GRAPH-SEARCH BASED RECOMMENDATION SYSTEMS: PROTOTYPE FOR A MOVIE STREAMING SERVICE

Goreva A., Grosheva A., Listarov G.,
BIDP 221

Basic information about the project

Project Communication Channel

Telegram Chat

Roles in the team

Listarov G. - Coordinator

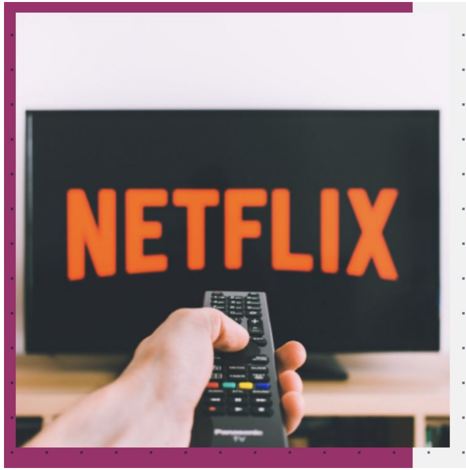
Goreva A. - Data Analyst

Grosheva A. - Resource Investigator

Key idea

Background:

Companies accumulate a huge continuous amount of data → information about user actions may help to effectively improve services and applications & increase company key financial metrics



Case

Netflix is moving the target - newest content engagement KPI counts the hours users spent instead of unique eyeballs

The advantages of using total streaming hours as a KPI:

- Time translates fairly easily to dollars.
- It rewards both binge-viewing and repeat watches, two long standing viewer behaviors that Netflix didn't invent but certainly exploits.
- It translates across media: whether you spend time watching movies, playing games or streaming a personalized algorithm of short videos, total time watched is a comparison metric.
- It's device-agnostic, whether you watch on a phone or a massive home entertainment system.

→ **Search for the most optimized and effective methods of using large amounts of user data to improve the accuracy of recommendation systems**

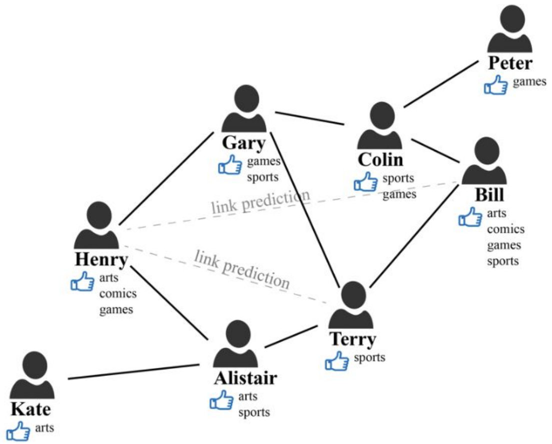
Goal and steps

The aim of this project:

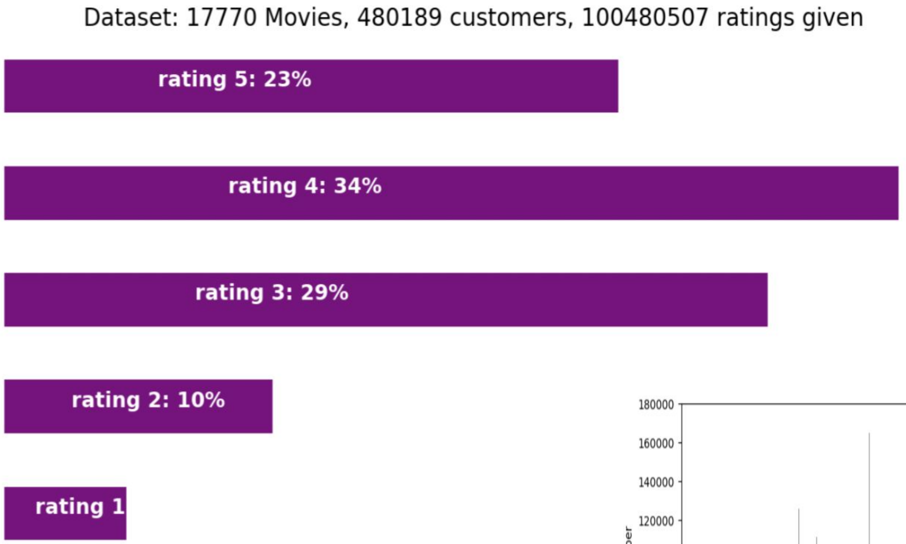
To propose a graph-based model for a recommendation system with Collaborative filtering approach that deals with the data of users of streaming service “Netflix” of TV shows and movies for modeling users’ preferences and setting up content recommendations.

Steps

- Identify the goal of the project, examine the existing theoretical literature and previous research
- Choose and prepare the dataset
- Design the network and give a description to it
- Prototype the recommendation system based on the dataset

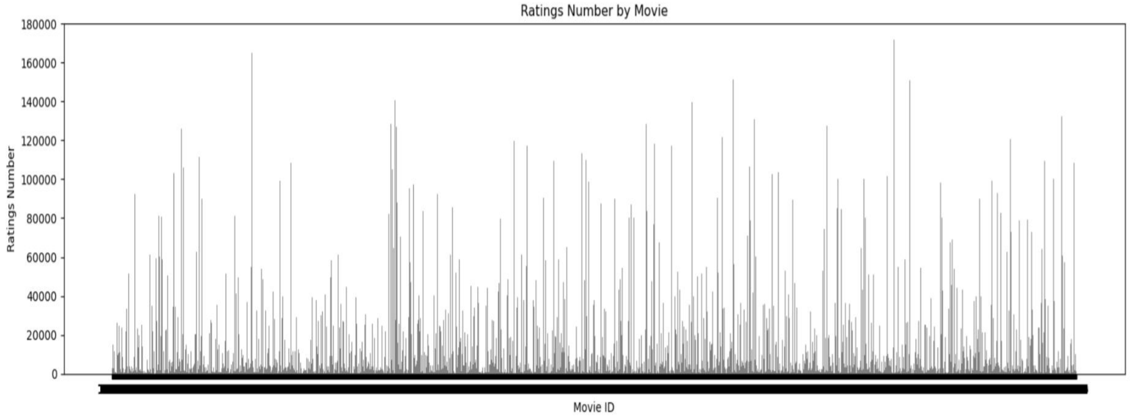


Dataset review



Insights:

- the rating tends to be relatively positive (>3)
- there seems to be a large spread in the number of reviews of films



Dataset filtering

Huge amount of data - what could be done?

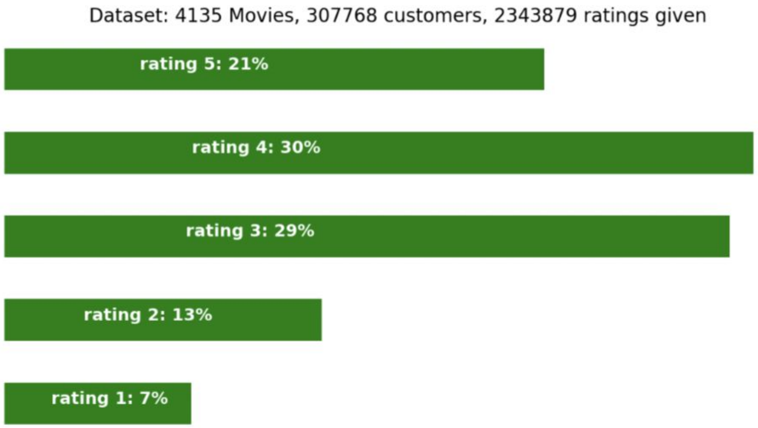
Filter out rarely rated movies and users who don't give enough ratings → allows to cut off the unneeded dataframe rows and optimise the future recommendation system

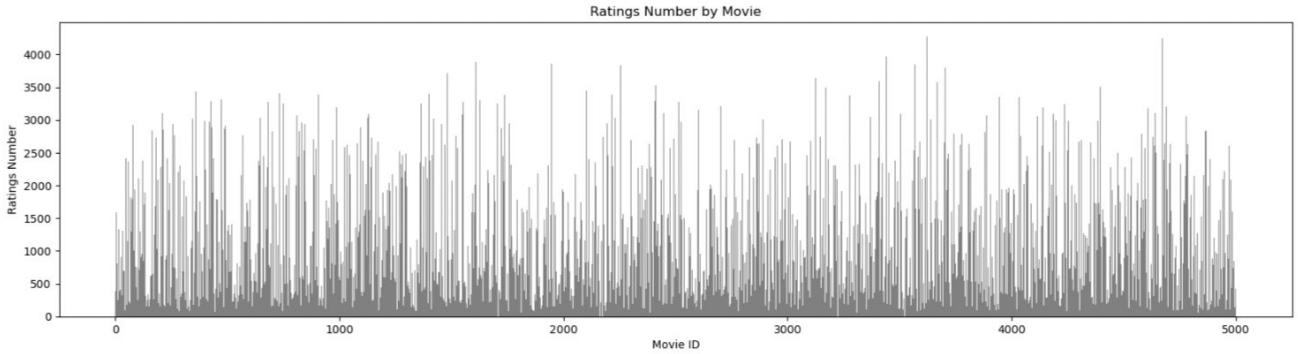
- **minimal movie ratings = 1000 & minimal user ratings = 200:**

100 M ➡ 75 M rows

- **movies amount = 4135**

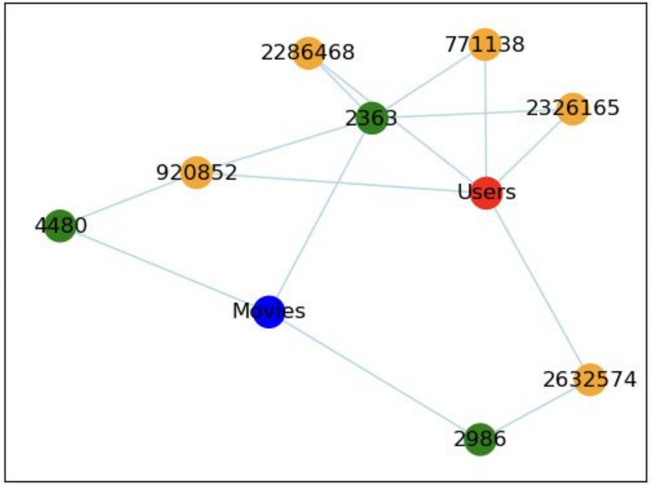
75 M ➡ 2,3 M rows





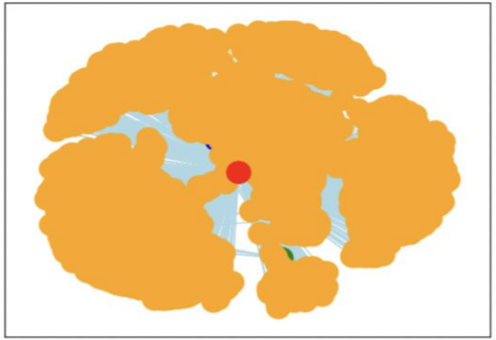
Network

3 Movies:
2363, 2986, 4480
5 Users:
771138, 2326165, 920852, 2286468, 2632574



Average degree centrality : 0.31
Average betweenness centrality : 0.11
Diameter : 3
Density : 0.31

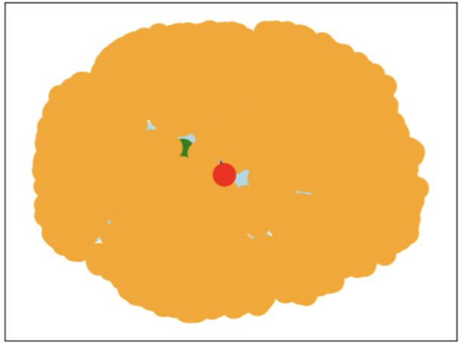
Average degree centrality : 0.0009342622508345556
Average betweenness centrality : 0.00023200861387749465
Diameter : 3
Density : 0.0009342622508344491



10 Movies
All Users

30 Movies
All Users

Average degree centrality : 0.0005475996746808504
Average betweenness centrality : 0.00013588109716537142
Diameter : 3
Density : 0.0005475996746809494



Singular value decomposition

Surprise is a Python scikit for building and analyzing recommender systems that deal with explicit rating data: various tools to run cross-validation procedures and search the best parameters for a prediction algorithm.

algo: SVD is a matrix factorization technique, which decomposes any matrix into 3 generic and familiar matrices
SVD is used as a collaborative filtering technique

Cross validation

Parameters

algo (**AlgoBase**) – The algorithm to evaluate.
data (**Dataset**) – The dataset on which to evaluate the algorithm.
measures (*list of string*) – The performance measures to compute. Allowed names are function names as defined in the **accuracy** module. Default is **['rmse', 'mae']**.

Returns

'test_*' where * corresponds to a lower-case accuracy measure, e.g. **'test_rmse'**: numpy array with accuracy values for each testset.
'train_*'
'fit_time': numpy array with the training time in seconds for each split.
'test_time': numpy array with the testing time in seconds for each split.

Result

```
{'test_rmse': array([1.0116773 , 1.01200926,
1.01307708, 1.01197559, 1.00952215]),
 'test_mae': array([0.79722085, 0.79820577,
0.7981748 , 0.79820789, 0.79589913]),

 'fit_time': (17.50406503677368,
  17.41688108444214,
  18.687373876571655,
  18.22533392906189,
  18.421394109725952),

 'test_time': (2.1945290565490723,
  2.1063990592956543,
  1.3900861740112305,
  2.1463277339935303,
   2.3499557971954346)}
```

Movies rated high by customer

	customer_id	rating	review_date	movie_id	year	name
0	785314	5	2002-03-18	57	1995	Richard III
1	785314	5	2005-08-09	395	1935	Captain Blood
2	785314	5	2004-11-09	907	1930	Animal Crackers
3	785314	5	2003-06-22	1552	1983	Black Adder
4	785314	5	2003-09-08	2713	1953	Glen or Glenda
5	785314	5	2002-02-03	2847	1920	The Mark of Zorro
6	785314	5	2003-09-08	3590	1963	Jason and the Argonauts
7	785314	5	2004-09-04	3949	1991	Terminator 2: Extreme Edition: Bonus Material
8	785314	5	2002-05-10	3984	1959	On the Beach
9	785314	5	2004-10-14	4253	1949	Kind Hearts and Coronets

Recommendation

Predict with SVD

Movies recommended for customer

	index	year	name	Estimate_Score
559	559	2003	Star Trek: Enterprise: Season 3	5.000000
3887	3887	1994	NYPD Blue: Season 2	4.970367
31	31	2004	ABC Primetime: Mel Gibson's The Passion of th...	4.963453
3004	3004	1992	As Time Goes By: Series 1 and 2	4.892540
1914	1914	2000	Law & Order: Special Victims Unit: The Second...	4.813029
3072	3072	1997	Ballykissangel: Series 2	4.793232
662	662	1999	La Femme Nikita: Season 3	4.770331
777	777	2003	A Touch of Frost: Seasons 7 & 8	4.748415
4237	4237	2000	Inu-Yasha	4.739102
1688	1688	2003	Concert for George	4.708004