Lez19_MovieLens

January 6, 2023

[1]:
```
# Imports
import numpy as np
import pandas as pd
import matplotlib.pylab as plt
```

[2]:
```
#Get the dataset
!wget https://files.grouplens.org/datasets/movielens/ml-latest-small.zip
!unzip ml-latest-small.zip
```

```
--2023-01-06 15:24:21--  https://files.grouplens.org/datasets/movielens/ml-
latest-small.zip
Resolving files.grouplens.org (files.grouplens.org)… 128.101.65.152
Connecting to files.grouplens.org (files.grouplens.org)|128.101.65.152|:443…
connected.
HTTP request sent, awaiting response… 200 OK
Length: 978202 (955K) [application/zip]
Saving to: ‘ml-latest-small.zip’

ml-latest-small.zip 100%[===================>] 955.28K   437KB/s    in 2.2s

2023-01-06 15:24:24 (437 KB/s) - ‘ml-latest-small.zip’ saved [978202/978202]

Archive:  ml-latest-small.zip
   creating: ml-latest-small/
  inflating: ml-latest-small/links.csv
  inflating: ml-latest-small/tags.csv
  inflating: ml-latest-small/ratings.csv
  inflating: ml-latest-small/README.txt
  inflating: ml-latest-small/movies.csv
```

[3]:
```
#Load the data
movies = pd.read_csv('ml-latest-small/movies.csv')
ratings = pd.read_csv('ml-latest-small/ratings.csv')

#Investigate the data
movies.head()
```

[3]:
```
   movieId                               title  \
0        1                    Toy Story (1995)
1        2                      Jumanji (1995)
2        3             Grumpier Old Men (1995)
3        4            Waiting to Exhale (1995)
4        5  Father of the Bride Part II (1995)

                                        genres
0  Adventure|Animation|Children|Comedy|Fantasy
1                   Adventure|Children|Fantasy
2                               Comedy|Romance
3                         Comedy|Drama|Romance
4                                       Comedy
```

[4]:
```python
ratings.head()
```

[4]:
```
   userId  movieId  rating  timestamp
0       1        1     4.0  964982703
1       1        3     4.0  964981247
2       1        6     4.0  964982224
3       1       47     5.0  964983815
4       1       50     5.0  964982931
```

[5]:
```python
#Drop the timestamp, since it is not important for the recommender system
ratings = ratings.drop('timestamp', axis=1)
ratings.head()
```

[5]:
```
   userId  movieId  rating
0       1        1     4.0
1       1        3     4.0
2       1        6     4.0
3       1       47     5.0
4       1       50     5.0
```

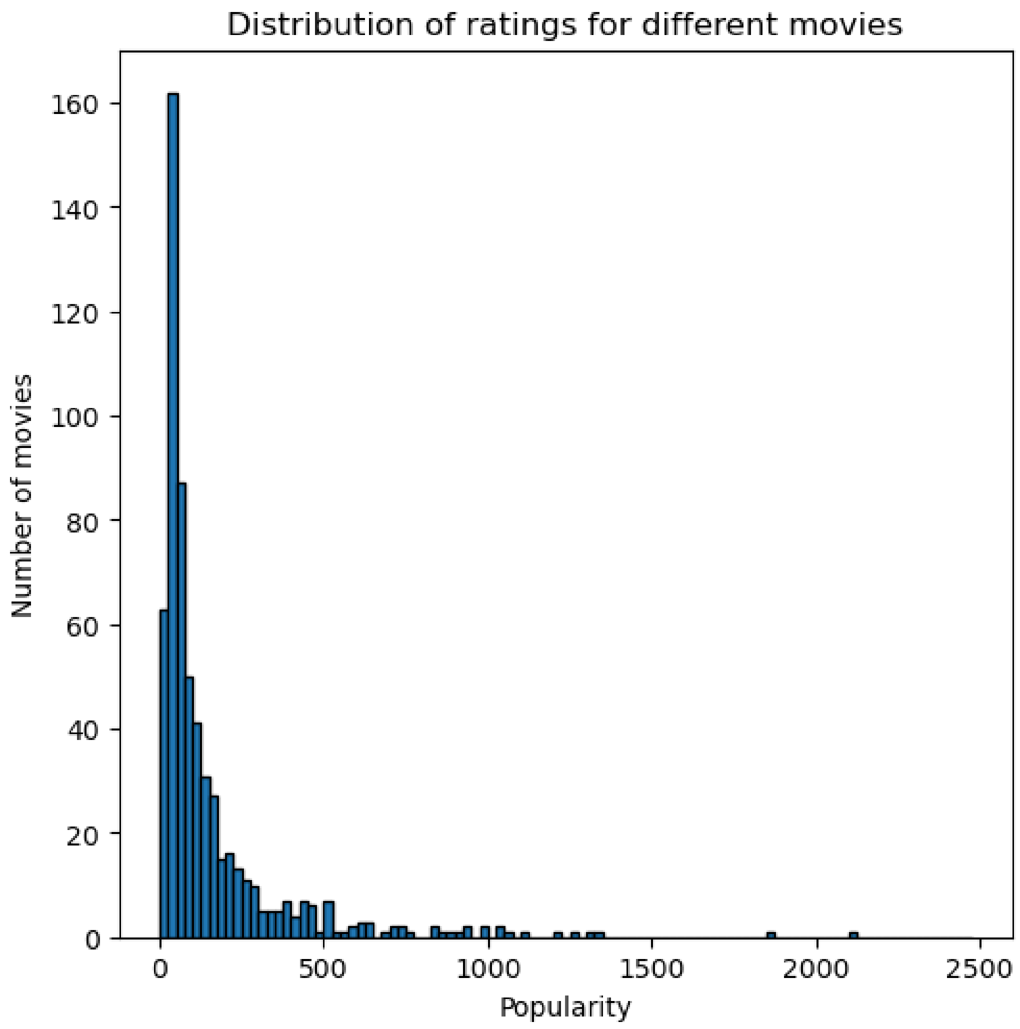
[6]:
```python
# Analyse sparsity of the dataset, given by sparsity = #ratings/(#users*#items)
number_ratings = len(ratings)
users = ratings['userId'].values
users_len = len(users)
items = movies['movieId'].values
items_len = len(items)
sparsity = number_ratings/(users_len*items_len)

print("Sparsity: ", sparsity)
```

```
Sparsity:  0.00010264832683227263
```

```
[7]: #Print sparsity matrix
     ratings.pivot('userId','movieId','rating').fillna(0)
```

```
[7]: movieId  1        2        3        4        5        6        7        8        \
     userId
     1            4.0      0.0      4.0      0.0      0.0      4.0      0.0      0.0
     2            0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0
     3            0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0
     4            0.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0
     5            4.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0
     …          …      …      …      …      …      …      …      …
     606          2.5      0.0      0.0      0.0      0.0      0.0      2.5      0.0
     607          4.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0
     608          2.5      2.0      2.0      0.0      0.0      0.0      0.0      0.0
     609          3.0      0.0      0.0      0.0      0.0      0.0      0.0      0.0
     610          5.0      0.0      0.0      0.0      0.0      5.0      0.0      0.0

     movieId  9        10       …  193565   193567   193571   193573   193579   193581   \
     userId                     …
     1            0.0      0.0  …     0.0      0.0      0.0      0.0      0.0      0.0
     2            0.0      0.0  …     0.0      0.0      0.0      0.0      0.0      0.0
     3            0.0      0.0  …     0.0      0.0      0.0      0.0      0.0      0.0
     4            0.0      0.0  …     0.0      0.0      0.0      0.0      0.0      0.0
     5            0.0      0.0  …     0.0      0.0      0.0      0.0      0.0      0.0
     …          …      …   …    …      …      …      …      …      …
     606          0.0      0.0  …     0.0      0.0      0.0      0.0      0.0      0.0
     607          0.0      0.0  …     0.0      0.0      0.0      0.0      0.0      0.0
     608          0.0      4.0  …     0.0      0.0      0.0      0.0      0.0      0.0
     609          0.0      4.0  …     0.0      0.0      0.0      0.0      0.0      0.0
     610          0.0      0.0  …     0.0      0.0      0.0      0.0      0.0      0.0

     movieId  193583   193585   193587   193609
     userId
     1            0.0      0.0      0.0      0.0
     2            0.0      0.0      0.0      0.0
     3            0.0      0.0      0.0      0.0
     4            0.0      0.0      0.0      0.0
     5            0.0      0.0      0.0      0.0
     …          …      …      …      …
     606          0.0      0.0      0.0      0.0
     607          0.0      0.0      0.0      0.0
     608          0.0      0.0      0.0      0.0
     609          0.0      0.0      0.0      0.0
     610          0.0      0.0      0.0      0.0

     [610 rows x 9724 columns]
```

```
[8]: # Plot the distribution of the popularity of the movies
     distribution_popularity_movies = ratings[['userId', 'movieId']].
      ↪groupby(['userId']).count()
     distribution_popularity_movies = distribution_popularity_movies.
      ↪rename(columns={"movieId": "#ratings"})

     plt.figure(figsize = (6, 6))
     plt.hist(distribution_popularity_movies['#ratings'], bins = range(0, 2500, 25),␣
      ↪edgecolor = 'black')
     plt.title('Distribution of ratings for different movies')
     plt.xlabel('Popularity')
     plt.ylabel('Number of movies')
     plt.show()
```



```
[ ]:
```