

# INF553 Foundations and Applications of Data Mining

Fall 2019

## Competition Project

**Deadline: Dec 8<sup>th</sup> 11:59 PM**

### 1. Overview of the Competition Project

In this competition project, you need to improve the performance of your recommendation system in Assignment 3. You can use any method (like the hybrid recommendation systems) to improve the prediction accuracy and efficiency.

### 2. Competition Requirements

#### 2.1 Programming Language and Library Requirements

- a. **You must use Python to implement the competition project.** You can use external Python libraries as long as they are available on Vocareum.
- b. You are required to only use the Spark RDD for Spark operations. **You will not receive any point if you use Spark DataFrame or DataSet.**

#### 2.2 Write your own code

**Do not share your code with other students!!**

We will combine all the code we can find from the Web (e.g., GitHub) as well as other students' code from this and other (previous) sections for plagiarism detection. We will report all the detected plagiarism.

### 3. Yelp Data

In this competition, the datasets you are going to use are in the Google drive:

<https://drive.google.com/drive/folders/1-ZJ0vFGuvKqOxO-vp7lq-zkMj4tkyNnG?usp=sharing>

We randomly took 60% of the data as the training dataset, 20% of the data as the validation dataset, and 20% of the data as the testing dataset.

- A. yelp\_train.csv: the training data, which only include the columns: user\_id, business\_id, and stars.
- B. yelp\_val.csv: the validation data, which are in the same format as the training data.
- C. We do not share the testing dataset, which only include the columns: user\_id and business\_id
- D. Other datasets: providing additional information (like the location of a business)
  - a. review\_train.json: review data only for the training pairs (user, business)
  - b. user.json: all user metadata
  - c. business.json: all business metadata, including locations, attributes, and categories

- d. checkin.json: user checkins for individual businesses
- e. tip.json: tips (short reviews) written by a user about a business
- f. photo.json: photo data, including captions and classifications

#### 4. Task (8 points)

In the competition, you need to build a recommendation system to predict the given (user, business) pairs. You can mine interesting and useful information from the datasets provided in the Google Drive folder “Competition” to support your recommendation system.

You must make an improvement to your recommendation system in terms of **accuracy**. You can utilize the validation dataset (yelp\_val.csv) to evaluate the accuracy of your recommendation system. There are two options to evaluate your recommendation system:

**1) Error Distribution:** You can compare your results with the corresponding ground truth and compute the absolute differences. You can divide the absolute differences into 5 levels and count the number for each level as the following:

>=0 and <1: 12345  
 >=1 and <2: 123  
 >=2 and <3: 1234  
 >=3 and <4: 1234  
 >=4: 12

This means that there are 12345 predictions with < 1 difference from the ground truth. This way you will be able to know the error distribution of your predictions and to improve the performance of your recommendation system.

**2) RMSE Error:** You can compute the RMSE (Root Mean Squared Error) by using following formula:

$$RMSE = \sqrt{\frac{1}{n} \sum_i (Pred_i - Rate_i)^2}$$

Where  $Pred_i$  is the prediction for business  $i$  and  $Rate_i$  is the true rating for business  $i$ .  $n$  is the total number of the business you are predicting.

#### Input format: (we will use the following command to execute your code)

Spark-submit competition.py <input\_path> <test\_file\_name> <result\_file\_name>

Param <input\_path>: the path of the data folder (e.g., Competition/), which contains the exact same files on the Google drive except the yelp\_val.csv

Param <test\_file\_name>: the name of the testing file (e.g., yelp\_val.csv), including the file path

Param: <result\_file\_name>: the name of the prediction result file, including the file path

#### Output format:

a. The output file is a CSV file, containing **all the prediction results for each user and business pair** in the validation/testing data. The header is “user\_id, business\_id, prediction”. There is no requirement for the order in this task. There is no requirement for the number of decimals for the similarity values. Please refer to the format in Figure 2.

```
user_id, business_id, prediction
C5QsUsQg5I3dMdLM02SXGA, PvGyzCh1PTga4ePE2-iB2Q, 5.0
oxd0FmY0YWW4gFq5jJr-hg, ZSCEkqlzZKRrZUz98CXtNw, 2.804287677476818
GGTF7hnQi6D5W77_qiKlqg, 5PyqkF8zZbfgFDyAcLUehQ, 4.688318401935079
```

Figure 2: Output example in CSV

b. You also need to write comments includes the description of your method (less than 300 words) **at the first part of your program**. The description should include the explanation to the models you are using, especially the way you improve the accuracy or efficiency of the system. We look forward to seeing creative methods. Please also report the error distribution, RMSE, and the total execution time on the validation dataset in the description. Figure 3 shows an example of the description file. **If the comments are not included or the comments are not informative, one-point penalty will be given.**

```
Method Description:
...

Error Distribution:
>=0 and <1: 12345
>=1 and <2: 123
>=2 and <3: 1234
>=3 and <4: 1234
>=4: 12

RMSE:
1.11

Execution Time:
200s
```

Figure 3: An example of description file

### **Grading:**

We will compare your prediction results against the ground truth. we will use **our testing data** to evaluate your recommendation systems and grade based on the accuracy using RMSE.

To get the full points for the competition project, **your RMSE result should beat TAs’**. TAs will also continuously improve their systems and will announce the accuracy. The result will be fixed **three days before the competition due**. However, if your recommendation system only beats the TAs’ for the validation data, you will receive **50% of the points for the competition**.

You should post your accuracy result for the validation data on the Discussion board “Competition project” and compete with TAs’ and other students’ results. The final submission with the highest accuracy will receive extra **6 points on the final grade**. The second place will receive extra **5 points**. The third one will receive extra **4 points** and so on until the sixth one will receive extra 1 point.

**NOTICE:** Current RMSE baseline is **1.03** for the validation dataset.

## **5. Submission**

You need to submit your Python script on Vocareum with exactly the same name: competition.py

## 6. Grading Criteria

(% penalty = % penalty of possible points you get)

1. You cannot use the extension for the competition. No late for the competition.
2. If we cannot run your programs with the command we specified (including version conflicts and library issues) on Vocareum, you will not receive point for the competition.
3. If the header of the output file is missing, there will be 20% penalty.
4. We will not regrade on the competition.
5. There will be no point if the total execution time exceeds **25 minutes**.