Report

author: Polina Bazhenova: p.bazhenova@innopolis.university

Introduction

This report presents the recommendation system. It is a type of information filtering system that suggests items or content to users based on their interests, preferences, or past behavior. These systems are commonly used in various domains, such as e-commerce, entertainment, social media, and online content platforms. This recommendation system suggests movies to the user. It was created based on the MovieLens 100K dataset

Data analysis

The path in solution began with analyzing the datasets. After exploring them the following distributions were received. Figure 1 shows distribution of users by their ages. On Figure 2 there is rating distribution. Finaly, Figure 3 presents the dependencies of the number of existing movies on the genre.

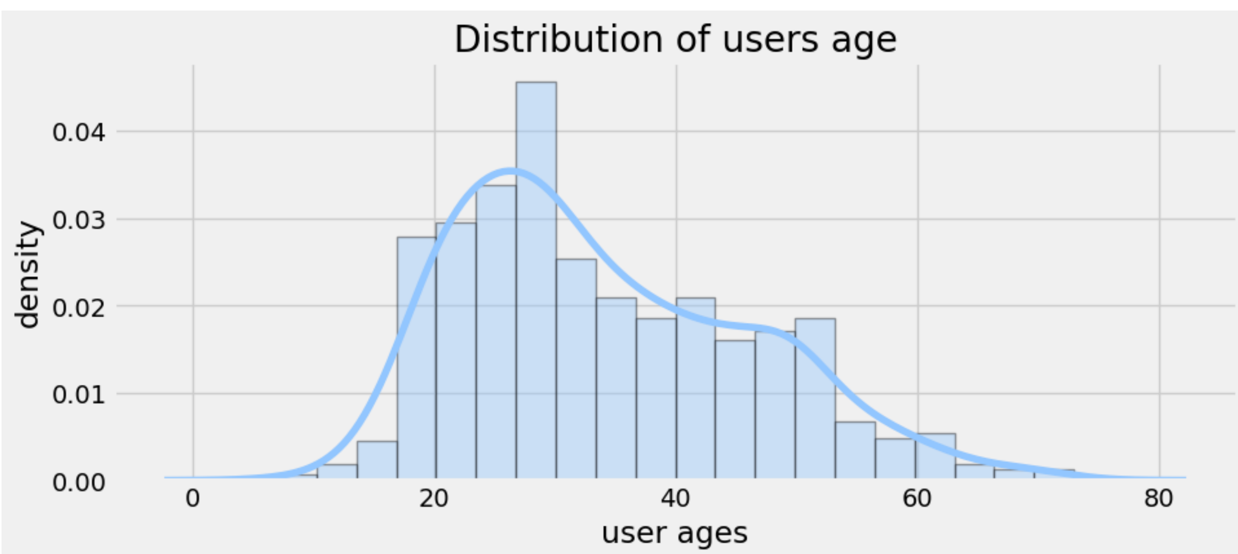

Figure 1

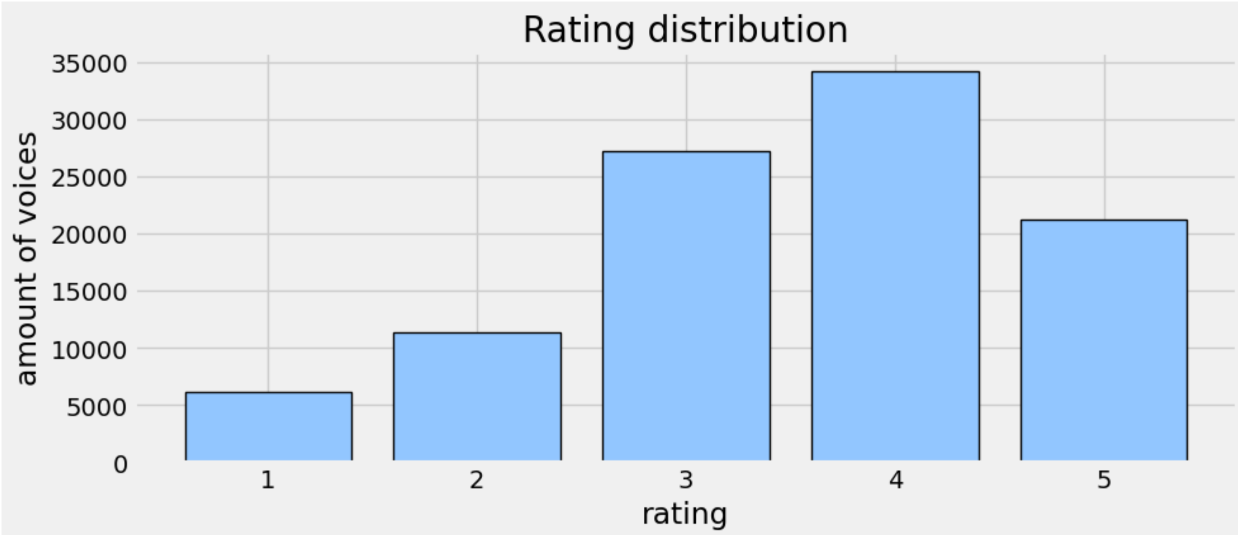

Figure 2

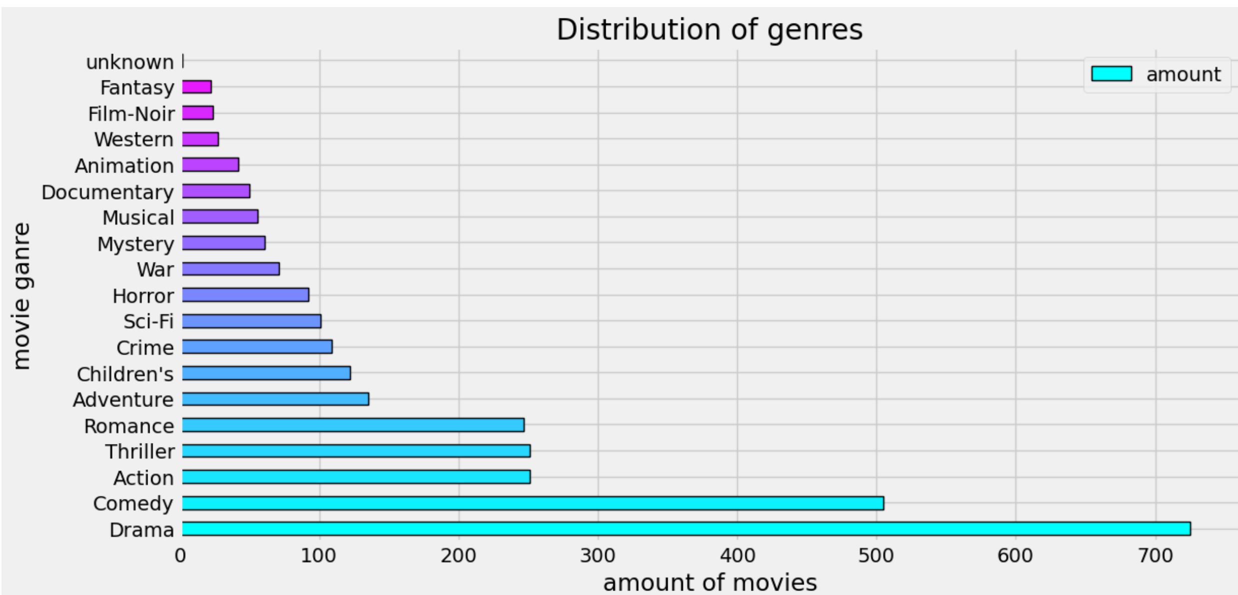

Figure 3

The data was converted to the weighted bipartite graph. In the graph, user and item (movie) nodes are connected by rating which user gave to the item (movie).

Model Implementation

The NGCFConv class implements the convolutional layers used in neural graph filtering models. It extends the `MessagePassing` class, a base class for defining message passing in GNNs. It utilizes message passing techniques, degree normalization, linear transformations, and dropout regularization.

In order to stack the NGCF convolution layers here was used the `RecSysGNN` class. It allows to change the number of layers of the model and the dropout. The ability of the model to defuse the information of recommendations between nodes depends on the number of layers.

Model Advantages and Disadvantages

This model archetecture offers several advantages and disadvantages.

Advantages:

- It leverages the graph structure. Hence,
- Usage of dropout reduces the risk of overfitting.
- The class design allows for bias in the linear layers and customization through various parameters.

Disadvantages:

- This model can be computationally intensive and memory demanding with large graphs.
- It is limited to Graph Data.

Training and Evaluation Process

For the train and evaluating process were used the Bayesian Personalized Ranking loss function for a single mini-batch of users, positive items, and negative items, as well as the `precision@K` and `recall@K` metrics, where `K` is the top K items we would like to recommend to user.

Model was trained on 3 epochs using cross validation method. In Figure 4 and Figure 5 you can see training loss results through epochs and evaluation recall precision correspondingly.

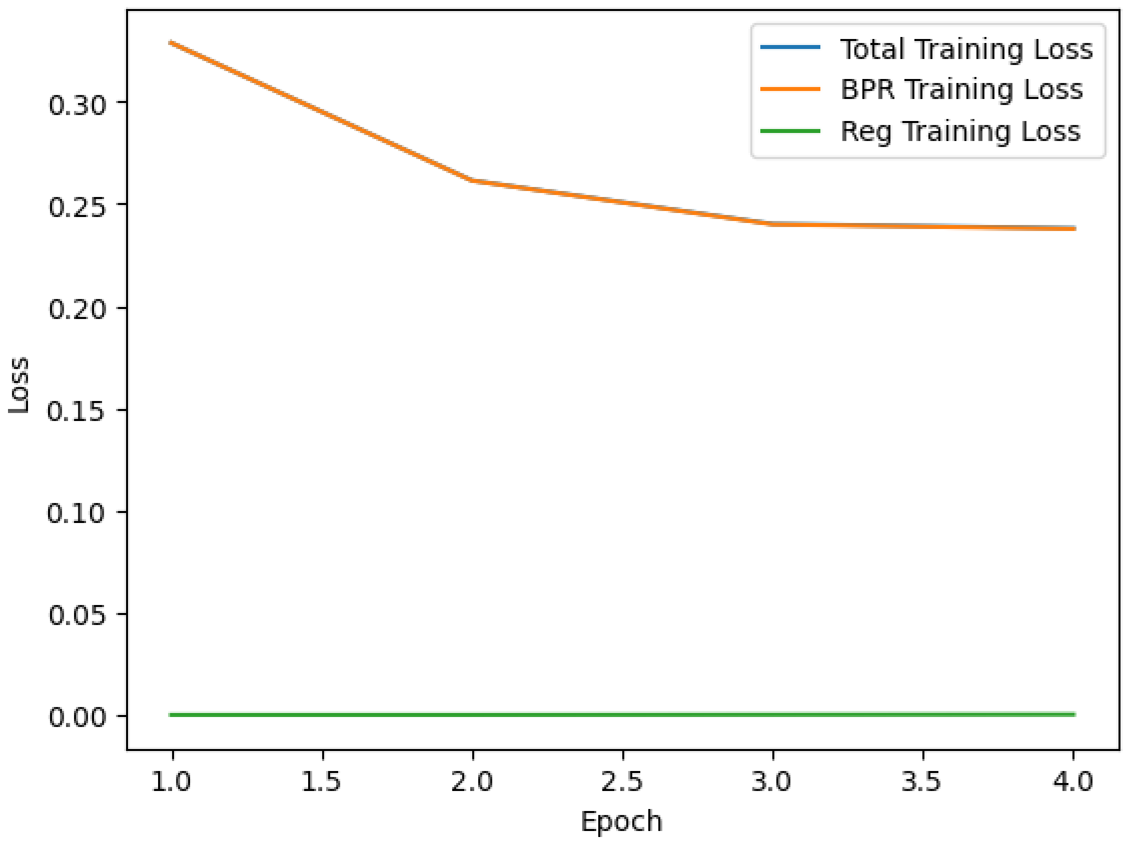
Total Training Loss
BPR Training Loss
Reg Training Loss
0.30
0.25
0.20
0.15
0.10
0.05
0.00
Loss
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Epoch

Figure 4

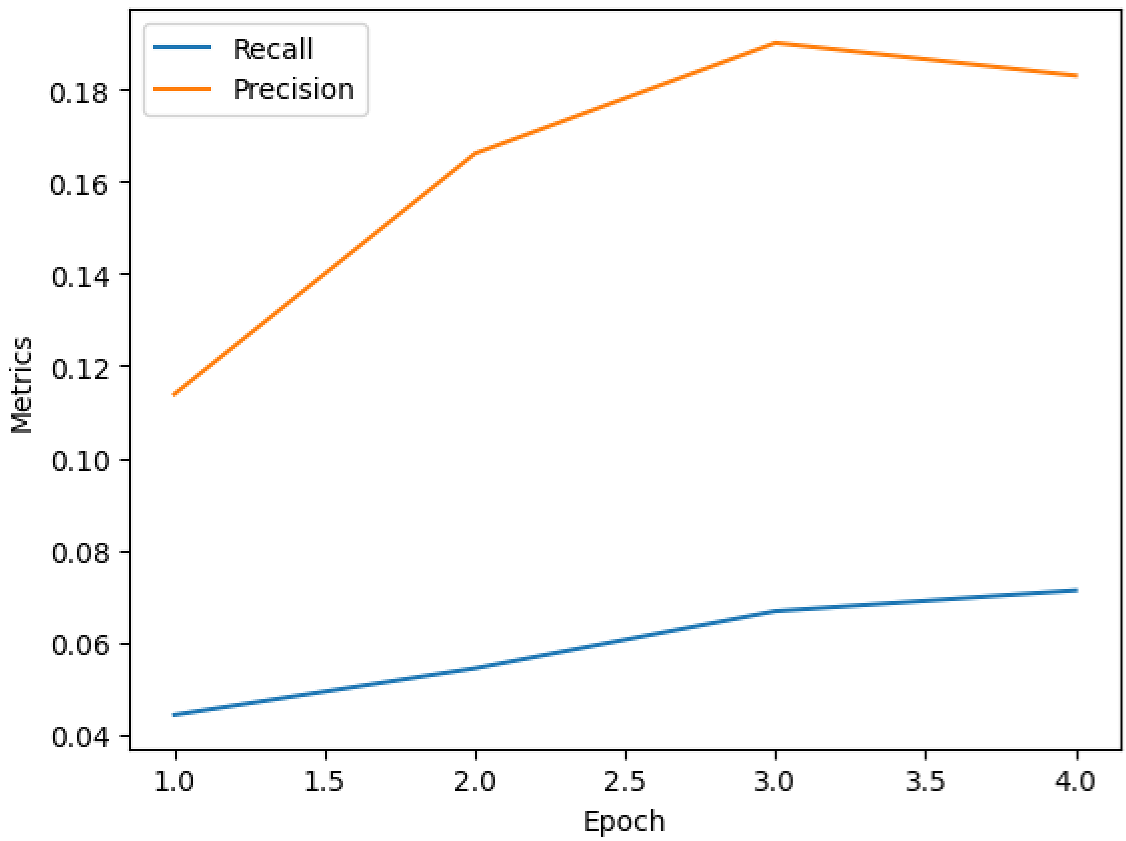

Figure 5

Results

In Figure 6 you can see top 10 predicted movies for five first and five last users above. The last row shows precision and recall of evaluation process.

```
      user_ID                                        top_rlvnt_itm
0           0  [935, 922, 926, 928, 941, 931, 810, 938, 911, 703]
1           1  [935, 922, 926, 928, 810, 941, 938, 931, 911, 703]
2           2  [935, 922, 926, 928, 941, 810, 931, 938, 911, 703]
3           3  [935, 922, 926, 941, 810, 928, 931, 938, 911, 703]
4           4  [935, 922, 926, 928, 941, 810, 938, 931, 911, 703]
...       ...                                                ...
1677     1677  [935, 922, 926, 938, 928, 810, 941, 882, 911, 916]
1678     1678  [935, 922, 926, 810, 911, 938, 928, 941, 512, 838]
1679     1679  [935, 922, 926, 810, 941, 938, 928, 911, 913, 931]
1680     1680  [935, 922, 926, 928, 938, 810, 931, 941, 911, 916]
1681     1681  [935, 922, 926, 810, 941, 939, 882, 916, 931, 928]

[1682 rows x 2 columns]
Precision: 0.02374727668845316, Recall: 0.005467816088117445
```

Figure 6

Reference

1. https://colab.research.google.com/drive/1VQTBxJuty7aLMepjEYE-d7E9kjo51CA1?usp=sharing#scrollTo=2ys1P7mtcr54
2. https://www.kaggle.com/code/dipanjandas96/lightgcn-pytorch-from-scratch/notebook