Recommendation Systems

PRESENTED BY SNEHA SARKAR

Contents

what it is. where it is used.

the one i've built.

code.

execution.

WHAT IS IT?

A recommender system or
a recommendation system is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item.

WHERE IS IT USED?

everywhere.

Types

Content-Based Filtering

Collaborative Filtering

The one I've built

- uses the correlation between the ratings assigned to different movies, in order to find the similarity between the movies.
- the MovieLens Dataset (contains 100000 ratings for 9000 movies by 700 users.)
- "movies.csv" and "ratings.csv" files.

''' The '''

(some of it)

Code

```
import numpy as np
import pandas as pd
ratings_data = pd.read_csv("ratings.csv")
movie_names = pd.read_csv('movies.csv')
movie_data = pd.merge(ratings_data, movie_names, on
= 'movieId')
```

```
movie_data.groupby('title')['rating'].mean().sort_values(ascending=False).head()
```

```
title
Burn Up! (1991)                                     5.0
Absolute Giganten (1999)                            5.0
Gentlemen of Fortune (Dzhentlmeny udachi) (1972)    5.0
Erik the Viking (1989)                              5.0
Reality (2014)                                      5.0
Name: rating, dtype: float64
```

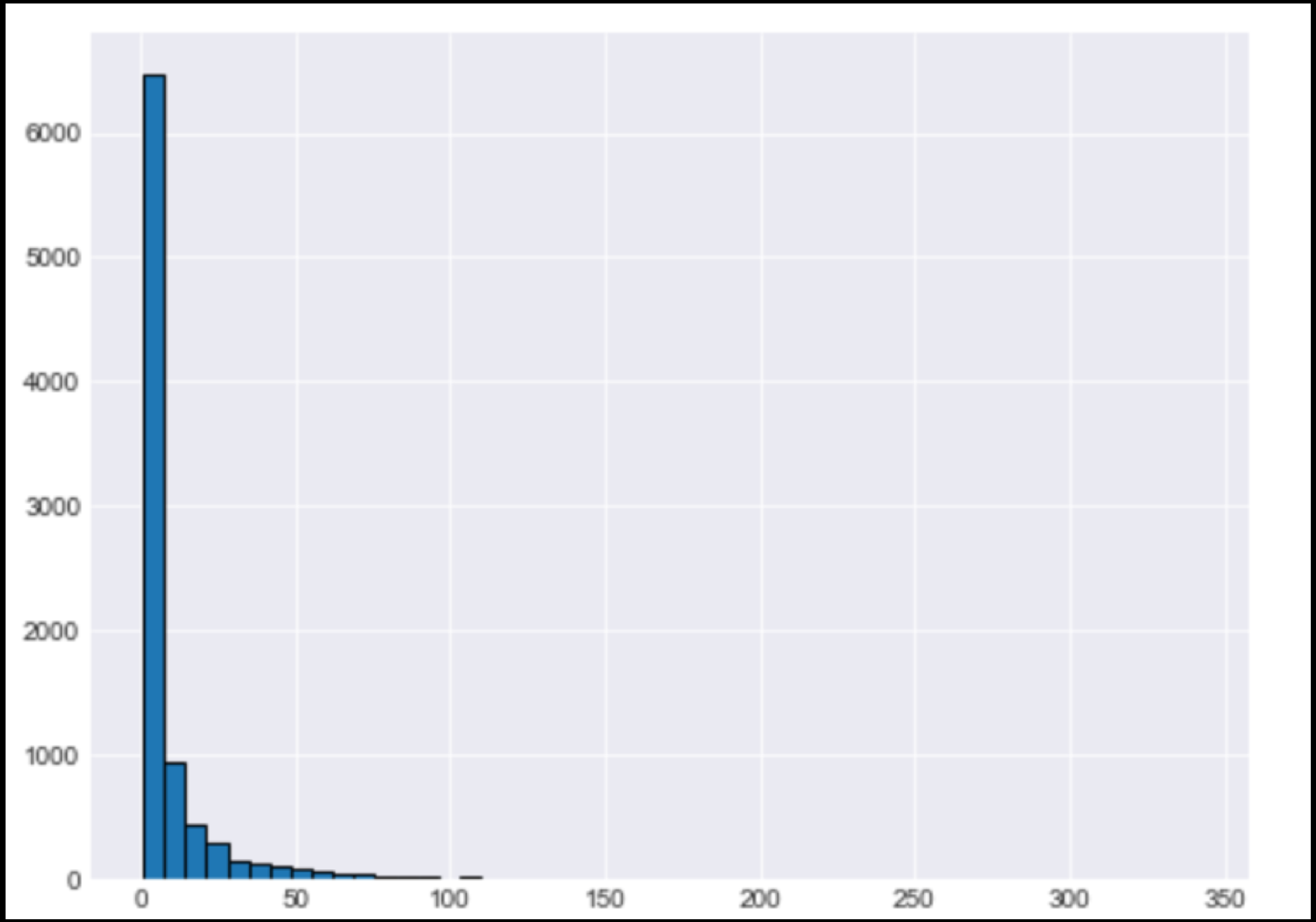
```
movie_data.groupby('title')
['rating'].count().sort_values(ascending=False).
head()
```

```
title
Forrest Gump (1994)                              341
Pulp Fiction (1994)                              324
Shawshank Redemption, The (1994)                 311
Silence of the Lambs, The (1991)                 304
Star Wars: Episode IV - A New Hope (1977)        291
Name: rating, dtype: int64
```

```
ratings_mean_count =
pd.DataFrame(movie_data.groupby('title')['rating'].mean())
ratings_mean_count['rating_counts'] =
pd.DataFrame(movie_data.groupby('title')['rating'].count())
```

title	rating	rating_counts
"Great Performances" Cats (1998)	1.750000	2
$9.99 (2008)	3.833333	3

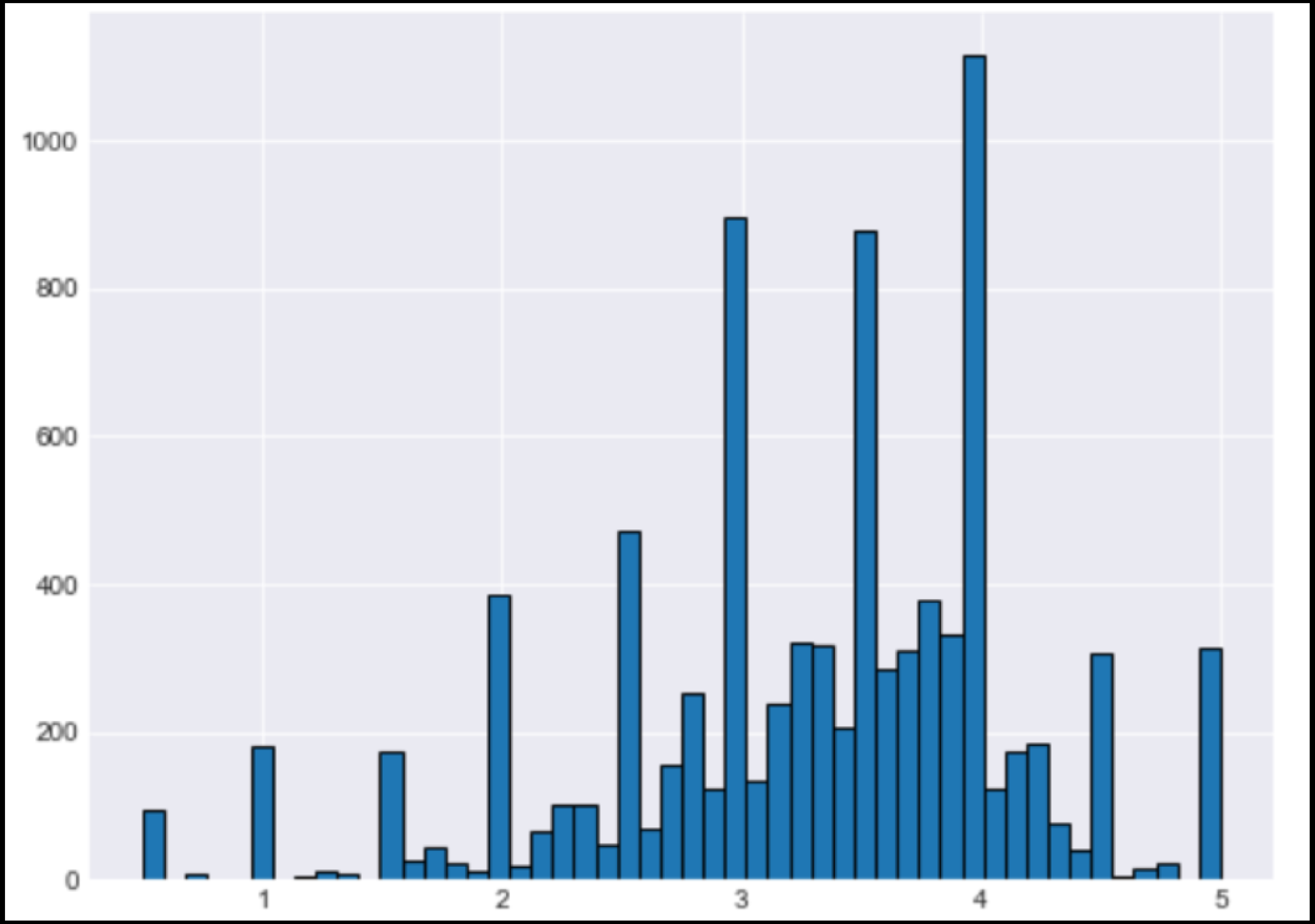
histogram for the number of ratings represented by the "rating_counts" column



movies

number of ratings

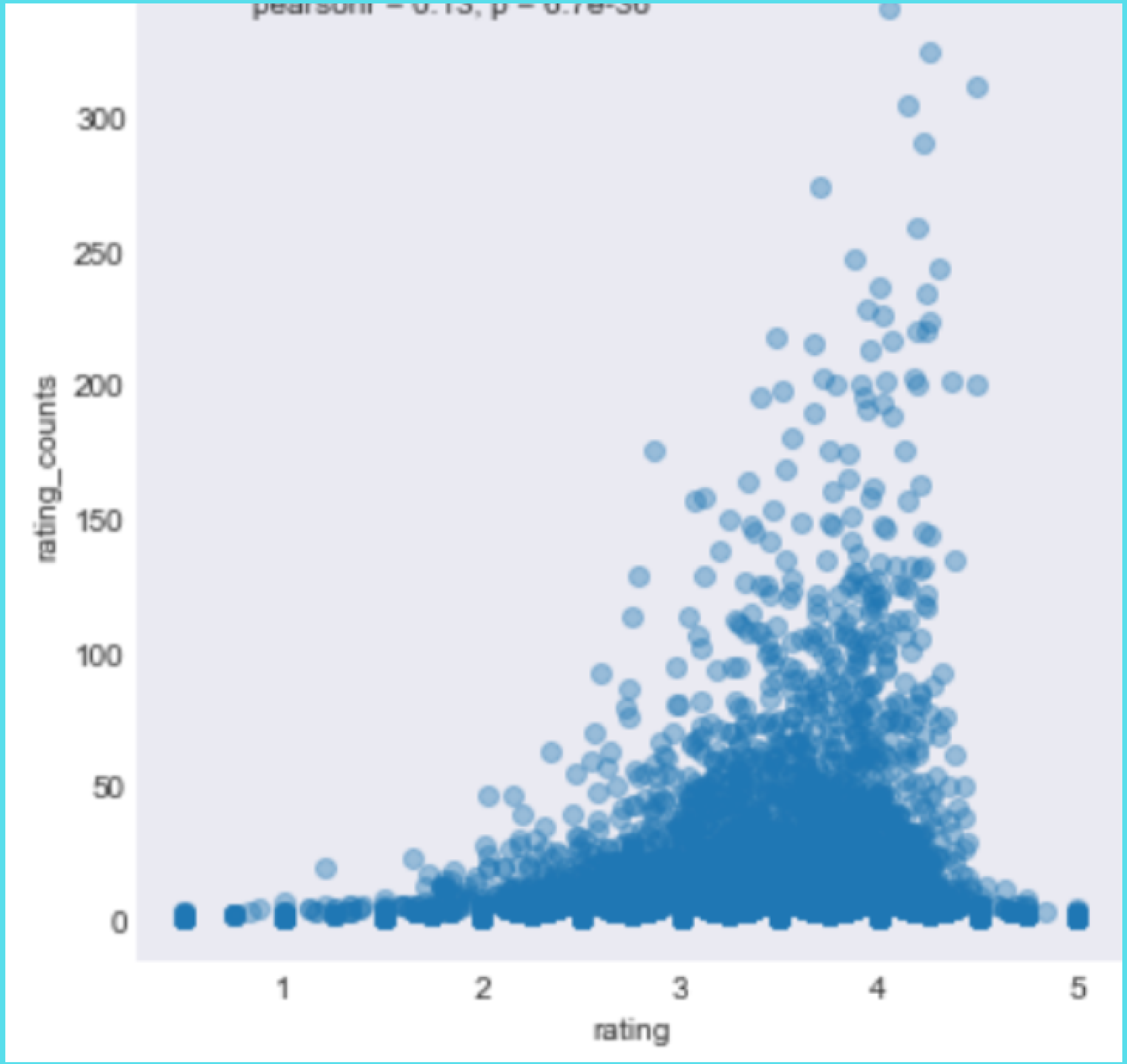
a histogram for average ratings



movies

average rating

Movies with a higher number of ratings usually have a high average rating as well since a good movie is normally well-known and a well-known movie is watched by a large number of people, and thus usually has a higher rating.



If we plot average ratings against the number of ratings

movies with higher average ratings actually have more number of ratings, compared with movies that have lower average ratings.

```
user_movie_rating =
movie_data.pivot_table(index='userId',
columns='title', values='rating')
```

creates a matrix where each column is a movie name and each row contains the rating assigned by a specific user to that movie.

title	"Great Performances" Cats (1998)	$9.99 (1998)	'Hellboy': The Seeds of Creation (2008)	'Neath the Arizona Skies (1934)	'Round Midnight (1986)	'Salem's Lot (2004)	'Til There Was You (1997)	'burbs, The (1989)	'night Mother (1986)	(500) Days of Summer (2009)	...	Zulu (1964)	Zulu (2013)
userId													
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN

```
forrest_gump_ratings = user_movie_rating['Forrest Gump (1994)']
```

```
userId
1     NaN
2     3.0
3     5.0
4     5.0
5     4.0
Name: Forrest Gump (1994), dtype: float64
```

```
movies_like_forrest_gump =
user_movie_rating.corrwith(forrest_gump_ratings)
corr_forrest_gump =
pd.DataFrame(movies_like_forrest_gump,columns=
['Correlation'])
corr_forrest_gump.dropna(inplace=True)
corr_forrest_gump =
corr_forrest_gump.join(ratings_mean_count['rating_counts'])
print(corr_forrest_gump.head())
print(corr_forrest_gump[corr_forrest_gump['rating_counts']>5
0].sort_values('Correlation', ascending=False).head())
```

title	Correlation	rating_counts
Forrest Gump (1994)	1.000000	329
Mr. Holland's Opus (1995)	0.652144	80
Pocahontas (1995)	0.550118	68
Grumpier Old Men (1995)	0.534682	52
Caddyshack (1980)	0.520328	52

The End.