NANYANG TECHNOLOGICAL UNIVERSITY

Final Year Project Interim Report

Analysing Amazon Review Data and Constructing A Collaborative-Filtering based recommender system

Student: Yong Hao

U1722282A

1. Introduction

Recommender systems adopt various algorithms and design paradigms which aims to provide the most relevant items or information from the information pool the system accumulates along its operation. Recommender systems in ecommerce context have great importance in marketing products, increase product exposure to users, and boost sales of products.

In this report, we experimented building a recommender system using collaborative-filtering method upon the review records, whose schema contains the user ID, user rating of the item, item ID, and timestamp, of Amazon Review Data’s Magazine-Subscription section[1]. We selected this dataset for its relatively small size, and it is easy to compute and visualize without the consumption of much computational power.

We then evaluated the performance of such model.

2. Methodology

2.1. Analysis of the dataset

2.1.1. Unique users and products

Note the analysis of this section is subject to change in accordance with the dataset we use.

```
unique_users = len(np.unique(fashion_data.user_id))

unique_products = len(np.unique(fashion_data.product_id))
```

We examined the size and the distribution of unique users and unique products and acquired the following results:

Total No of Users	2428
Total No of products:	72098

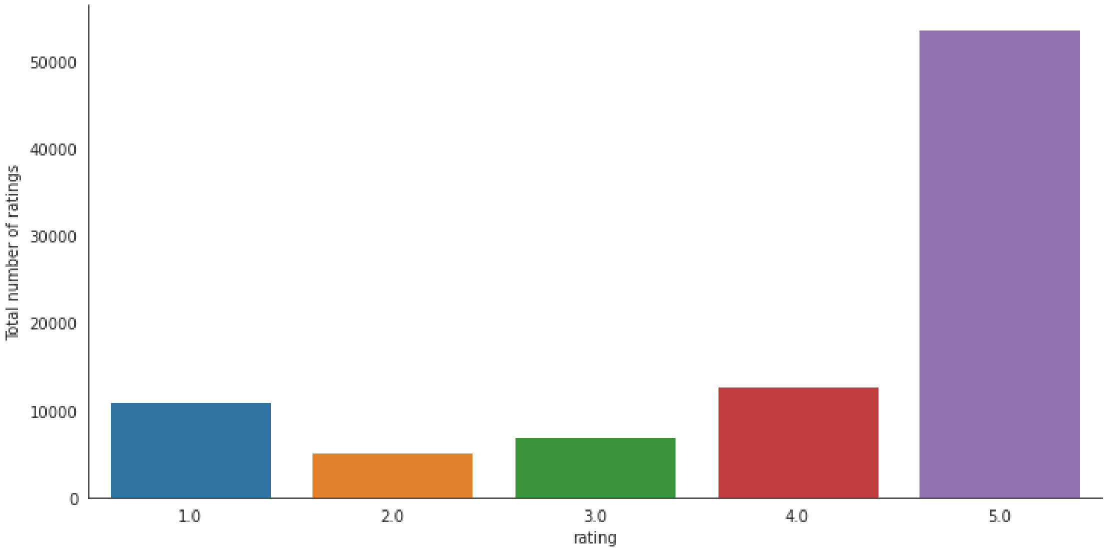
2.1.2. Rating Distribution

We checked the distribution of the ratings using the following code:

```
with sns.axes_style('white'):

   g = sns.factorplot("rating", data=fashion_data, aspect=2.0,kind='count')

   g.set_ylabels("Total number of ratings")
```



We can see from the above diagram, most rating are 5.0, with other ratings taking similar proportions.

A more detailed description on ratings:

count	89689.000000
mean	4.036638
std	1.419791
min	1.00000
25% (percentile, same as below)	3.00000
50%	5.00000
75%	5.00000
max	5.00000

2.2. Collaborative filtering

2.2.1. Item-item collaborative-filtering using k-nearest neighbors with means algorithm.

We experimented to build an item-item collaborative-filtering recommender model using KNN with mean algorithm to test the performance of such algorithm using the python scikit-learn surprise package. We evaluated the accuracy of the model in terms of root mean square error using 3-fold cross-validation method.

The settings of the algorithm are as following:

```
algo = KNNWithMeans(k=5, sim_options={'name': 'pearson_baseline', 'user_based': False}, verbose=True)
```

We tested the performance of KNN with 5 nearest neighbors, and the result is as following:

Fold	RMSE
1	1.3882114476233347
2	1.3673196574806734
3	1.3640057635730114
Mean	1.3731789562256732

2.2.2. Item-item collaborative-filtering using SVD.

We experimented to build an item-item collaborative-filtering recommender model using SVD algorithm to test the performance of such algorithm. We evaluated the accuracy of the model in terms of root mean square error using 3-fold cross-validation method.

The settings of the algorithm are as following:

```
algo_SVD = SVD(n_factors=5, biased=False, verbose = True)
```

We tested the performance of SVD algorithm, and the result is as following:

Fold	RMSE
1	0.7963266035488908
2	0.7884440536231141
3	0.7785707791693307
Mean	0.7877804787804452

The code for doing these experiments is summarised in the Jupyter notebook.

3. Future Work

3.1. Observations

1> Obtainable datasets often contain an excessive number of records, resulting in prolonged time of training the recommender model. An adoptable practice is to only include users who have more than 50 ratings to ensure the recommendations are significant. Taking a portion of the entire data is also viable.

2> For this particular dataset, the performance of SVD is significantly better than KNN, though the hyperparameters are randomly chosen for experiment uses.

4. Conclusion

After the analysis and experimenting building recommender systems using various algorithms, the efforts of this project can be directed to how to enhance the performance of the recommender system while choosing one algorithm as the baseline.
Possible approaches include:

1. Besides ratings of each item, we can include the sentiment of the review as a weighted rating to adjust integer-based ratings with human emotions.[2]

2. Integrate deep convolutional neural network with SVD algorithm to enhance the learning of features.

References

[1] Jianmo Ni, Jiacheng Li, Julian McAuley, “Justifying recommendations using distantly-labeled reviews and fined-grained aspects. Empirical Methods in Natural Language Processing (EMNLP)”, 2019

[2] G. Preethi, P. V. Krishna, M. S. Obaidat, V. Saritha and S. Yenduri, "Application of Deep Learning to Sentiment Analysis for recommender system on cloud," 2017 International Conference on Computer, Information and Telecommunication Systems (CITS), Dalian, 2017, pp. 93-97, doi: 10.1109/CITS.2017.8035341.