CIS Algorithm and Applications of Data Mining

Assignment 3

**1. Overview of the Assignment**

In Assignment 3, you will complete three tasks. The goal is to let you be familiar with various types of recommendation systems.

**2. Requirements**

You must use Python 3.6+ and Spark 3.0+ to implement all tasks.

**3. Datasets**

In this assignment, we generated the review data from the original Yelp review dataset with some filters, such as the condition: “state” == “CA”. We randomly took 80% of the data for training, 10% of the data for testing, and 10% of the data as the blind dataset.

You can access and download the following JSON files in the Google Drive: <https://drive.google.com/drive/folders/11gtveW96LZxtB8NxOfdgzaTSJsSqOdKf?usp=sharing>

1. train\_review.json
2. test\_review.json – containing only the target user and business pairs for prediction tasks
3. test\_review\_ratings.json – containing ground truth rating for the testing pairs (evaluation purpose)
4. user\_avg.json – containing the average stars for the users in the train dataset
5. business\_avg.json – containing the average stars for the businesses in the train dataset
6. We do not share the blind dataset.

**4. Task: Collaborative Filtering Recommendation System**

You need to turn in the following files:

1. [REQUIRED] Python scripts: train.py, predict.py
2. [REQUIRED] Result files: task1.model, task1.res
3. [OPTIONAL] You can include other scripts to support your programs (e.g., callable functions).

4.1 Task description

In this task, you will build collaborative filtering recommendation systems with train reviews and use the model to predict the ratings for a pair of user and business. You are required to implement the “Item-based CF recommendation system”.

During the learning process, you will build a model by computing the Pearson correlation for the business pairs that have **at least three co-rated users**. During the predicting process, you will use the model to predict the rating for a given pair of user and business. You must use **at most three business neighbors** that are most similar to the target business for prediction.

4.1.2 Execution commands

Training commands:

|  |  |
| --- | --- |
| Python | $ python train.py --train\_file <train\_file> --model\_file <model \_file> |
|  | <train\_file>: the train review set  <model\_file>: the output model |

Predicting commands:

|  |  |
| --- | --- |
| Python | $ python predict.py --train\_file <train\_file> --test\_file <test\_file> --model\_file <model\_file> --output\_file <output\_file> |
|  | <train\_file>: the train review set  <test\_file>: the test review set (only target pairs)  <model\_file>: the model generated during the training process  <output\_file>: the output results |

4.1.3 Output format:

Model format:

You must write the model in the JSON format using **exactly the same tags as the** **example in Figure 1**. Each line represents for a business pair (“b1”, “b2”) for item-based model. Pairs should be in alphabetical order, i.e., there is no need to have (“b2”, “b1”).

A close up of a logo

Description automatically generated

Figure 1: An example item-based model

Prediction format:

You must write a target pair and its prediction in the JSON format using **exactly the same tags as the** **example in Figure 2**. Each line represents for a predicted pair of (“user\_id”, “business\_id”).

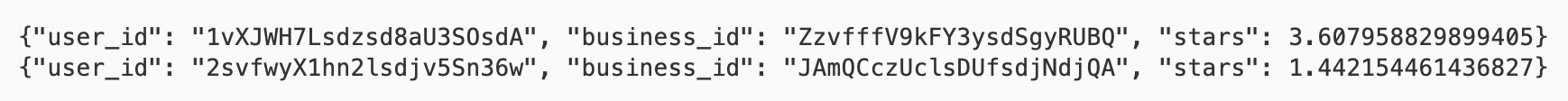


Figure 2: An example output for task3 in JSON format

4.1.4 Grading

You should be able to generate the item-based model. We will compare your prediction results against the ground truth. You should **ONLY** output the predictions that can be generated from the model. For those pairs that your model cannot predict (e.g., due to cold start problem or too few co-rated users), **we will first predict them with the business average stars for the item-based model**. We provide the file containing the average stars for businesses in the training dataset. There is a tag **“UNK”** that is the overall average stars, which can be used for predicting those new businesses and users. Then we use RMSE (Root Mean Squared Error) to evaluate the performance as the following formula:

Where is the prediction for business and is the true rating for business . is the total number of the user and business.

The suggested execution time of the training process is less than 600 seconds. The execution time of the predicting process should be less than 100 seconds. **The RMSE requirements is 0.9 for the testing data.**