人工智能实践课程项目一
音乐推荐系统

项目背景：

此推荐系统项目是基于人工智能课程理论知识学习后，学生通过小组协作的形式将学到的理论知识应用于实践中，以实现理论和实践相结合的一次实践项目。各小组成员用相同的数据集不同的理论知识方法实现各不相同的音乐推荐系统。

据报道[1]，中国有超过9.77亿人每周都听音乐，而66%的人通过流媒体来听音乐。为了给用户提供更好的体验，如何为用户推荐喜爱的音乐就变得非常重要。本项目使用的数据集来自Last.fm音乐网站[2]，数据集在2011推荐系统会议(ACM RecSys)中发布。

There are basically three types of recommender systems:

Demographic Filtering- They offer generalized recommendations to every user, based on movie popularity and/or genre. The System recommends the same movies to users with similar demographic features. Since each user is different , this approach is considered to be too simple. The basic idea behind this system is that movies that are more popular and critically acclaimed will have a higher probability of being liked by the average audience.

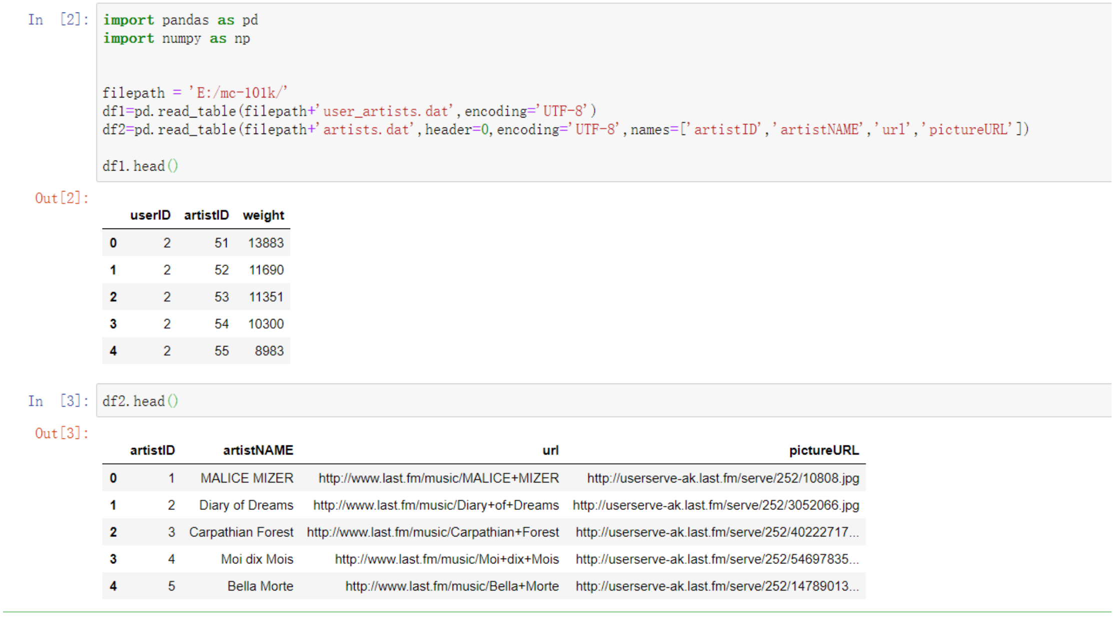
Content Based Filtering- They suggest similar items based on a particular item. This system uses item metadata, such as genre, director, description, actors, etc. for movies, to make these recommendations. The general idea behind these recommender systems is that if a person liked a particular item, he or she will also like an item that is similar to it.

Collaborative Filtering- This system matches persons with similar interests and provides recommendations based on this matching. Collaborative filters do not require item metadata like its content-based counterparts.

因初次接触人工智能理论基础学习不扎实，所以此次课程项目我们小组采用的是较为简单的 **Demographic Filtering(**基于人口统计学的过滤方法**)**

下面是我们的实践过程：

读取数据集并处理数据：



In [2]:
```
import pandas as pd
import numpy as np


filepath = 'E:/mc-101k/'
df1=pd.read_table(filepath+'user_artists.dat',encoding='UTF-8')
df2=pd.read_table(filepath+'artists.dat',header=0,encoding='UTF-8',names=['artistID','artistNAME','url','pictureURL'])

df1.head()
```

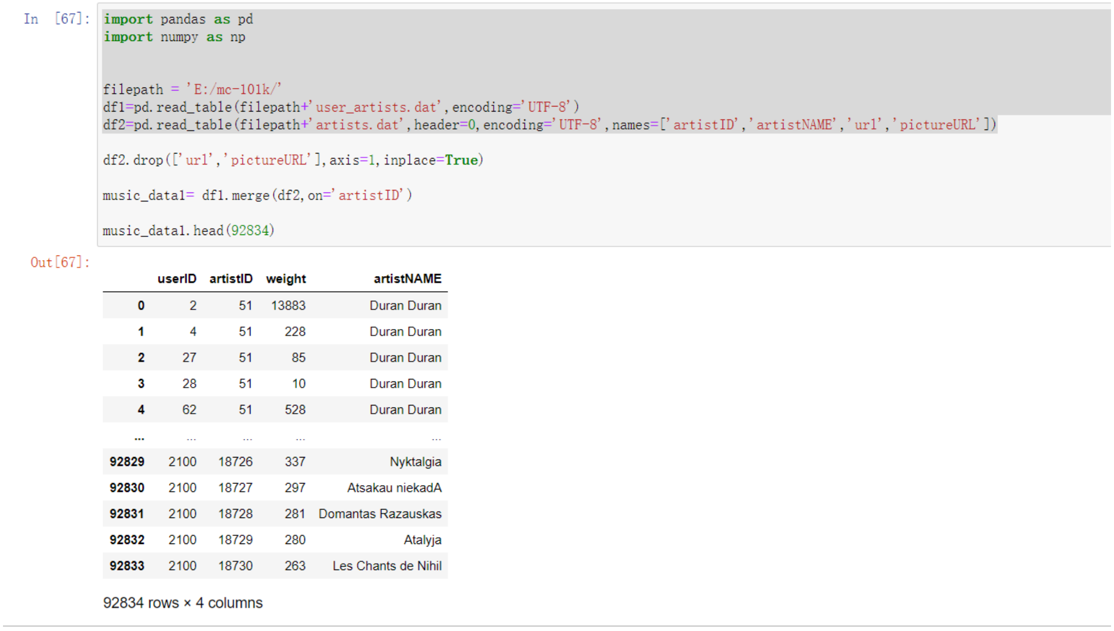
Out[2]:

	userID	artistID	weight
0	2	51	13883
1	2	52	11690
2	2	53	11351
3	2	54	10300
4	2	55	8983

In [3]:
```
df2.head()
```

Out[3]:

	artistID	artistNAME	url	pictureURL
0	1	MALICE MIZER	http://www.last.fm/music/MALICE+MIZER	http://userserve-ak.last.fm/serve/252/10808.jpg
1	2	Diary of Dreams	http://www.last.fm/music/Diary+of+Dreams	http://userserve-ak.last.fm/serve/252/3052066.jpg
2	3	Carpathian Forest	http://www.last.fm/music/Carpathian+Forest	http://userserve-ak.last.fm/serve/252/40222717...
3	4	Moi dix Mois	http://www.last.fm/music/Moi+dix+Mois	http://userserve-ak.last.fm/serve/252/54697835...
4	5	Bella Morte	http://www.last.fm/music/Bella+Morte	http://userserve-ak.last.fm/serve/252/14789013...

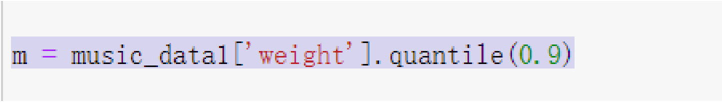


```
In  [67]: import pandas as pd
          import numpy as np


          filepath = 'E:/mc-101k/'
          df1=pd.read_table(filepath+'user_artists.dat',encoding='UTF-8')
          df2=pd.read_table(filepath+'artists.dat',header=0,encoding='UTF-8',names=['artistID','artistNAME','url','pictureURL'])

          df2.drop(['url','pictureURL'],axis=1,inplace=True)

          music_data1= df1.merge(df2,on='artistID')

          music_data1.head(92834)
```

Out[67]:

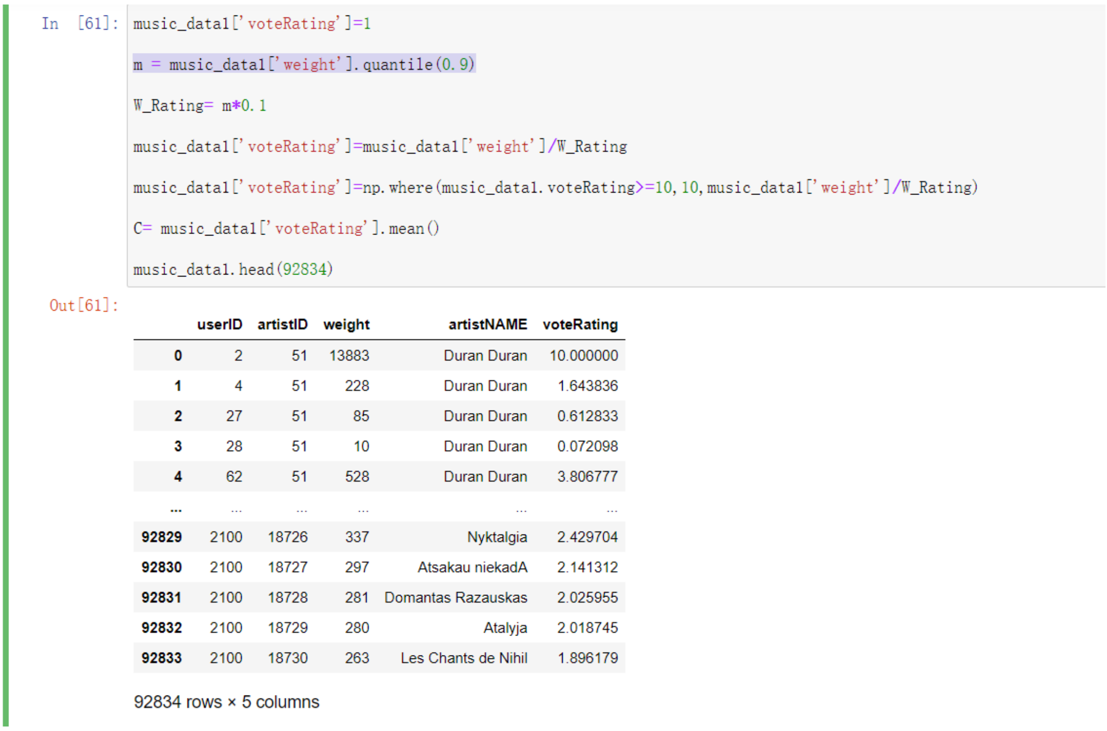
	userID	artistID	weight	artistNAME
0	2	51	13883	Duran Duran
1	4	51	228	Duran Duran
2	27	51	85	Duran Duran
3	28	51	10	Duran Duran
4	62	51	528	Duran Duran
...	...	...	...	...
92829	2100	18726	337	Nyktalgia
92830	2100	18727	297	Atsakau niekadA
92831	2100	18728	281	Domantas Razauskas
92832	2100	18729	280	Atalyja
92833	2100	18730	263	Les Chants de Nihil

92834 rows × 4 columns



```
m = music_data1['weight'].quantile(0.9)
```

通过分位数计算weight 的大致分布，并通过一定比例计算 VoteRating 值



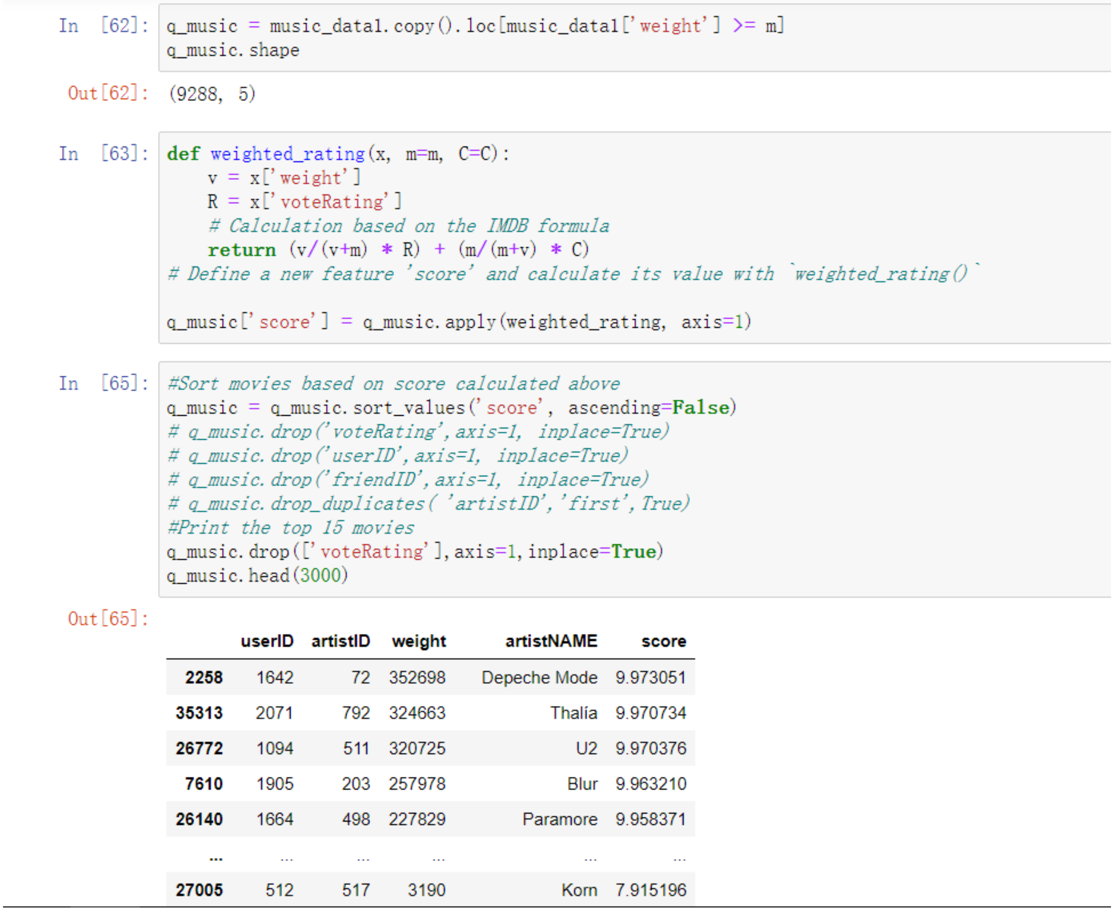
```
In  [61]: music_data1['voteRating']=1

          m = music_data1['weight'].quantile(0.9)

          W_Rating= m*0.1

          music_data1['voteRating']=music_data1['weight']/W_Rating

          music_data1['voteRating']=np.where(music_data1.voteRating>=10,10,music_data1['weight']/W_Rating)

          C= music_data1['voteRating'].mean()

          music_data1.head(92834)
```

Out[61]:

	userID	artistID	weight	artistNAME	voteRating
0	2	51	13883	Duran Duran	10.000000
1	4	51	228	Duran Duran	1.643836
2	27	51	85	Duran Duran	0.612833
3	28	51	10	Duran Duran	0.072098
4	62	51	528	Duran Duran	3.806777
...	...	...	...	...	...
92829	2100	18726	337	Nyktalgia	2.429704
92830	2100	18727	297	Atsakau niekadA	2.141312
92831	2100	18728	281	Domantas Razauskas	2.025955
92832	2100	18729	280	Atalyja	2.018745
92833	2100	18730	263	Les Chants de Nihil	1.896179

92834 rows × 5 columns

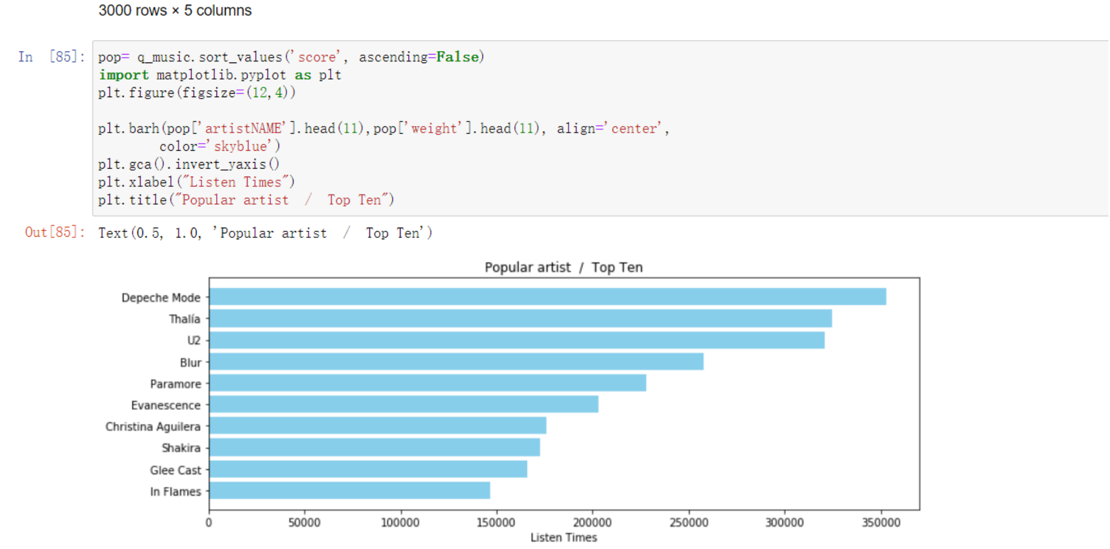
通过矩阵计算得分并通过得分排序：



```
In  [62]: q_music = music_data1.copy().loc[music_data1[’weight’] >= m]
          q_music.shape

Out[62]:  (9288, 5)

In  [63]: def weighted_rating(x, m=m, C=C):
              v = x[’weight’]
              R = x[’voteRating’]
              # Calculation based on the IMDB formula
              return (v/(v+m) * R) + (m/(m+v) * C)
          # Define a new feature ’score’ and calculate its value with `weighted_rating()`

          q_music[’score’] = q_music.apply(weighted_rating, axis=1)

In  [65]: #Sort movies based on score calculated above
          q_music = q_music.sort_values(’score’, ascending=False)
          # q_music.drop(’voteRating’,axis=1, inplace=True)
          # q_music.drop(’userID’,axis=1, inplace=True)
          # q_music.drop(’friendID’,axis=1, inplace=True)
          # q_music.drop_duplicates( ’artistID’,’first’,True)
          #Print the top 15 movies
          q_music.drop([’voteRating’],axis=1,inplace=True)
          q_music.head(3000)

Out[65]:
```

	userID	artistID	weight	artistNAME	score
2258	1642	72	352698	Depeche Mode	9.973051
35313	2071	792	324663	Thalía	9.970734
26772	1094	511	320725	U2	9.970376
7610	1905	203	257978	Blur	9.963210
26140	1664	498	227829	Paramore	9.958371
...	...	...	...	...	...
27005	512	517	3190	Korn	7.915196

应用 MATLAB 库通过得分统计排名显示排名前十艺术家名字和收听次数：



3000 rows × 5 columns

```
In  [85]: pop= q_music.sort_values(’score’, ascending=False)
          import matplotlib.pyplot as plt
          plt.figure(figsize=(12,4))

          plt.barh(pop[’artistNAME’].head(11),pop[’weight’].head(11), align=’center’,
                  color=’skyblue’)
          plt.gca().invert_yaxis()
          plt.xlabel(”Listen Times”)
          plt.title(”Popular artist  /  Top Ten”)

Out[85]: Text(0.5, 1.0, ’Popular artist  /  Top Ten’)
```

参考文献：

[1] Getting Started with a Movie Recommendation System

Ibtesam Ahmed

https://www.kaggle.com/ibtesama/getting-started-with-a-movie-recommendation-system