_movies_per_genre.csv", index=False)
```

```
                genre                                               title  \
0   (no genres listed)             Marco Polo: One Hundred Eyes (2015)
1   (no genres listed)                     Round Trip to Heaven (1992)
2   (no genres listed)                                    Pablo (2012)
3   (no genres listed)                                 The Take (2009)
4   (no genres listed)                     The 50 Year Argument (2014)
5   (no genres listed)                             Li'l Quinquin (     )
6   (no genres listed)               The Big Broadcast of 1936 (1935)
7               Action                                  Chase, The (1994)
8               Action  Friend Is a Treasure, A (Chi Trova Un Amico, T...
9               Action  Ghost in the Shell: Stand Alone Complex - The ...
10              Action                           Gunfighter, The (1950)
11              Action                            Heaven & Earth (1993)
12              Action              Love Exposure (Ai No Mukidashi) (2008)
13              Action                  Resident Evil: Retribution (2012)
14              Action                                    Speedy (1928)
```

```
15            Action              Star Wreck: In the Pirkinning (2005)
16            Action            Superman/Batman: Public Enemies (2009)
17         Adventure                                  Chase, The (1994)
18         Adventure                Everything's Gonna Be Great (1998)
19         Adventure  Friend Is a Treasure, A (Chi Trova Un Amico, T...

    avg_rating
0          4.0
1          4.0
2          3.5
3          3.5
4          2.5
5          2.0
6          2.0
7          5.0
8          5.0
9          5.0
10         5.0
11         5.0
12         5.0
13         5.0
14         5.0
15         5.0
16         5.0
17         5.0
18         5.0
19         5.0
```

任务三：每个用户评分最高的前 5 类型

```
[32]:
query3 = '''
WITH genre_avg AS (
    SELECT
        userId,
        genres AS genre,
        AVG(rating) AS avg_rating
    FROM
        genre_expanded
    GROUP BY
        userId, genre
),
ranked AS (
    SELECT *,
           ROW_NUMBER() OVER (PARTITION BY userId ORDER BY avg_rating DESC) AS rank
    FROM genre_avg
)
SELECT userId, genre, avg_rating
FROM ranked
WHERE rank ⩽ 5
'''
task3 = pd.read_sql_query(query3, conn)

display(task3.head(20))
task3.to_csv("task3_top5_genre_user1.csv", index=False)
```

```
   userId         genre  avg_rating
0       1         Crime    4.209677
```

```
1        1          War    4.200000
2        1    Animation    4.000000
3        1    Film-Noir    4.000000
4        1      Musical    4.000000
5        2    Animation    4.500000
6        2        Drama    4.363636
7        2     Children    4.333333
8        2        Crime    4.333333
9        2      Fantasy    4.250000
10       3  Documentary    5.000000
11       3      Mystery    4.250000
12       3        Crime    4.000000
13       3       Horror    4.000000
14       3         IMAX    4.000000
15       4    Animation    4.750000
16       4          War    4.562500
17       4      Mystery    4.400000
18       4        Drama    4.342105
19       4     Children    4.333333
```

任务三：每个用户评分最高的前 5 类型(其他表现新形势)

[33]:
```python
top5_df = pd.read_sql_query(query3, conn)

# 转换为透视表形式：userId 为行，genre 为列，值为 avg_rating
pivot_table = top5_df.pivot(index='userId', columns='genre', values='avg_rating')

# Nah填入‘-’
pivot_table = pivot_table.fillna('-')

display(pivot_table.head(20))
pivot_table.to_csv("task3_top5_genre_user2.csv", index=False)
```

```
 (no genres listed)    Action Adventure Animation  Children Comedy  \
userId
1                       -         -         -       4.0         -      -
2                       -         -         -       4.5  4.333333      -
3                       -         -         -         -         -      -
4                       -         -         -      4.75  4.333333      -
5                       -         -         -  4.095238  3.904762      -
6                       -         -         -         -         -      -
7                       -         -         -         -         -      -
8                       -         -         -  4.333333  4.166667      -
9                       -         -         -         -         -      -
10                      -         -         -      4.25      4.25      -
11                      -         -    3.8125  3.846154         -      -
12                      -         -       4.0       4.5       4.5      -
13                      -         -         -         -       3.5    3.6
14                      -         -         -       4.0  4.333333      -
15                      -         -         -       5.0       5.0      -
16                      -  4.333333         -         -         -      -
17                      -         -  4.293103       4.7     4.375      -
18                      -         -         -    3.9375         -      -
19                      -         -         -         -         -      -
20                      -         -  3.928571         -         -      -

genre      Crime Documentary     Drama   Fantasy Film-Noir    Horror  \
userId
```

```
1       4.209677           -         -         -       4.0         -
2       4.333333           -  4.363636      4.25         -         -
3            4.0         5.0         -         -         -       4.0
4              -           -  4.342105         -         -         -
5              -           -         -   3.84375         -         -
6       4.363636           -         -         -         -       4.5
7       3.854839         4.0         -         -       5.0         -
8              -           -  4.133333       4.0         -         -
9       3.086957         3.0         -         -         -         -
10           4.0           -         -         -         -       4.0
11             -           -         -         -         -       4.0
12             -           -         -     4.125         -         -
13             -           -  3.727273     3.625         -         -
14           5.0           -         -       4.0         -         -
15             -           -         -         -       5.0         -
16             -           -         -         -         -  4.666667
17             -           -         -         -         -         -
18             -           -         -  3.692308       4.0         -
19             -        4.25  3.858108         -         -         -
20        3.9375           -         -         -         -         -

genre       IMAX   Musical   Mystery   Romance    Sci-Fi  Thriller       War  \
userId
1              -       4.0         -         -         -         -       4.2
2              -         -         -         -         -         -         -
3            4.0         -      4.25         -         -         -         -
4              -         -       4.4         -         -         -    4.5625
5       4.277778  4.090909         -         -         -         -         -
6              -       4.5         -  4.366667  4.333333         -         -
7              -       5.0  3.861111         -         -         -         -
8              -         -  4.333333         -         -         -         -
9            4.0         -         -         -         -  2.866667       3.0
10             -         -         -         -         -         -       4.0
11             -         -         -         -  3.904762         -  3.666667
12             -       4.5         -         -         -         -         -
13             -         -     3.875         -         -         -         -
14             -       5.0         -         -         -         -         -
15           5.0       5.0         -         -         -         -         -
16             -         -       4.5         -  4.454545      4.25         -
17             -         -       4.5         -  4.317073         -         -
18             -         -  4.033333  3.892857         -         -         -
19             -         -  3.833333         -         -         -  4.117647
20      4.333333         -  4.214286         -         -  4.166667         -

genre  Western
userId
1            -
2            -
3            -
4            -
5            -
6            -
7            -
8            -
9            -
10           -
11           -
12           -
```

```
13            -
14            -
15            -
16            -
17            -
18            -
19          4.0
20            -
```

任务四：每个用户观影次数最多的前 5 类型

```
query4 = '''
WITH genre_counts AS (
    SELECT
        userId,
        genres AS genre,
        COUNT(*) AS view_count
    FROM
        genre_expanded
    GROUP BY
        userId, genre
),
ranked AS (
    SELECT *,
           ROW_NUMBER() OVER (PARTITION BY userId ORDER BY view_count DESC) AS rank
    FROM genre_counts
)
SELECT userId, genre, view_count
FROM ranked
WHERE rank ⩽ 5
'''
task4 = pd.read_sql_query(query4, conn)

display(task4.head(20))
task4.to_csv("task4_top5_viewcount_user1.csv", index=False)
```

[34]:

```
   userId      genre  view_count
0       1     Action          46
1       1      Drama          45
2       1   Thriller          43
3       1  Adventure          31
4       1     Comedy          31
5       2   Thriller          12
6       2     Comedy          11
7       2      Drama          11
8       2  Adventure          10
9       2     Action           9
10      3      Drama          36
11      3     Comedy          35
12      3    Romance          22
13      3   Thriller          21
14      3     Action          13
15      4      Drama          76
16      4     Comedy          46
17      4    Romance          37
18      4      Crime          18
19      4   Thriller          18
```

```
       top5_vc = pd.read_sql_query(query4, conn)
       pivot_table = top5_vc.pivot(index='userId', columns='genre', values='view_count')
       pivot_table = pivot_table.fillna('-')

       display(pivot_table.head(20))
[35]:  pivot_table.to_csv("task4_top5_viewcount_user12.csv", index=False)
```

```
 Action Adventure Animation Children Comedy Crime Drama Fantasy Horror  \
userId
1        46.0      31.0         -        -   31.0     -   45.0       -      -
2         9.0      10.0         -        -   11.0     -   11.0       -      -
3        13.0         -         -        -   35.0     -   36.0       -      -
4           -         -         -        -   46.0  18.0   76.0       -      -
5           -      22.0      21.0     21.0   45.0     -      -       -      -
6           -      14.0         -        -   28.0     -   31.0       -      -
7        95.0      53.0         -        -   46.0     -      -       -      -
8           -      12.0         -        -   24.0     -   30.0       -      -
9        39.0         -         -        -   53.0     -   49.0       -      -
10        4.0         -         -        -   10.0     -   11.0       -      -
11       29.0      32.0         -        -   36.0     -   32.0       -      -
12        4.0         -         -        -    8.0     -   10.0     4.0      -
13        8.0       7.0         -        -   10.0     -   11.0       -      -
14        6.0       7.0         -        -   12.0     -    6.0       -      -
15          -      16.0         -        -   26.0     -   37.0       -   21.0
16       36.0         -         -        -   25.0     -   23.0       -      -
17       61.0      58.0         -        -   63.0     -   49.0       -      -
18       37.0         -         -        -   28.0  24.0   32.0       -      -
19       29.0      24.0         -        -   18.0     -   74.0       -      -
20       11.0         -         -      9.0   16.0     -   39.0       -      -

genre  IMAX Musical Mystery Romance Sci-Fi Thriller War
userId
1         -       -       -       -      -     43.0   -
2         -       -       -       -      -     12.0   -
3         -       -       -    22.0      -     21.0   -
4         -       -       -    37.0      -     18.0   -
5         -       -       -    21.0      -        -   -
6         -       -       -    15.0      -     13.0   -
7         -       -       -       -   63.0     71.0   -
8         -       -       -    17.0      -     16.0   -
9         -       -       -    24.0      -     45.0   -
10        -     3.0       -     8.0      -        -   -
11        -       -       -       -      -     27.0   -
12        -       -       -     6.0      -        -   -
13        -       -       -       -      -     15.0   -
14        -       -       -       -      -      8.0   -
15        -       -       -       -      -     20.0   -
16        -       -       -       -   22.0     36.0   -
17        -       -       -       -   41.0        -   -
18        -       -       -       -      -     47.0   -
19        -       -       -       -      -     20.0   -
20        -       -       -    19.0      -        -   -
```