**3. Method**

**3.1 Data Feature Extraction and Selection**

We are using the MovieLens-100K [1] dataset, which has 100,000 ratings (1-5) from 943 users on 1682 movies. Each user has rated at least 20 movies. Simple demographic info for the users (age, gender, occupation, zip). The dataset divides into 5 group of data, each of the group has a train dataset and a test dataset, are 80%/20% splits of the total data set. The input of our recommendation system is three parts. One is the train dataset with the users rating for each of the movie, and the other two arrays including the label and feature information for the users and movies. The output is the prediction of the users’ rating information for each of the movie. We will try two different collaborative filtering techniques including Funk SVD and deep learning based collaborative filter.To preprocess the data, we removed some useless data such as the ***zip code*** for the user and ***IMDb URL*** for the movies. We also replace the feature of the user and movies into an integer number to represent the details information, which can be used in our collaborative filter. For example, transform the ***‘occupation’*** into 21 integers from **0-20**, and store these data in a new dataset.

**4. result Benchmark Result**

We implement our collaborative filtering system on the Funk SVD and Multilayer Perceptron (MLP). and the hyperparameter for the Funk SVD is the recommendation by Netflix Prize [2], with ***latent\_variable=8, learning\_rate=0.01, maximum\_epoches=25, regularization=0.005***. For MLP the hyperparameters are well tuned with one hidden layer and 40 hidden neurons, And we use Stochastic gradient descent (SGD) optimizer along with MSE loss to train our model. To evaluate the result of the experiment, we apply MAE（mean absolute error） and RMSE（root mean squared error) to test the accuracy of the model. Since the data set splits into 5 groups, we can evaluate the model in K-fold Cross Validation to derive a more accurate estimate of model prediction performance. The result shows in the Table 1 below:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Train set | U1 | U3 | U4 | U4 | U5 | Cross-val |
| Funk SVD | R: 0.95  M: 0.75 | R: 0.95  M: 0.74 | R: 0.94  M: 0.74 | R: 0.94  M: 0.74 | R: 0.93  M: 0.74 | R: 0.94  M: 0.74 |
| MLP | R: 0.15  M: 0.97 | R: 1.13  M: 0.94 | R: 1.11  M: 0.93 | R: 1.11  M: 0.93 | R: 1.11  M: 0.94 | R: 1.11  M: 0.94 |

Table 1: Result of the recommendation system

By observing the result of the experiment, we can find that both the Funk SVD and MLP could get a relatively good performance and the Funk SVD perform better than MLP model. However, compared to the benchmark result Table 2, there is still room for improvement.

|  |  |  |
| --- | --- | --- |
| Method | RMSE | Comment |
| GraphRec | 0.904 | Result from [2] |
| IGMC | 0.905 | Result from [3] |
| GC-MC | 0.91 | Result from [4] |
| Self-Supervised Exchangeable Model | 0.91 | Result from [5] |
| Factorized EAE | 0.92 | Result from [6] |

Table 2: Movie-lens 100k: Baseline result without extra training data in E1

**Visualization of Data**

After we implement the movie recommendation model, we design a data visualization method to display a small subset of the 1682 movies on a 2-dimensional plane to visualize their similarity. After the dimensional reduction by PCA and clustering by K-mean++, we visualized the 100 movie ratings randomly selected from the prediction matrix. Since the rating for a movie ranges from 1 to 5, we just simply set the clustering group into 5, and the result show below.

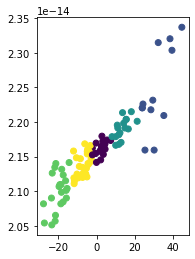


Figure 2 Visualization of 100 movie ratings

Obviously, we can find that there is a grouping phenomenon, in order to very that the grouping is valid, we calculate the mean rating for each of the movie show in table 3. We can find that in each of the group the mean rating for a movie is very similar there is an obvious stratification. With this phenomenon, we can evaluate the overview of popularity and public praise for a movie.

|  |  |
| --- | --- |
| Group | Mean rating |
| 1 | 2.8093897164163435 |
| 2 | 3.2229840335404067 |
| 3 | 3.503141915780966 |
| 4 | 3.7855726457000136 |
| 5 | 4.179602458975345 |

**5. Conclusion and Future Work:**

In this project, we propose and implement two collaborative filtering techniques for the movie recommendation system. We tuned the hyperparameter for these two algorithms and get a relatively good result. We could predict the complete rating information for all of the movies which enables us to recommend a movie for a user how has not watched this movie. On the other hand, to evaluate this result, we design the visualization method to view the dimensional reduced matrix by clustering it, and we find that the data could cluster into several groups according to the ratings. With this finding, we can evaluate the public praise for a movie. However, compared to the baseline results, our model, the result has relative slow accuracy, which can be improved in the future work.

Funk SVD makes it easy for us to find a way to evaluate our recommendation engine and create good U and transform of T matrices, so that we can make recommendations even if we have a very sparse matrix. But funk SVD should not be used alone, because we may face a common problem in recommendation System, which is called "cold start problem". This problem means that we can't recommend for new users or new movies. A good way is to combine funk SVD with less advanced methods, such as ranking based algorithm [7] or content-based algorithm, This can be the future work. Besides that, there are some more details feature for movie and the users, those features may also connect with the rating for a movie by users, these will be the extra training data, along with this data we can get a better result up to 0.88 in RMSE [8], which could perform better than our model.

**Reference**

[1] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets:History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015), 19 pages.DOI=http://dx.doi.org/10.1145/2827872

[2] Mukund Deshpande and George Karypis. 2004. Item-based top-n recommendation algorithms. ACM Transactions on Information Systems (TOIS) 22, 1 (2004), 143--177.

[3] Muhan Zhang, Yixin Chen. 2019 Inductive Matrix Completion Based on Graph Neural Networks,CLR-2020

[4] Rianne van den Berg, Thomas N. Kipf, Max Welling. 2017 Graph Convolutional Matrix Completion. https://github.com/riannevdberg/gc-mc

[5] Jason Hartford, Devon R Graham, Kevin Leyton-Brown, Siamak Ravanbakhsh, 2017 Deep Models of Interactions Across Sets https://github.com/mravanba/deep\_exchangeable\_tensors

[6] Jason Hartford, Devon R Graham, Kevin Leyton-Brown, Siamak Ravanbakhsh 2018 Deep Models of Interactions Across Sets. ICML 2018

[7] Steffen Rendle and Li Zhang. 2019the Difficulty of Evaluating Baselines A Study on Recommender Systems.