COMP4332 Project 3: Rating Prediction

TRINH Van Hoan, NGUYEN Kha Nhat Long, DO Van Quyet
The Hong Kong University of Science and Technology
{vhtrinh, knlnguyen, vqdo}@connect.ust.hk

1. Introduction

For an online business platform, a recommendation system with information regarding users and products can be used to suggest relevant items. This task is important for businesses with massive amounts of data such as Google, Netflix, Amazon, or Yelp. In this project, we will work on the **Yelp Dataset**, in order to predict ratings for businesses based on various user and business data. The overall performance will be measured using the root mean square error (RMSE), between the predictions and the actual ratings.

We first perform some data exploration, build a Neural CF model for this project and then try Wide and Deep learning, a more complex model.

2. Data Observation

There are three major parts of the data: ratings, users, and businesses. The train data consists of the stars given by the users for the businesses. Extra information for both users and businesses contains different information and is identified by their IDs.

It is observed that the training data consists of 60080 rows, while the validation and the testing data consists of 7509 entries. The information provided in the training.csv has the user id of a certain user, the business id of the business being reviewed, and the corresponding number of stars (1 to 5) provided by the users for the business.

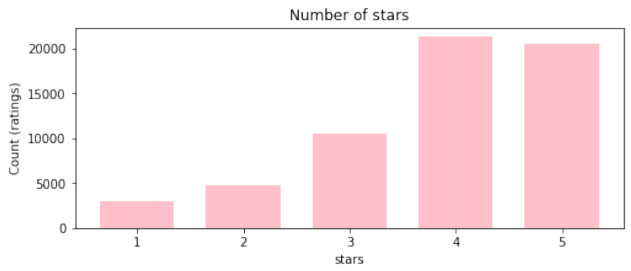


Figure 1: Histogram of the ratings

The ratings is skewed towards 4 and 5 stars.

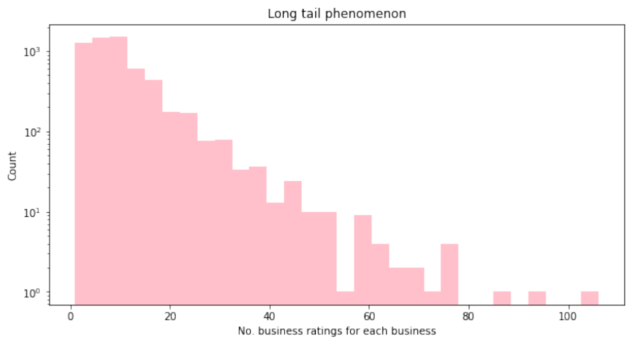
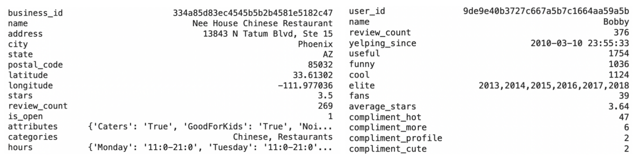


Figure 2: Long Tail phenomenon for recommendation

There are 3602 businesses (5938 in total) receiving less than 10 reviews.

The business.csv file provides additional information for businesses. Each business is identified by their business id, with some notable information are: name, location, attributes of the business, type of business (categories), opening hours, review count, average stars received, and state of operation. For the user.csv file, the following information was provided: average stars given, the number of user interactions (e.g. cute, hot, more), the number of fans, and the number of useful, funny, and cool votes.



```
business_id                             334a85d83ec4545b5b2b4581e5182c47
name                                        Nee House Chinese Restaurant
address                                      13843 N Tatum Blvd, Ste 15
city                                                             Phoenix
state                                                                 AZ
postal_code                                                        85032
latitude                                                        33.61302
longitude                                                    -111.977036
stars                                                                3.5
review_count                                                         269
is_open                                                                1
attributes        {'Caters': 'True', 'GoodForKids': 'True', 'Noi...
categories                                           Chinese, Restaurants
hours             {'Monday': '11:0-21:0', 'Tuesday': '11:0-21:0'...
```

```
user_id                  9de9e40b3727c667a5b7c1664aa59a5b
name                                                Bobby
review_count                                          376
yelping_since                         2010-03-10 23:55:33
useful                                               1754
funny                                                1036
cool                                                 1124
elite                       2013,2014,2015,2016,2017,2018
fans                                                   39
average_stars                                        3.64
compliment_hot                                         47
compliment_more                                         6
compliment_profile                                      2
compliment_cute                                         2
```

Figure 3: Extra information for users and businesses

The information provided is a mix of continuous, categorical and textual data. When we build the NCF model, we do not need to preprocess the data. However, for the Wide and Deep model, extensive data pre-processing is required to transform the data to usable features.

3. Experiments

3.1. Neural Collaborative Filtering

We started with Neural Collaborative Filtering (NCF).

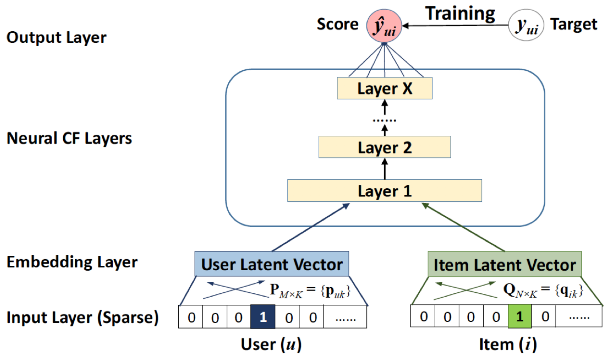


Figure 4: Neural Colaborative Filtering Model

The model embeds user and item into vector space and concatenate them into the input layer and run the neural network. Here, we use a 3-layer network.

We used Grid Search on different sets of hyperparameters, focusing on the hidden layer size of the neural network and the dropout rate.

Specifically, the number of nodes in the hidden layer is tested with 64,128,256,512 and the dropout rate is tested between 0.2 and 0.35.

3.2. Wide and Deep Learning

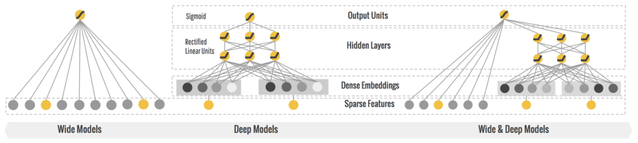


Figure 5: Wide and Deep learning

This model is based on the **Wide and Deep learning** released by Google for Recommender Systems. This model has two main sections, the Deep part and the Wide part.

The Deep part is divided into continuous features and categorical features. Most of the categorical features are extracted from item_attributes, which contains details about the item. For the continuous features, a scaling is performed to normalize the values. For each categorical feature, we add an embedding layer. These embeddings are then concatenated with the continuous features. Afterwards, we feed the embedded values into an MLP network. The Wide part utilizes the values inside item_categories. It extracts all categories and filters the most common ones (both singles and 2,3,or 4-combinations). Next, the wide features are obtained by generating an array of binary outputs, whose values corresponds to the selected top common by assigning 1 if the item belongs to the category(s) and 0 otherwise. Finally, we concatenate the outputs from Deep and Wide parts and output it to a Dense layer of size 1, which is the rating.

We manually tested with different sets of input columns (deep continuous, deep categorical, wide features) e.g. including and not including attributes about item's location, item's opening hours, user's personal information. We also used Grid Search for other parameters. Specifically, the number of layers (1,2,3), hidden size (128,256,1024), embedding size (10,30,50), dropout rate (0.1,0.2,0.3), other parameters we kept unchanged. We found that for different settings, the performance of this model did not change significantly, and none of the settings surpasses the previous model, which is NCF.

All experimental results can be found in Table 1.

3.3. Results Summary and Analysis

Models	Validation RMSE
NCF (32,0.2,256,0.2)	1.062
NCF (32,0.2,128,0.2)	1.064
NCF (32,0.2,64,0.2)	1.070
NCF (32,0.2,32,0.2)	1.076
NCF (32,0.24,256,0.22)	**1.061**
Wide and Deep	1.13

Table 1: Experimental results for the two models.

As can be seen in Table 1, when tuning hyperparameters the NCF model, we see that the size of two hidden layer (32,256) give a much better result. After that, we tune the this model with different dropout rate to get the best parameter list (32,0.24,256,0.22), which yields RMSE as **1.061**.

It is also observed that the simple model NCF has better result than the complex model Wide and Deep, which means using the relationship between users and items (only collaborative filtering) is enough to get good results, while using extra user and item features with complicated model seems to decrease the performance.

4. Conclusion

In this project, we first explored the data to observe the long-tail phenomenon of recommendation and get some insights about the dataset. Based on that, we applied two models NCF and Wide and Deep. Overall, the result of the simple NCF model is significantly better than Wide and Deep. Therefore, we chose NCF as our final model to produce the predictions.