Introduction

The goal of this assignment is to create a recommender system for movies that suggests movies to users based on their demographic information and favorite movies. Recommender systems have become an essential part of various domains, such as e-commerce, entertainment, social media, and online content platforms. In this project, I am using a machine learning model to suggest movies to users. While solving the task at hand, I created a benchmark evaluating the created model on metrics such as: RMSE, MSE, MAE and FCP. This report will describe the data exploration, solution implementation, training process, and evaluation on the benchmark. The benchmark scores of the system will also be explicitly stated.

Data analysis

We were asked to use the MovieLens 100K dataset to solve this task. It contains 100000 ratings by 943 users of 1682 movies. It also contains information about movies title, genre, release date and about users demography such as age, gender, occupation. This data has been analyzed, you can see more details in notebooks/1.1-data-exploration.ipynb file. For learning purposes two sets of data were prepared from the original u.data file, which was splitted in proportion 80/20.

Model Implementation

To select a model, a grid search was performed on models performing the collaborative filtering task. The search was performed on the following models: SVD, K-Nearest Neighbors, K-Nearest Neighbors with Means, K-Nearest Neighbors with Z-Score. SVD had the best performance. The implementation was taken from library scikit-surprise.

Model Advantages and Disadvantages

Advantages:
1. SVD is a popular and widely used algorithm in recommendation systems.
2. It can handle sparse data effectively, which is important in recommendation systems where users typically rate only a small subset of items.
3. SVD can identify latent features that are not explicitly present in the data, which can improve the accuracy of recommendations.
4. It can be used for both item-based and user-based recommendations.
5. SVD can be easily scaled to large datasets using parallel computing techniques.

Disadvantages:
1. SVD requires a lot of computational resources, especially for large datasets.
2. It can be sensitive to noise and outliers in the data, which can lead to inaccurate recommendations.
3. SVD assumes that the data is normally distributed, which may not be true for all datasets.
4. It may not perform well for new users or items that have not been rated by many users.
5. SVD does not take into account contextual information such as time and location, which may be important in some recommendation systems.

Training process

scikit-surprise library provides a convenient interface for training a model, first an object of class Dataset is created which contains the data to be trained, then the fit method is called on the model which performs the training. The training itself takes less than one second on CPU.

Evaluation

To evaluate the model, a script was written in benchmark/evaluate.py. The model has the following metrics:

RMSE	MSE	MAE	FCP
0.9440	0.8911	0.7449	0.700

Results

As a result, I developed a model for recommending movies to users based on their ratings of the movies they have watched. The model showed good metrics and is ready to be used for the recommendation system.