Clustering-based movie recommender system

Jakub Jagła, Łukasz Borak
Maksymilian Żmuda-Trzebiatowski,
Krzysztof Bryszak

Poznan University of Technology
Marii Skłodowskiej-Curie 5, 60-965 Poznan, Poland

Abstract.

This paper presents an algorithm for a recommender system based on the MovieLens dataset. It's designed to create user ratings for yet unseen movies. It combines K-means clustering with genre-based user profiling. Results show significant improvement over a baseline score of 0.0561. Optimal cluster number (15) achieves MSE as low as 0.0381. This shows our proposed solution successfully managed to improve the recommendation accuracy.

Keywords: *recommender systems, data mining, association rules*

1. Introduction

Recommender systems are algorithms that provide presonalized suggestions for items most relevant for a given user based on the user's previous likes, dislikes and interaction. They've become a crucial concept in some industries. Nowadays, every major online vendor is using a recommender system, whether it's Netflix suggesting the next movie or Ceneo showing a shiny new phone.

Machine learning models predict the rating of a user on a given item. Then, they recommend items having the highest predicted rating. The idea behind it is to find patterns in similar consumer behavior towards a product. Clustering is one of the options when choosing the approach for this problem.

2. Related Work

While implementing our solution we looked at different implementations for this dataset. Here's two of them that we found very insightful and inspiring for our work.

2.1. Recommender System on MovieLens dataset by rposhala. [1]

The study also utilized the MovieLens dataset, although a smaller version was selected. An extensive data analysis was conducted that allowed us to gain some insights about the dataset. The approach was similar, involving the use of movie genres to predict the rating a user would give to a particular movie.

2.2. Movielens Dataset Recommender System by GdMacmillan [2]

The study also used the MovieLens dataset, but a smaller version was selected. A different metric, root mean squared error, was used to evaluate the solution. Additionally, a different approach was taken using ALS from PySpark.

3. Dataset

In our research, we used the MovieLens dataset from GroupLens. GroupLens is a research lab in the Department of Computer Science and Engineering at the University of Minnesota, Twin Cities, specializing in recommender systems, online communities, mobile and ubiquitous technologies, digital libraries, and local geographic information systems.

GroupLens Research has collected and made available rating data sets from the MovieLens web site[3]. The data sets were collected over various periods of time, depending on the size of the set.

4. Algorithm

When creating a recommender system, one must decide whether to use a content-based approach or a collaborative filtering approach. We have decided to use collaborative filtering in our system by calculating the mean of all the users in a given cluster to predict the rating.

The main work in our algorithm is done by the MovieRatingEstimator class. The process involves the following steps:

1. Data Preprocessing.
    - One-hot encoded movie genres.
    - Discretized release years.

- Calculated average ratings for each genre per user.

2. Clustering.

   - K-means clustering applied to group users with similar tastes.

3. Prediction.

   - Rating predicted by taking an average of all the users in a given cluster.

5. Results

We verified our model's performance with Mean-Squared Error (MSE) metric. The most basic approach that we took as a baseline was always predicting the global mean of the dataset - 3.5. This approach accounted to MSE of approximately 0.0561. For our model, we tested cluster configurations from 1 to 52 with a step of 2.

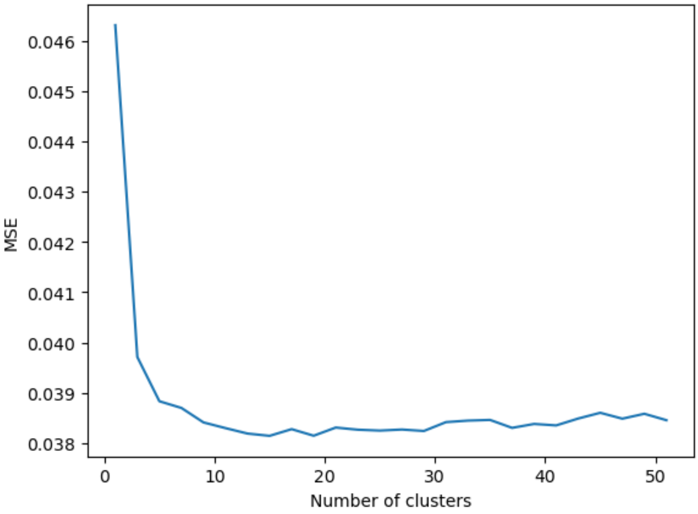

Figure 1. Mean Squared Error for Different Numbers of Clusters

The best number of clusters turned out to be K=15, which was good for an MSE score of approximately 0.03814.

K	**MSE**
K=1	MSE=0.04630768129328536
K=3	MSE=0.03971269853935524б
K=5	MSE=0.0388340652239941
K=7	MSE=0.038700796518337514
K=9	MSE=0.03841333267973151
K=11	MSE=0.03829988628889153
K=13	MSE=0.038191659270263655
K=15	**MSE=0.03814637435386054**
K=17	MSE=0.03827938976559852
K=19	MSE=0.03814875161963327
K=21	MSE=0.03831102060612177
K=23	MSE=0.038268322074502585
K=25	MSE=0.038250959849629396
K=27	MSE=0.03827158702454151
K=29	MSE=0.03824316497102219
K=31	MSE=0.03841893650511165
K=33	MSE=0.03844938938744321
K=35	MSE=0.03846294229771472
K=37	MSE=0.0383049377994991
K=39	MSE=0.03838275555217986
K=41	MSE=0.03835365823801215
K=43	MSE=0.038490701340281666
K=45	MSE=0.03860453875638069
K=47	MSE=0.03848773886944572
K=49	MSE=0.03858488961950332
K=51	MSE=0.0384573124601915

Table 1. MSE scores for different number of clusters

6. Conclusions

This study presents an effective recommender system that combines genre-based user profiling and K-means clustering to predict the movie ratings. Our research proved fruitful as we managed to get a very sizeable improvement compared to the baseline approach. The best performance can be seen at 15 clusters.

References

[1] rposhala. Recommender System on MovieLens dataset. *GitHub Repository*, 2020. URL `https://github.com/rposhala/Recommender-System-on-MovieLens-dataset`.

[2] GdMacmilan. MovieLens Dataset Recommender System. *GitHub Repository*, 2017. URL `https://github.com/GdMacmillan/spark_recommender_systems`.

[3] GroupLens. MovieLens Dataset. *Website*, 2018. URL `https://movielens.org/`.