Final Solution Report

Karina Denisova, BS21-DS-01

December 3, 2023

1 Introduction

In the age of information overload, recommendation systems have become integral in guiding users through the vast landscape of digital content. This paper aims to explore the effectiveness of recommendation systems using the MovieLens dataset, a widely recognized source of movie ratings and user preferences. By delving into the intricacies of user behavior and preferences, this study seeks to shed light on the potential enhancements that can be made to recommendation systems, ultimately aiming to improve user satisfaction and engagement.

2 Data analysis

The main dataset we use is the MovieLens 100K dataset consisting user ratings to movies. During EDA we found out that the most popular genre is drama

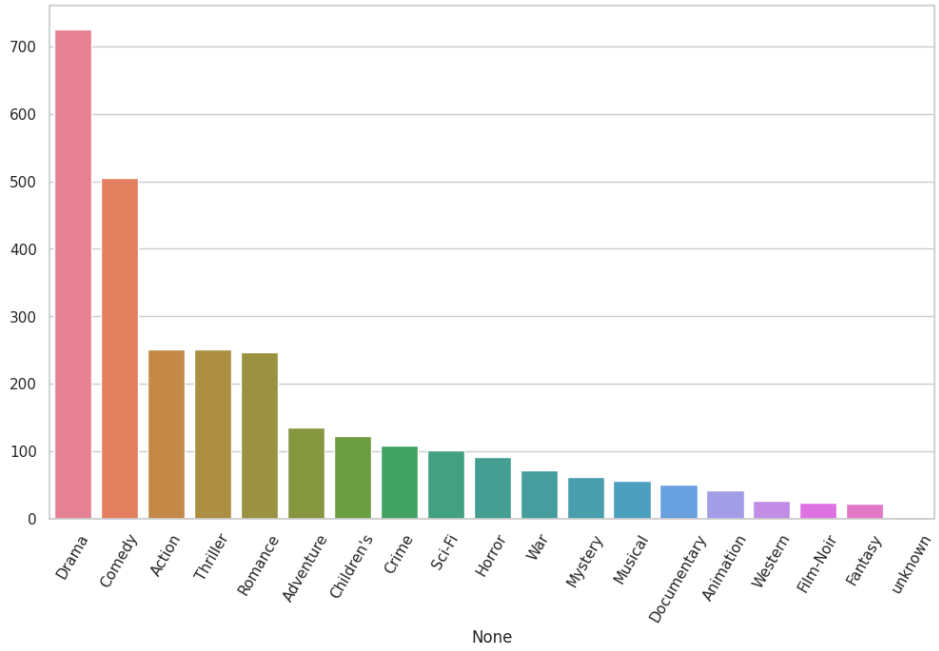

Figure 1: EDA

As it seen from figure 1, the dataset performs extremely extremely large gap between genres. These significant differences may affect the results of the recommendation.

3 Model Implementation

In this study, we will leverage dimensionality reduction methods to enhance the robustness and accuracy of Memory-Based Collaborative Filtering (CF) for recommendation systems. Our approach

involves compressing the user-item matrix into a lower-dimensional representation, thereby optimizing the computational efficiency and improving the overall performance of the recommendation system. Techniques such as Singular Value Decomposition (SVD), a low-rank factorization method, and Principal Component Analysis (PCA) will be employed for dimensional reduction, enabling us to effectively address the inherent sparsity in the data.

Furthermore, we will explore model-based methods, which are grounded in matrix factorization and are known for their proficiency in handling sparsity. To implement these techniques, we will utilize the "Surprise" library, which offers a comprehensive suite of algorithms including SVD, KNN, and Non-negative Matrix Factorization (NMF) for model-based recommendation systems. The "Surprise" library serves as a valuable resource, providing the necessary tools and implementations to facilitate our research endeavors. More information on the "Surprise" library and its functionalities can be found in the documentation available at [1].

Also we compare results of this model with simple Neural Network

3.1 K-Nearest Neighbours

K-Nearest Neighbors (KNN) is a popular algorithm in recommendation systems, leveraging the concept of similarity to make predictions. By identifying the nearest neighbors to a target user or item based on their characteristics or behavior, KNN offers a simple yet effective approach to generating personalized recommendations. This method relies on the assumption that similar users or items tend to have similar preferences, making it a valuable tool for enhancing the accuracy and relevance of recommendations.

So in my implementation i trying to use basic KNN from surprize library and then try to tune parameter using GridSearchCV to get better results

3.2 Singular Value Decomposition

Singular Value Decomposition (SVD) is a powerful mathematical tool commonly used in recommendation systems to reduce the dimensionality of user-item interaction matrices. By decomposing the original matrix into lower-rank approximations, SVD enables the system to capture latent features and patterns, facilitating more accurate and efficient recommendations. This method plays a pivotal role in uncovering the underlying structure of user preferences and item characteristics, ultimately enhancing the system's ability to make personalized and relevant suggestions.

As for KNN SVD was also tuned using GridSearchCV

3.3 Non-Negative Matrix Factorization

Non-Negative Matrix Factorization (NMF) is a valuable technique in recommendation systems, particularly for its ability to uncover interpretable and non-negative latent features from user-item interaction matrices. By constraining the factors to be non-negative, NMF offers a more intuitive representation of user preferences and item characteristics, contributing to the system's capacity to generate meaningful and tailored recommendations. This method is instrumental in capturing the inherent non-negativity of user-item interactions, thereby enhancing the system's ability to discern relevant patterns and deliver more impactful suggestions.

In my implementation was tuned with GridSearchCV and compared with original one

3.4 Matrix Factorization using Deep Learning

All previous models was compared with simple NN build with Keres library. An embedding is a mapping from discrete objects, such as words or movies in our case, to a vector of continuous values. These are used to find similarities between discrete objects.

The concept behind matrix factorization models is that the preferences of a user can be determined by a small number of hidden factors. And these are called as Embeddings.

And finally we take dot product which gives us the user's rating for the movie,

Estimated Rating = [(How much he likes action movies?) x (To what scale is it an action movie)] + [(Whether he likes old movies?) x (Whether the movie is recently released)]

and finally architecture of my model looked like following :

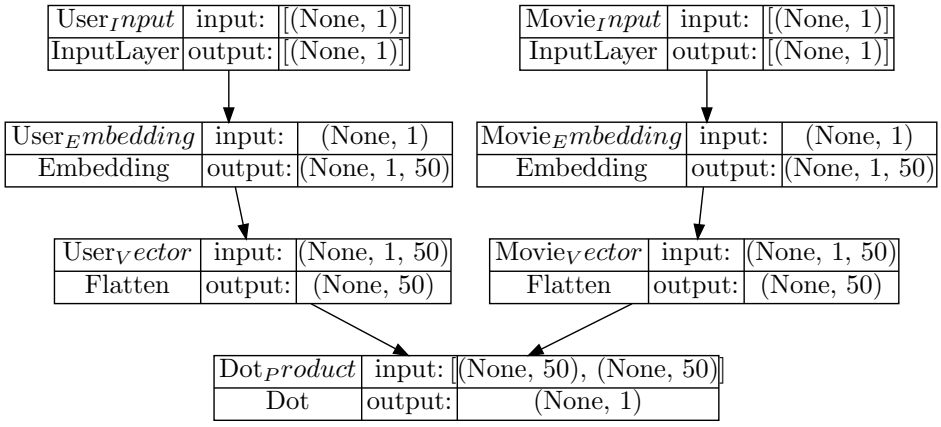

Figure 2: Model Srchitecture

4 Model Advantages and Disadvantages

4.1 Advantages

4.1.1 Memory-Based Collaborative Filtering (CF) with Dimensionality Reduction:

- Enhanced robustness and accuracy through dimensionality reduction, optimizing computational efficiency and improving overall performance.
- Effective handling of sparsity in the data, leading to more meaningful recommendations.

4.1.2 K-Nearest Neighbours (KNN):

- Simple and effective approach for generating personalized recommendations based on similarity.
- Ability to enhance the accuracy and relevance of recommendations by leveraging the assumption of similar user/item preferences.

4.1.3 Singular Value Decomposition (SVD) and Principal Component Analysis (PCA):

- Powerful tools for reducing dimensionality and capturing latent features, leading to more accurate and efficient recommendations.
- Uncovering underlying structures of user preferences and item characteristics, enabling personalized and relevant suggestions.

4.1.4 Non-Negative Matrix Factorization (NMF):

- Uncovering interpretable and non-negative latent features, contributing to the generation of meaningful and tailored recommendations.
- Enhanced ability to discern relevant patterns and deliver impactful suggestions by capturing the inherent non-negativity of user-item interactions.

4.1.5 Matrix Factorization using Deep Learning:

- Utilization of embeddings for finding similarities between discrete objects, leading to more personalized recommendations.
- Ability to determine user preferences based on a small number of hidden factors, providing a comprehensive approach to estimating user ratings for items.

4.2 Disadvantages

4.2.1 Memory-Based Collaborative Filtering (CF) with Dimensionality Reduction:

- Potential challenges in handling large-scale datasets and real-time recommendation scenarios.

4.2.2 K-Nearest Neighbours (KNN):

- Sensitivity to the choice of similarity metric and the number of nearest neighbors, which may impact the quality of recommendations.

4.2.3 Singular Value Decomposition (SVD) and Principal Component Analysis (PCA):

- Computational complexity and resource requirements for large-scale matrices and high-dimensional data.

4.2.4 Non-Negative Matrix Factorization (NMF):

- Sensitivity to the choice of hyperparameters and potential overfitting in complex recommendation scenarios.

4.2.5 Matrix Factorization using Deep Learning:

- Complexity in training and tuning deep learning models, requiring substantial computational resources and expertise.

5 Training Process

5.1 KNN

I use KNN model from surprize library and test it with following hyperparameters:

- number of neighbours: 5, 10, 20, 30
- epochs: 5, 10, 20

5.2 SVD

I use SVD model from surprize library and test it with following hyperparameters:

- number of factors: 50, 70
- epochs: 5, 10, 20
- learning rate: 0.5, 0.05
- regression: 0.06, 0.04

5.3 NMF

I use NMF model from surprize library and test it with following hyperparameters:

- number of factors : 15, 50, 75
- epochs: 5, 10, 20

5.4 NN

I build simple NN model and test it with following hyperparameters:

- batch size : 128
- epochs: 20
- optimizer: Adam(learning_rate = 0.0005)
- loss: MSE

6 Evaluation

7 Evaluation

We utilize RMSE as the primary evaluation metric for the following reasons:

7.1 Advantages of RMSE for Evaluating Recommendation Systems:

- Reflects Prediction Accuracy
- Sensitive to Prediction Errors
- Commonly Used and Interpretable
- Alignment with User Experience
- Considers All User-Item Pairs

RMSE offers a robust and interpretable means of assessing the accuracy and effectiveness of the recommendation systems under investigation, making it a suitable choice for evaluation.

7.2 Resuting metrics

Comparison of all the results of Model based CF are in the table:

Models	RMSE	RMSE after tuning	Best Parameters
KNN	0.9876	0.9771	n_neighbours: 5, n_epochs: 5
SVD	0.9456	0.9279	n_epochs: 5, n_factors: 75, lr_all: 0.05, reg_all: 0.06
NMF	0.9738	0.9774	n_epochs: 20, 'n_factors: 15

For the NN model, I got RMSE: 0.9192
For better visualization of the results, I use the following plot3:

8 Results

As we can see it from the table the lowest RMSE give as the simple NN model. But unfortunately 0.91 it is steel to much. That means that in future I need to improve my solution maybe by adding mode layers, but not epochs cause as it in 3 losses was the same from 5th to 20th epoch

References

[1] https://surprise.readthedocs.io/en/stable/.

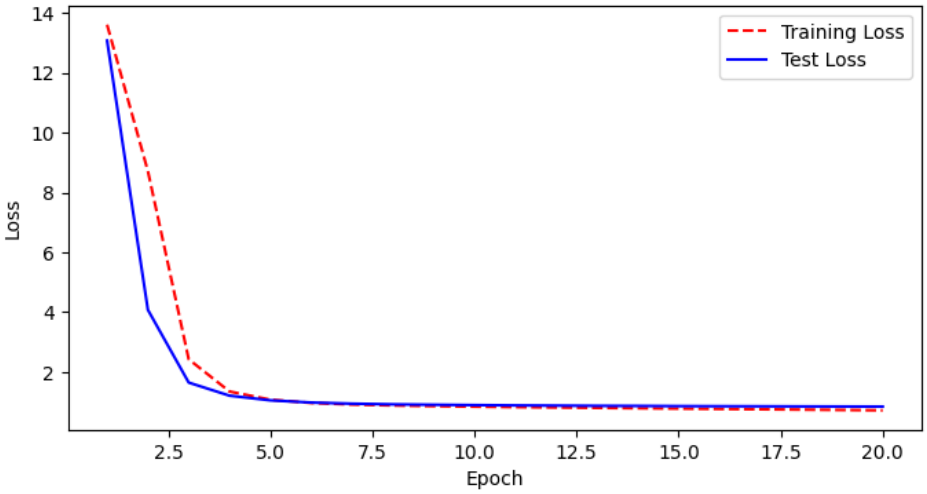

Figure 3: Results