MSML651 - Movie Rating Predictor

Akshaya Anand

Motivation

- Spend a lot of time trying to figure out what to watch on Netflix
- Interested in learning about recommendation system
- Large Netflix user-movie dataset available on Kaggle

Dataset - Netflix Challenge Dataset

- 100 million ratings from 480,000 users

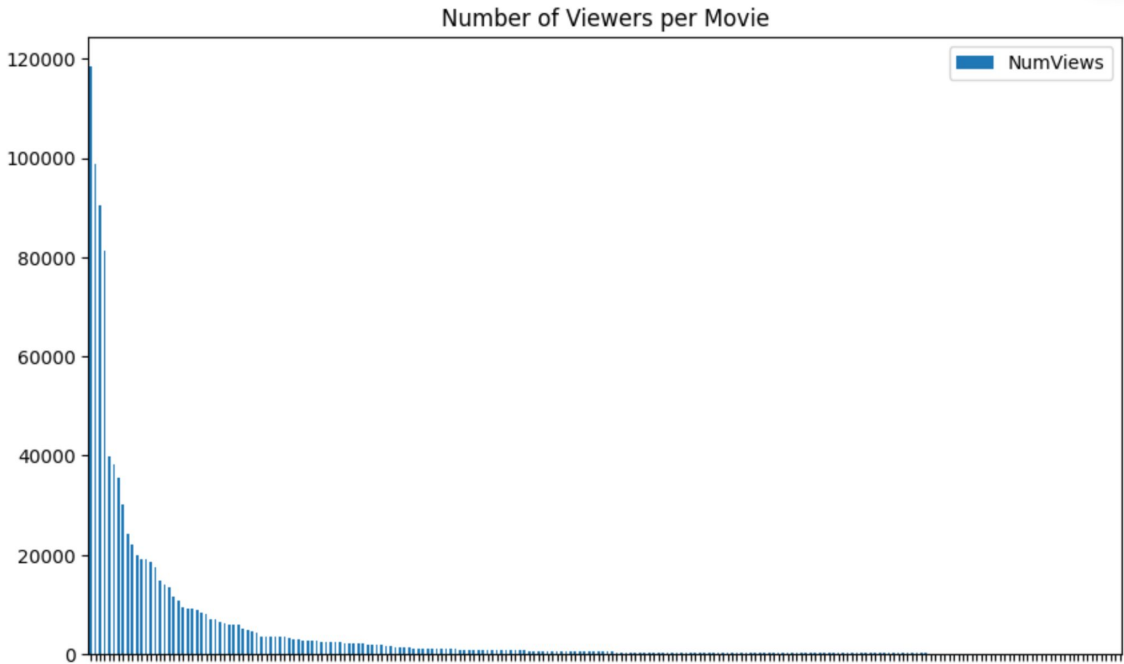
- 17000 movie titles.

- between 1998-2005

Least popular movie: 62 views

Most popular movie: 118,413 views

Viewers per Movie



Method

1) Extract first 1M rows (easy, no distribution needed)
   - 225 Movies
   - 283,670 users

2) Reformat data (**chunking**)

3) Split into train/val/test (easy, no distribution needed)

   Train (80%): 180, val: 22 (10%), test: 23 (10%)

4) Compute movie rating vector for each user (**parallelize**)

5) Compute adjusted cosine similarity between users (**parallelize)**

Method - Reformat data

- While reading, wrote intermediate file named by each movie_id
  - Rows were user_id, rating, date
- 225 chunks
- Dask dataframe for computing statistics

Method - User Movie Ratings Vector

For each user:

- All movies they watched were columns
- Values were rating - mean(rating)

Ran 3 jupyter notebooks, each assigned 75 users to process

Ratings written to intermediate file storage

Methods - Cosine Similarity

For each user in val set, compute cosine similarity against all other users in train

- Find intersection of movies shared between pair
  - val_user_shared.dot(train_user_shared)/Mag(val_user)*Mag(train_user)
  - (user,user) filtering bc # users is much more than # movies
- 3 notebooks processing 75 users each, sharing read access to 1 train file

Sort the cosine similarities

- Weighted average of k similar users to predict rating for each movie
  - Weighted average = cosine similarity score * similar users movie rating
    - If similar user did not watch that movie, then use the users average rating
  - Tried k values of 5,10,15,20. Best k value used for test evaluation

Results

- Still working on this
- How to best present analysis

Predictions for user rating of movie 142

Name	497196	192061	76196	2625420	724592	566733	502355	1604278	
Actual	4	3	3	2	3	5	3	4	
top: 5	3	4	3	1	4	4	3	4	22/27
top: 10	3	4	2	1	3	4	3	4	22/27
top: 15	3	4	3	1	3	4	3	4	23/27
top: 20	3	4	3	1	3	4	3	4	23/27

Next Steps

- Fix bugs
  - movie _id swapped with user_id
- Report results
  - Distribution of cosine similarities
- Incorporate Dask (currently running multiple jupyter notebooks)
  - Can try 12M rows?
- Incorporate movie titles

Fun what if: Incorporate social media friend network into user similarity calculation

- Close friends are more likely to have similar movie preferences
- This may be better than geography based or genre/movie cluster based

References

1. Dataset: https://www.kaggle.com/datasets/netflix-inc/netflix-prize-data
2. Starter Notebook: https://www.kaggle.com/code/laowingkin/netflix-movie-recommendation
3. Recommendation systems: Recommendation Systems and Netflix Challenge — A comprehensive introduction (with code) to collaborative filtering. Medium
4. Dask: https://examples.dask.org/dataframe.html
5. Pyspark: https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.RDD.html