Python 程序设计与数据科学导论 期中大作业 2200017701 李思涵

Task 1:

Part 1

首先将数据合并，获得(1000209, 10)的总数据

这里采用先分性别计算每部电影平均值，再选出评分个数≥300 的顺序，因为前者的结果后面还要用到。分别呈现男性、女性评分最高的电影和评分差异最大的电影，按评分/分差高低排序并输出电影 title，共 4 个输出如下：

这是男性平均打分最高的20部电影

0	Usual Suspects, The (1995)
1	Star Wars: Episode IV - A New Hope (1977)
2	Shawshank Redemption, The (1994)
3	Schindler's List (1993)
4	Close Shave, A (1995)
5	Dr. Strangelove or: How I Learned to Stop Worr...
6	Godfather, The (1972)
7	Rear Window (1954)
8	Casablanca (1942)
9	Maltese Falcon, The (1941)
10	Sunset Blvd. (a.k.a. Sunset Boulevard) (1950)
11	Wrong Trousers, The (1993)
12	One Flew Over the Cuckoo's Nest (1975)
13	Raiders of the Lost Ark (1981)
14	Lawrence of Arabia (1962)
15	Third Man, The (1949)
16	Godfather: Part II, The (1974)
17	Great Escape, The (1963)
18	Seven Samurai (The Magnificent Seven) (Shichin...
19	Double Indemnity (1944)

这是女性平均打分最高的20部电影

0	Usual Suspects, The (1995)
1	Shawshank Redemption, The (1994)
2	Schindler's List (1993)
3	Wallace & Gromit: The Best of Aardman Animatio...
4	Close Shave, A (1995)
5	Philadelphia Story, The (1940)
6	Rear Window (1954)
7	It Happened One Night (1934)
8	Some Like It Hot (1959)
9	Sunset Blvd. (a.k.a. Sunset Boulevard) (1950)
10	Notorious (1946)
11	Mr. Smith Goes to Washington (1939)
12	Wrong Trousers, The (1993)
13	To Kill a Mockingbird (1962)
14	Third Man, The (1949)
15	Grand Day Out, A (1992)
16	Seven Samurai (The Magnificent Seven) (Shichin...
17	Strangers on a Train (1951)
18	Life Is Beautiful (La Vita Ã¨ bella) (1997)
19	Sixth Sense, The (1999)

这是女性打分高于男性最多的20部电影

0	Sense and Sensibility (1995)
1	French Kiss (1995)
2	Beverly Hills Cop III (1994)
3	Gone with the Wind (1939)
4	Mary Poppins (1964)
5	Sound of Music, The (1965)
6	Dirty Dancing (1987)
7	Grease (1978)
8	Grease 2 (1982)
9	Anastasia (1997)
10	Hope Floats (1998)
11	Jewel of the Nile, The (1985)
12	Patch Adams (1998)
13	Jumpin' Jack Flash (1986)
14	Never Been Kissed (1999)
15	Rocky Horror Picture Show, The (1975)
16	Color Purple, The (1985)
17	Little Shop of Horrors, The (1960)
18	Skulls, The (2000)
19	Steel Magnolias (1989)

这是男性打分高于女性最多的20部电影

0	From Dusk Till Dawn (1996)
1	Dumb & Dumber (1994)
2	Barb Wire (1996)
3	Cable Guy, The (1996)
4	Good, The Bad and The Ugly, The (1966)
5	Evil Dead II (Dead By Dawn) (1987)
6	Akira (1988)
7	Beavis and Butt-head Do America (1996)
8	Exorcist, The (1973)
9	Rocky II (1979)
10	Rocky III (1982)
11	Fright Night (1985)
12	Longest Day, The (1962)
13	Animal House (1978)
14	Caddyshack (1980)
15	For a Few Dollars More (1965)
16	Porky's (1981)
17	Big Trouble in Little China (1986)
18	Me, Myself and Irene (2000)
19	Kentucky Fried Movie, The (1977)

Part 2

这里的关键在于如何定义男/女偏好程度。在这里我选择的是统计评分个数超过 300 的电影。

评分个数≥300的电影

iter and split

获得所有genres的list

遍历genres，by male and female, as input to func

```
def get_genre_ratings(genre, gender)
    all genre data used here
    "popularity score":
    rating.mean()^3 * number of ratings
```

同时考虑了观影评分和观影次数，综合分析性别偏好

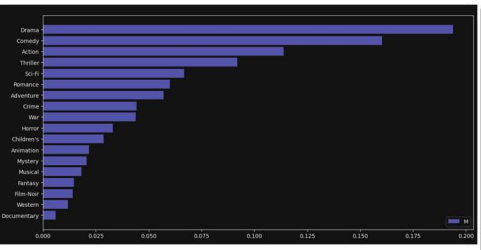
这里曾经尝试过采用特殊的genre处理方法，比如一个电影是(genre1|genre2|genre3)的形式，那么它的评分将被均摊给3个genre。但是效果不太理想，故最终没有采用。

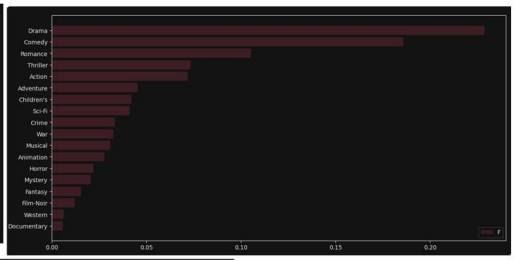
popularity score for each genre for both genders:

	genre	M	F
0	Drama	1.370788e+07	5.300042e+06
1	Animation	1.535917e+06	6.439480e+05
2	Children's	2.031498e+06	9.713498e+05
3	Musical	1.278359e+06	7.192747e+05
4	Romance	4.236893e+06	2.441724e+06
5	Comedy	1.133611e+07	4.306598e+06
6	Action	8.047949e+06	1.663750e+06
7	Adventure	4.030168e+06	1.052691e+06

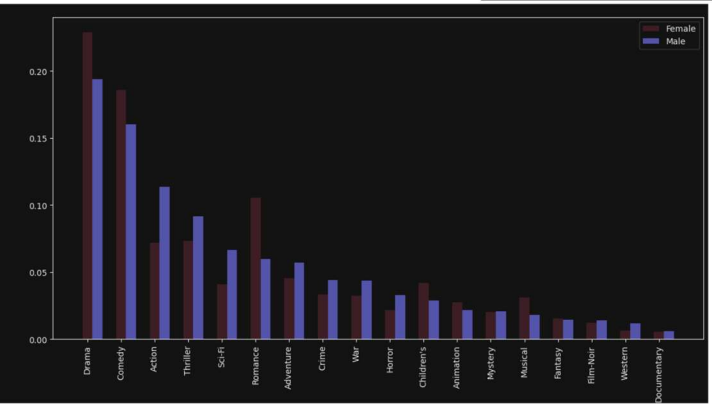
此处的样例省略了8-17的数据

偏好程度（Popularity Score）





相对差值，表现了性别偏好程度。例如图中可看出女性比男性更偏爱 Romance 电影约 27%。这里得出的结果还是较为符合直觉的。



分性别绘制的直方图和共同绘制的直方图

Task 2:

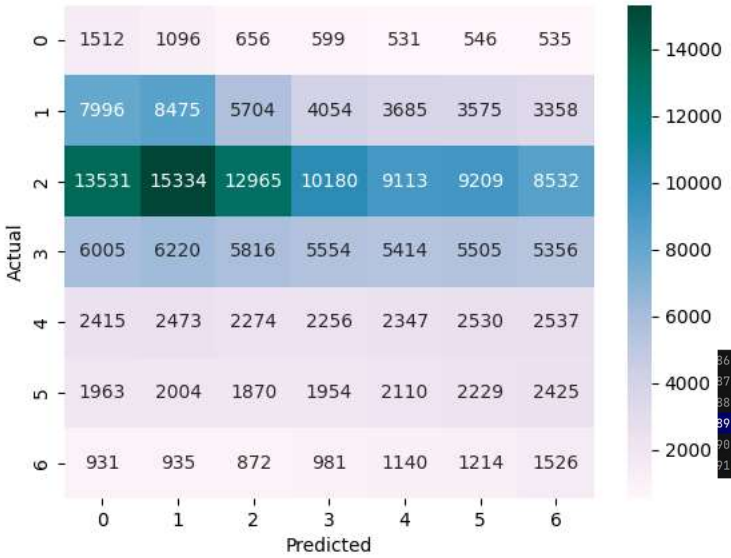
经过多次尝试，最终选择了随机森林、KNN、naive Bayes 等模型。有些模型可以通过调参达到更高的准确度，但是根据 confusion matrix 可以看出其结果基本上会受到样本数据 bias 更大的影响（这个 bias 即指 25-34 岁的样本远远多于其他年龄段）

```
Under 18:27211
18-24:183536
25-34:395556
35-44:199003
45-50:83633
50-55:72490
56+:38780
```

我尝试过调用随机过采样和随机欠采样，然而准确率反而下降，虽然使用多个模型测试后都得出对于 25-34 以外的全年龄段的预测准确率上升的结果，然而因为测试集中 25-34 占比也过高，总准确率仍然下跌。分析热力图可知调整采样后一定量的 25-34 岁样本被预测为其他年龄段，并导致总准确率大幅减少。

感觉如果结合 genre 可能能有一定的提升，但是没来得及尝试。

右图是使用默认参数进行的随机欠采样，与下面最终选择的方法对比可以明显看出前述的区别。（使用随机森林模型）

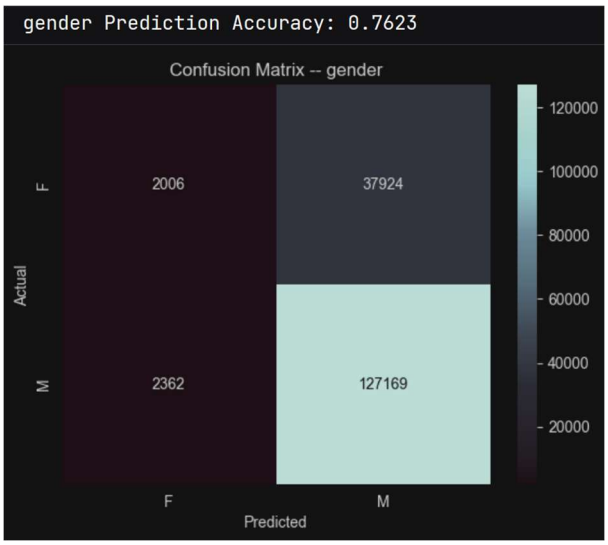


随机欠采样结果（年龄段预测）

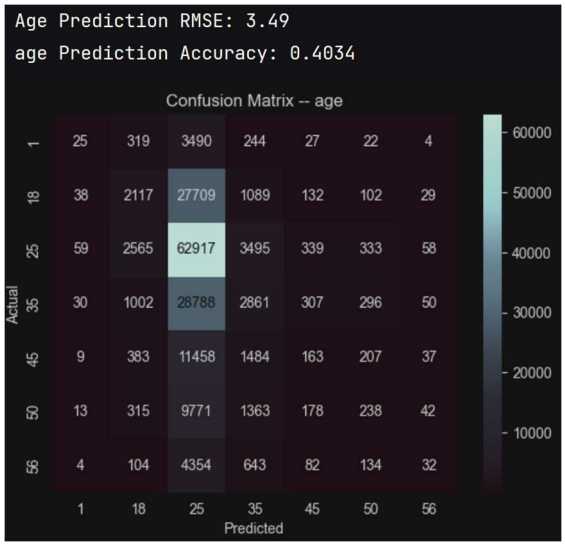
下面的是尝试随机欠采样时的代码和调参

```
from imblearn.under_sampling import RandomUnderSampler
strategy = {
    0: 21736, 1: 70664, 2: 103747,
    3: 73587, 4: 47704, 5: 44413, 6: 31181}
rus = RandomUnderSampler(strategy=strategy)
X2, y2 = rus.fit_resample(X_train, y_train[:, 1])
```

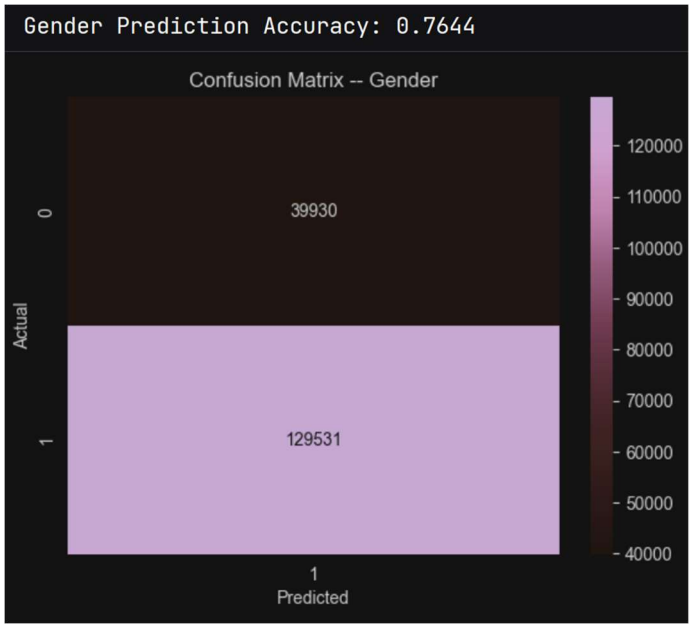
随机森林+PCA

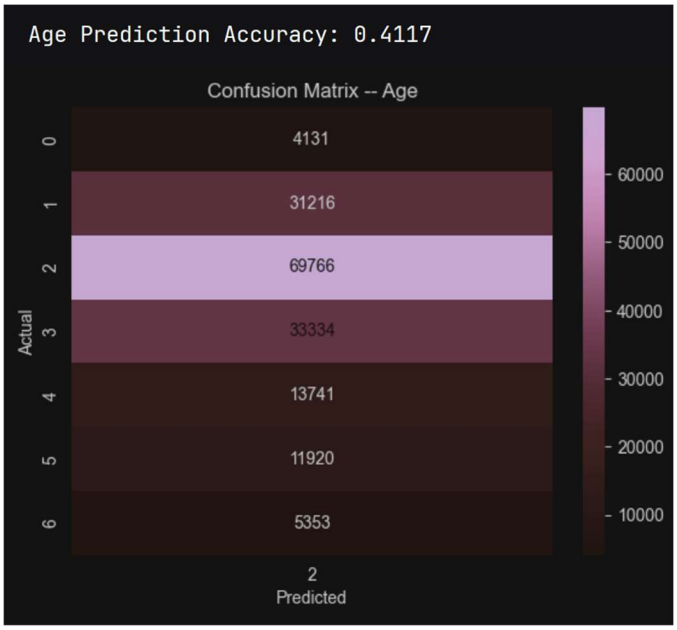


这里进行了有序回归，RMSE 的单位是（岁）

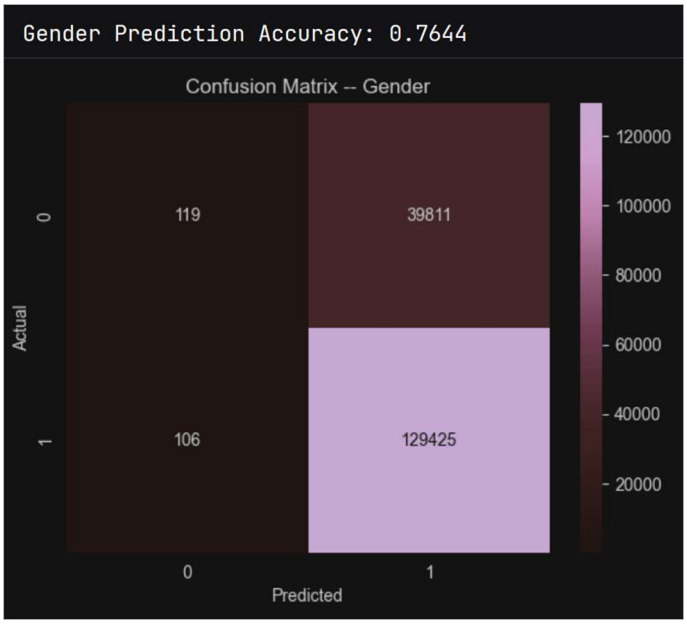


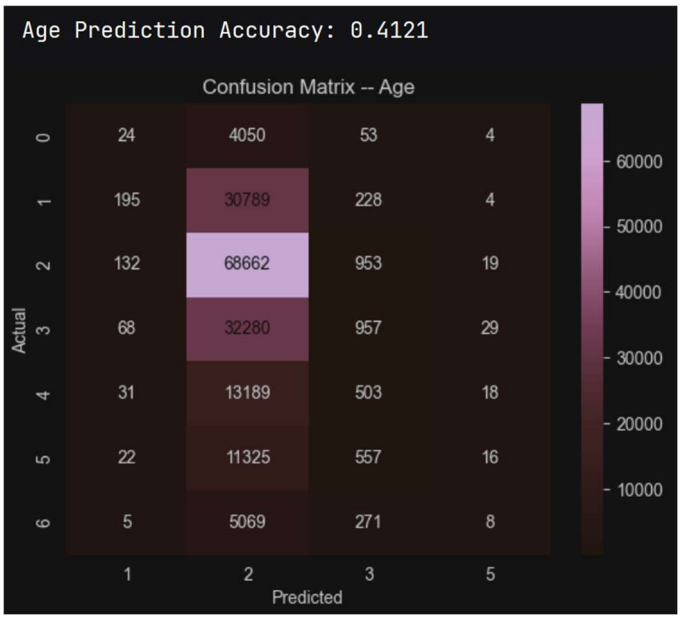
纯贝叶斯，性别预测结果中只有男性。(0 为女性，1 为男性)。
年龄预测结果中只有 25-34 岁。(0-6 代表 7 个年龄段)





KNN 模型，是三种中准确度相对最高的。年龄预测中缺失了 Under 18, 45-50, 56+





在最开始的 cell，我对 customized PCA、贝叶斯，以及后面要频繁用到的可视化和读取数据等进行了封装，因此后面的代码相对较为整洁简练。

```
1  # Task 2
2
3  from numpy import array, sqrt, pi, exp
4  import seaborn as sns
5  import matplotlib.pyplot as plt
6  from sklearn.metrics import accuracy_score, mean_squared_error
7  from sklearn.model_selection import train_test_split
8  # customized GaussianNB
9> class MyGaussianNB:...
46 # customized PCA
47> class MyPCA:...
67 # 集成准确度和混淆矩阵输出
68> class Report:...
91 # 加载数据
92> def load(mode='pandas'):...
```

Task 3:

Part 1

先统计每个用户对每种电影风格的打分的平均分，以及每种电影风格观看过的数量

```
18 res_ratings.head()
Executed in 4s, 2 May at 14:54:33
```

5 rows × 18 columns pd.Dataframe

user_id	rating_Drama	rating_Comedy	rating_Action	rating_War	rating_Romance
2	3.898734	3.560000	3.500000	3.733333	3.7083
5	3.096154	3.410714	2.612903	3.500000	3.1000
8	3.814815	3.888889	4.027778	4.200000	4.0666
9	3.888889	3.740741	3.500000	4.142857	3.5555
10	4.181034	4.136612	3.913580	3.833333	4.3731

```
3 res_count.head()
Executed in 9ms, 2 May at 14:54:45
```

5 rows × 18 columns pd.Dataframe

user_id	count_Drama	count_Comedy	count_Action	count_War	count_Romance	co
2	79	25	56	15.0	24	
5	104	56	31	6.0	30	
8	108	18	36	10.0	30	
9	54	27	20	7.0	18	
10	116	183	81	24.0	67	

再将每个风格组合[genre1|genre2|…]视作独立的一个风格，进行统计，共 301 种

5 rows × 301 columns pd.Dataframe

user_id	count_Drama	count_Comedy	count_Action|Drama|War	co
2	28.0	4.0	4.0	
5	43.0	22.0	2.0	
8	44.0	1.0	4.0	
9	23.0	7.0	2.0	
10	38.0	67.0	3.0	

假设：对于这 301 种风格，我们有理由认为观看次数最多的一定就是这位用户最喜欢的风格，而风格的喜好程度与用户对该风格电影的平均打分并没有显著相关

分析如下：1. 只有一个人喜欢某类电影的情况下，他才会频繁地观看它们。看得越多，越说明他对这类电影有兴趣，主动地去寻找相似的电影。

2. 看得多的电影与看得少的电影相比，在客观水平相同的情况下，前者更可能得到更低的评分，因为审美水平和要求都随着观影数量而提升。看得少的电影更容易因为观看时的新鲜感而获得更高的评分，但这不代表该用户喜欢这类电影。同时，评分还受到各电影水平的强烈影响，与类型的关联相对弱得多。也就是说，平均评分和类型喜好之间无法找出显著的相关性，用作分析材料不太恰当。

3. 作为一个总观影数达到 100 的用户，完全没有看过的类型基本可以断定是其不感兴趣、

拒绝观看的，因此将其没看过的类型认定为不喜欢的类型是有依据的，至少可以认为截至数据产生的时间，没看过的（或看得最少的）类型是用户最不喜欢的类型。
在此基础上，产生了下列随机示例：

```
# 示例，测试
# 随机抽取10个用户，输出其最喜欢的电影类型和最不喜欢的电影类型
import random
random.seed(42)
random_users = random.sample(list(res_count.index), 10)
for i in random_users:
    print('用户', f"\033[3;34;40m{i}\033[0m", '最喜欢的电影类型:', f"\033[1;32;
     40m{res_count['最喜欢'][i]}\033[0m", '最不喜欢的电影类型:', f"\033[9;31;
     40m{least_watched_dict[i]}\033[0m")
```

Executed in 4ms, 3 May at 10:25:47

```
用户 2533 最喜欢的电影类型: Comedy|Romance 最不喜欢的电影类型:
Crime/Horror/Film-Noir/Documentary
用户 2194 最喜欢的电影类型: Action|Sci-Fi 最不喜欢的电影类型:
Children's/Documentary
用户 2025 最喜欢的电影类型: Action|Adventure|Sci-Fi 最不喜欢的电影类型:
Animation
用户 1271 最喜欢的电影类型: Comedy 最不喜欢的电影类型: Mystery
用户 958 最喜欢的电影类型: Drama 最不喜欢的电影类型: Fantasy
用户 6003 最喜欢的电影类型: Drama 最不喜欢的电影类型: Animation/Documentary
```

与原始数据进行直观对照，可以发现这种判断方法还是较为准确的。

Part 2

接下来进行代表性的电影推荐，继续以 10 个随机用户为目标
一个电影的代表性可以以其收到的平均评分衡量
采用了下述算法

```
# 对每个用户，优先推荐 3 部其最喜爱的风格的电影中，评分最高且没有看过的电影，再推荐 3 部其第二喜爱的风格的电影中，评分最高的没有看过的电影，如果数量不够则依次递推
# 这 2*3 部电影是从一定范围的高评分电影中随机选取的，再取其中的前 5 部作为结果。
```

随机性可以保证推荐系统的多样性，
且能够增大容错程度（避免单调地推荐部分电影）

部分结果展示如右图：

```
]

为用户6003推荐的电影：
[
Inheritors, The (Die Siebtelbauern) (1998)
Apple, The (Sib) (1998)
I Am Cuba (Soy Cuba/Ya Kuba) (1964)
His Girl Friday (1940)
General, The (1927)
]

为用户4899推荐的电影：
[
Smashing Time (1967)
Monty Python and the Holy Grail (1974)
Palm Beach Story, The (1942)
Mamma Roma (1962)
Foreign Student (1994)
]
```

Part 3

封装自定义雷达图类如下：

```
class RadarGraph:
    def __init__(self, data, labels, title, r=1):...
    def draw_radar(self):...
```

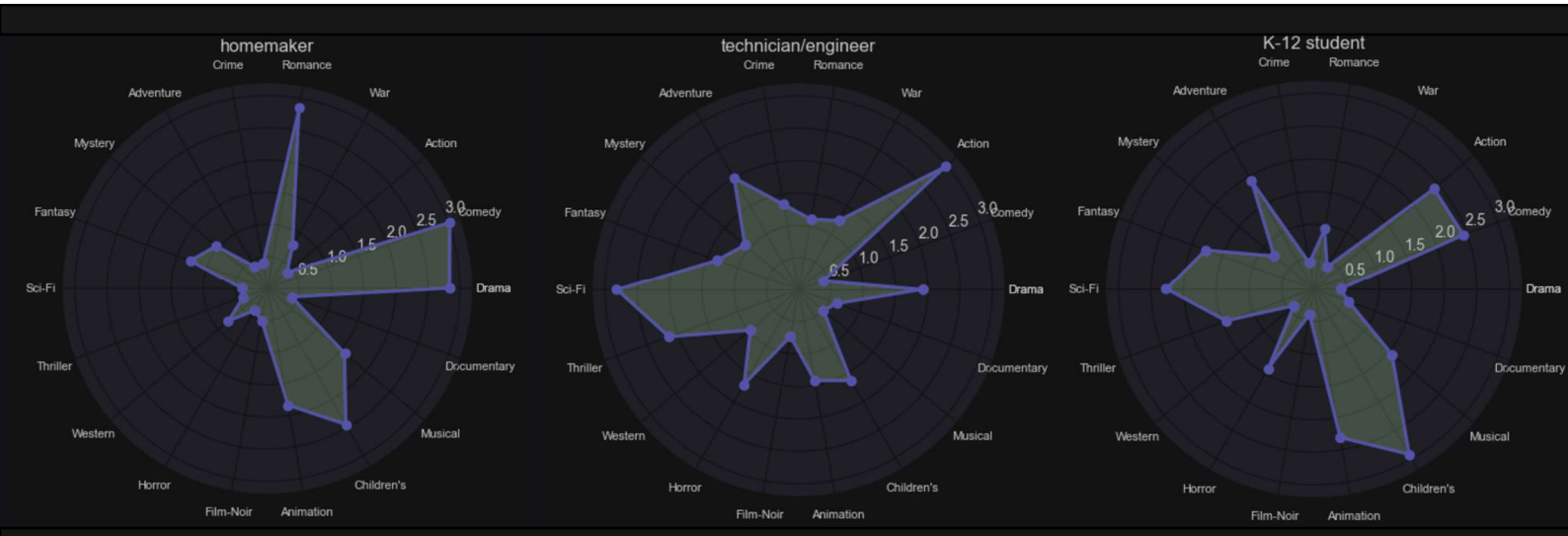
这里为每种职业各绘制了一份描述对 18 个基本电影风格的偏好程度的雷达图。

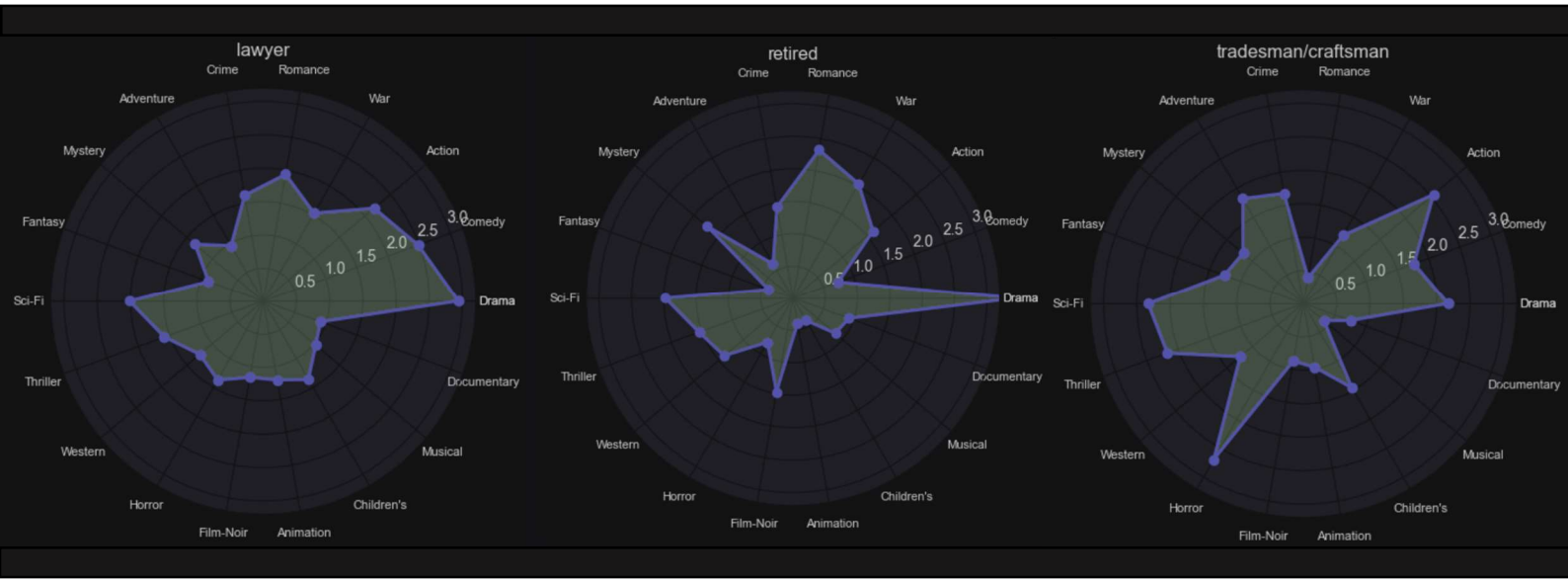
关键在于如何定义偏好程度。这里我采用了如下方法：

1. 计算各职业观看各类电影的总数量
2. 计算各职业观看每类电影的数量占该职业观看的全部电影总数的比例（如假设职业 A 群体共观看了 X 类电影 1000 部，A 群体观看所有类型电影共 10000 部，那么对 A 来说 X 的比例就是 0.1
3. 因为考察的变量是“偏好”，所以要在不同职业间进行对比。这里的方法是将上一步中计算得到的某个职业观看某种类型电影的比例减去所有职业中该比例的最小值，取差值并进行对数运算等操作，最终将不同职业群体的“偏好”呈现出来，并在一定的数据处理后绘制雷达图。代码如下：

```
# 统计各职业人群看过的各类电影的个数
re_indexed = res_ind_count.reset_index()
occ_count = re_indexed.merge(users[['user_id','occupation']])
occ_count = occ_count.groupby('occupation').sum()
occupations = data[['occupation','occ_desc']].drop_duplicates().set_index
 ('occupation')
occ_count.drop('user_id', axis=1, inplace=True)
# 统计各职业人群看过的各类电影的占比
occ_count_ratio = occ_count.apply(lambda x: x/x.sum(), axis=1)
# 统计各职业人群看过的各类电影的占比与全体看过的各类电影的占比的最小值的差值，并做放大处理
min_count_ratio = occ_count_ratio.apply(lambda x: x.min()-0.005, axis=0)
occ_count_ratio_diff = (occ_count_ratio - min_count_ratio).mul(1e2*3).round(2)
# 对各职业人群对各类电影的偏好进行雷达图绘制
for i in occ_count_ratio_diff.index:
    RadarGraph(np.log(np.array(occ_count_ratio_diff.loc[i].tolist())), [i[10:]
      for i in occ_count_ratio_diff.columns.tolist()], occupations.loc[i]
     .tolist()[0],r=3.2).draw_radar()
```

挑选输出中有代表性的几张雷达图附于下方



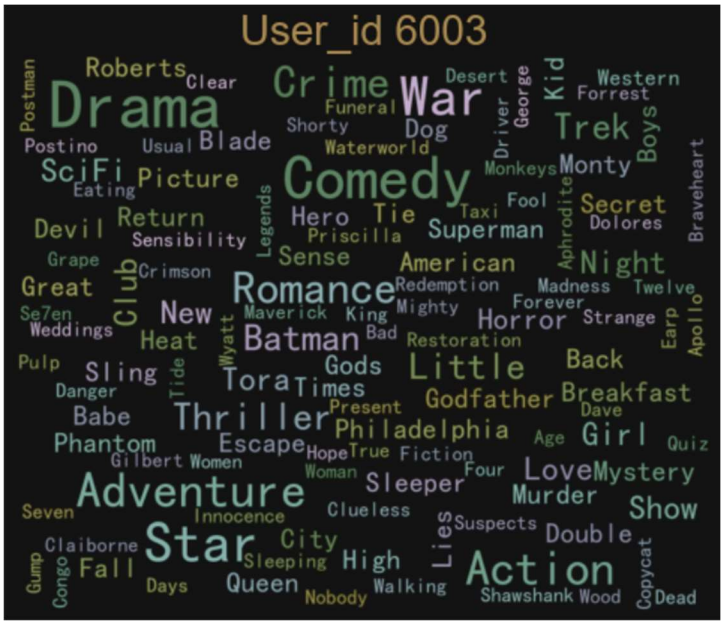


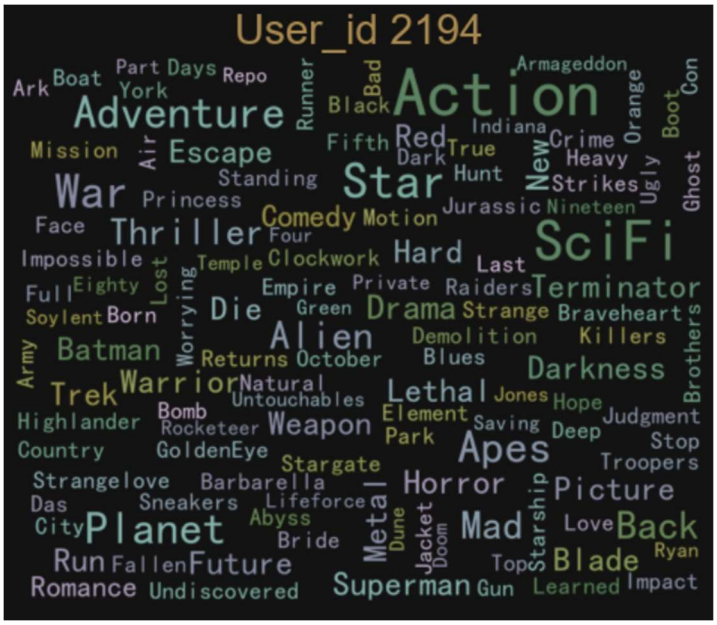
Part 4

使用 jieba 分词和 wordcloud 生成用户偏好词云

仍然对随机抽取的 10 个用户进行操作，对其看过的电影的标题进行分词，提高标题权重后和拆分过的电影 genre 一起制作词云，这样可以达到大体风格一目了然、具体细节充实丰满的效果。

部分词云展示如下：





可以看出不同风格的用户之间的区别在词云中体现得还是很明显的。

作业报告结束，助教老师您辛苦了😊

2200017701 李思涵

2023.5.3