MOVIE RECOMMENDER SYSTEMS
K-MEANS COLLABORATIVE FILTERING

Jakub Kubiak, Julia Lorenz, Mateusz Nowicki, Krzysztof Skrobała, Wojciech Bogacz

POZNAN UNIVERSITY OF TECHNOLOGY

ABSTRACT

In the era where enorumous volume of available movies can overwhelm users, movie recommendation systems are crucial to effectively navigate them to the areas of their preferences. This paper introduces a movie recommendation system utilizing K-means clustering to predict the rating a user would assign to the movie from the dataset. Multiple methods and evaluation criteria are used to assess the effectiveness of the approach.

INTRODUCTION

Our recommendation system leverages the K-means clustering algorithm to predict the rating a user would assign to a specific movie based on the preferences of similar users.

For the training and testing process we used MovieLens Full dataset with approximately 33 000 000 ratings assigned by over 300 000 users.

FIRST METHOD

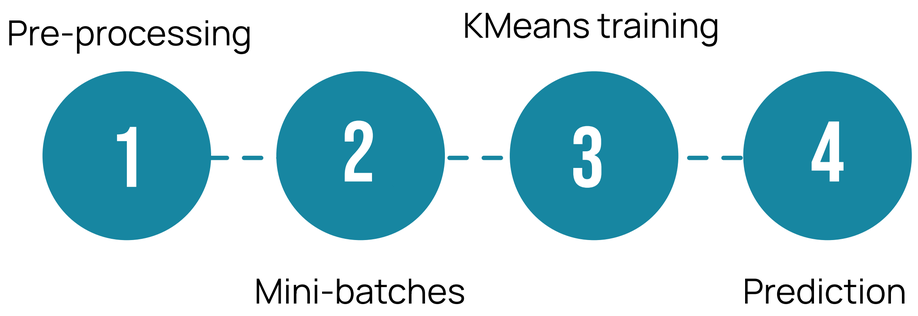

Figure 1. Basic classification process scheme for the first method

APPROACH

- Our initial approach involved several preprocessing steps to ensure meaningful and efficient predictions. Initially, we filtered out movies with fewer than five reviews to eliminate statistically insignificant data. We then focused exclusively on positive reviews, selecting those with scores greater than four.
- For data organization, we employed a **mini-batch strategy**, dividing the dataset into 49 batches. This data was represented in a binary matrix where rows corresponded to users and columns to movies, indicating whether a user had positively reviewed considered movie.
- We utilized **k-means clustering** to assign users clusters based on their review patterns. Due to the complexity of this operation, we performed an additional looping process instead of simple batching to manage memory constraints more effectively.

RESULTS

The results have not been satisfactory. The clustering approach gave worse MSE than simply making the prediction equal to the average review value of considered movie.

The reason, most probably, lays in the usage of K-means. Since the data have been encoded into binary table of reviews, the Euclidean distance used by the algorithm reduces to counting the number of variables on which two cases disagree.

This then leads to plenty of situations where the ties occurs, which is resolved arbitrarily.

SECOND METHOD

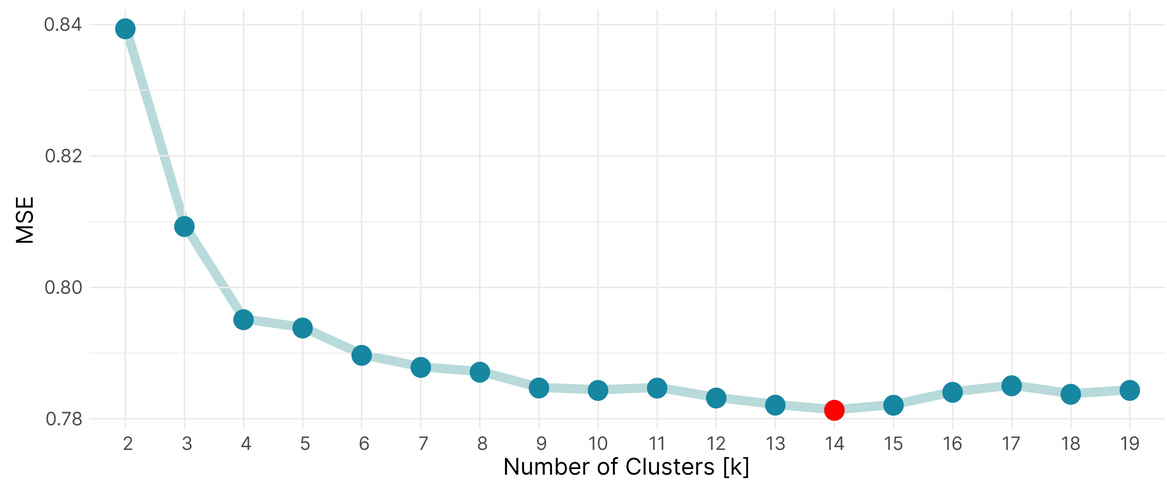

Figure 2. K-Means MSE rating for various cluster values

IMPROVED STRATEGY

In the improved approach we did not encode the data into binary table. By grouping the reviews on the movie type, the processed data became much smaller, therefore we could avoid using mini-batch strategy.

At this point rows correspond to specific users and columns to mean rating values assigned to movie category by the user. After all, we have set number of cluster to 14, as this value produced the best experimental results.

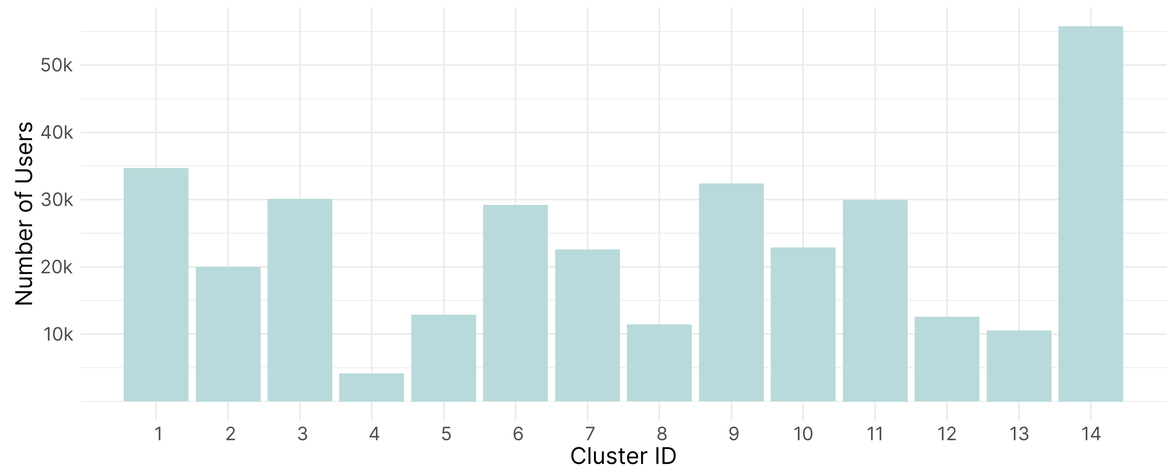

Figure 3. User distribution by clusters

FINAL RESULTS

We have found 14 basic preference groups. The discrepancies between the cluster sizes can be due to widespread popularities of certain movie genres, which naturally match people taste.

	Mean Squared Error
Average Movie Review	0.933
First Method	1.744
Second Method	0.781

Basing our predictions on simply average movie review yield substantially better results, when compared to the first method utilizing binarized data. Following further improvements, the second method, which leverages movie type grouping outperformed the aforementioned approaches.