DSCI 553: Foundations and Applications of Data Mining

Spring 2024

Competition Project

Deadline: May 3rd 11:59 PM PST

Note: make sure submission data (yelp_val.csv) is NOT available to students and that the submission report shows the RMSE that the student achieved

1. Overview of the Assignment

In this competition project, you need to improve the performance of your recommendation system from 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 any external Python libraries as long as they are available on Vocareum.

b. You are required to only use the Spark RDD to understand Spark operations. You will not receive any points if you use Spark DataFrame or DataSet. However, if an external python library requires a separate data structure you may use it to load the data into the library, but make sure to do all data pre/post-processing using a Spark RDD.

2.2 Programming Environment

Python 3.6, Scala 2.12, JDK 1.8 and Spark 3.1.2

We will use these library versions to compile and test your code. There will be a **20% penalty** if we cannot run your code due to the library version inconsistency.

2.3 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 from:
https://drive.google.com/drive/folders/1SIlY40owpVcGXJw3xeXk76afCwtSUx11?usp=sharing

We generated the following two datasets from the original Yelp review dataset with some filters. 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 training data.

C. We are not sharing the test dataset.

D. other datasets: providing additional information (like the average star or 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 to support your recommendation system.

You **must make an improvement** to your recommendation system from homework assignment 3 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 to 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 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 systems.

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

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

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 commands to execute your code)

/opt/spark/spark-3.1.2-bin-hadoop3.2/bin/spark-submit competition.py <folder_path> <test_file_name> <output_file_name>

Param: folder_path: the path of dataset folder, which contains exactly the same file as the google drive

Param: test_file_name: the name of the testing file (e.g., yelp_val.csv), including the file path

Param: output_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 1.

```
user_id, business_id, prediction
C5QsUsQg5I3dMdLM02SXGA,PvGyzCh1PTga4ePE2-iB2Q,5.0
oxdOFmY0YWW4gFq5jJr-hg,ZSCEkqlzZKRrZUz98CXtNw,2.804287677476818
GGTF7hnQi6D5W77_qiKlqg,5PyqkF8zZbfgFDyAcLUehQ,4.688318401935079
```

Figure 1: Output example in CSV

b. You also need to write comments that include the description of your method (less than 300 words) **in the first part of your program**. The description should include the explanation of the models you are using, especially the way you improved 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 2 shows an example of the description file. **If the comments are not included or the comments are not informative, there will be a one-point penalty.**

```
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 2: 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 that of the TAs' which is 0.9800 for testing data**. If your recommendation system **only beats .9800 for the validation data**, you will receive **4 points for the competition**.

The final submission with the highest accuracy will receive an **extra 6 points on the final grade**. The second place will receive an extra 5 points. The third one will receive extra 4 points and so on until the sixth one will receive extra 1 point.

To be more like a competition, you can see a "Leaderboard" button in the "Competition" on Vocareum. Every time you submit the code, your **RMSE** for **validation data** will be scored and show up on the leaderboard. You will have the option to choose your display name on the leaderboard.

Partial credit will be given if your RMSE for testing and submission data cannot achieve the threshold. If your homework 3 accuracy is x, and competition is y, and you do not meet the threshold, you would get (1-(y-0.98)/(x-0.98))*4.

5. Submission

You need to submit your **Python scripts on Vocareum and NOT the trained model** 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 submissions will be accepted for the competition.
2. 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. If plagiarism is detected, you will receive no points for the entire assignment and we will report all detected plagiarism.
3. All submissions will be graded on Vocareum. Please strictly follow the format provided, otherwise you won't receive points even though the answer is correct.
4. Do **NOT** use Spark DataFrame, DataSet, sparksql.
5. We will not conduct regrades on competition submissions.
6. There will be no points awarded if the total execution time exceeds **25 minutes**.

7. Common problems causing fail submission on Vocareum/FAQ

(If your program runs seem successfully on your local machine but fail on Vocareum, please check these)

1. Try your program on Vocareum terminal. Remember to set python version as python3.6,

```
export PYSPARK_PYTHON=python3.6
```

And use the latest Spark

/opt/spark/spark-3.1.2-bin-hadoop3.2/bin/spark-submit

2. Check the input command line format.

3. Check the output format, for example, the header, tag, typos.

4. Your Python script should be named as competition.py

5. Check whether your local environment fits the assignment description, i.e. version, configuration.

6. If you implement the core part in Python instead of Spark, or implement it in a high time complexity way (e.g. search an element in a list instead of a set), your program may be killed on Vocareum because it runs too slowly.