Inf553 – Foundations and Applications of Data Mining

Fall 2018
The 4^{nd} USC Informatics Data Mining Competition

Starting Date: Oct 12 Friday 2018
End Date: Nov 29 Thursday 2018 11:59 PM PST

1 Competition Overview

This competition is based on the assignment 3 recommendation system. You need to keep improving the performance of your recommendation system on Yelp challenge dataset. You can use any method (including SparkMLlib) to improve your rating prediction. You can use Scala or Python for this competition.

Environment Requirements

Python: 2.7 Scala: 2.11 Spark: 2.3.1

Student must use python to complete both Task1 and Task2.

There will be 10% bonus if you use Scala for both Task1 and Task2 (i.e. 10 - 11; 9 - 10).

IMPORTANT: We will use these versions to compile and test your code. If you use other versions, there will be a 20% penalty since we will not be able to grade it automatically.

Write your own code!

For this assignment to be an effective learning experience, you must write your own code! I emphasize this point because you will be able to find Python implementations of most or perhaps even all of the required functions on the web. Please do not look for or at any such code!

TA will combine some python code on Github which can be searched by keyword ”INF553” and every students’ code, using some software tool for detecting Plagiarism.

Do not share code with other students in the class!!

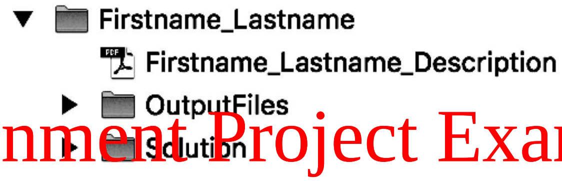
Submission Details

For the competition you will need to turn in a Python or Scala program depending on your language of preference.

Your submission must be a .zip file with name: **Firstname_Lastname_competition.zip**. The structure of your submission should be identical as shown below.

The Firstname_Lastname_Description.pdf file contains helpful instructions on how to run your code along with other necessary information as described in the following sections. This file also need to contain the description of the method you use to improve the performance, and other detail of the implementation, as detailed as possible

The OutputFiles directory contains the deliverable output files for each problem and the Solution directory contains your source code.



```
▼ Firstname_Lastname
      Firstname_Lastname_Description
  ▶ OutputFiles
  ▶ Solution
```

Figure 1: Submission Structure

Data

In this assignment, we will use the yelp challenge dataset, please download the ”yelp challenge data” from this link: Yelp Challenge. In order to download the dataset, you need to use your email to sign up individually in the Yelp challenge website. Detailed introduction of the data can also be found through the link, in the document tab. After download and unzip the data, the dataset contain 6 .json file and two .pdf file.

In this assignment, need the reviews.json file and three columns of the review will be used: user_id, business_id, stars.

About Competition

In the competition of the recommendation system, you can use all other file in this dataset to improve the performance. You can use the property of the user, business, or even tips and use any method you know to make the improvement. For instance, you can use different hybrid recommendation system mentioned in the lecture, or some machine learning methods like regression to make the improvement.

However, you can only use the information related to the user and business in the training file. You cannot use other users and businesses that don’t in the training file to make the improvement.

Here is the useful link contain some example of the dataset Yelp Dataset Examples. And also you can find some papers using the dataset Paper about Dataset.

More details of the Yelp dataset and challenge can be find from the official website. You still have a chance to win rewards, Fight on!

Dataset Description

yelp_academic_dataset_business.json : 188,593 records
Attributes: Business ID, address, name, city, Business hours, Categories, rating and reviews_count
yelp_academic_dataset_review.json : 5,996,996 records
Attributes: review ID, user ID, business ID, rating, comments
yelp_academic_dataset_user.json : 1,518,169 records
Attributes: user ID, name, review_count, Yelp_join_date
yelp_academic_dataset_checkin.json : 157,075 records
Attributes: Business ID, time
yelp_academic_dataset_tip.json : 1,185,348 records
Attributes: user ID, Business ID, text, likes, date
yelp_academic_dataset_photo.json : 280,992 records
Attributes: photo ID, Business ID, text

1.1 Task of Recommendation System

The task of this the recommendation system is to use the records in the train.csv to **predict** the stars for businesses in the test.csv. Then, you need to use the stars in testing data as the ground truth to evaluate the accuracy of your recommendation system.

Example: Assuming train.csv contains 1 million records and the test.csv contains two records: (12345, 2, 3) and (12345, 13, 4). You will use the records in the train.csv to train a recommendation system (1 million). Finally, given the user_id 12345 and business_id 2 and 13, your system should produce rating predictions as close as 3 and 4, respectively.

You are going to predict the testing datasets mentioned above. In your code, you can set the parameters yourself to reach a better performance. You can make any improvement to your recommendation system: **speed**, **accuracy**.

After achieving the prediction for ratings (stars), you need to compare your result to the correspond ground truth and **compute the absolute differences**. You need to divide the absolute differences into 5 levels and count the number of your prediction for each level as following:

>=0 and <1: 12345 (there are 12345 predictions with a < 1 difference from the ground truth)

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

Additionally, you need to compute the RMSE (Root Mean Squared Error) by using following formula:

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

Where $Pred_i$ is the prediction for movie i, $Rate_i$ is the true rating for movie i, n is the total number of the movies. Read the Microsoft paper mentioned in class to know more about how to use RMSE for evaluating your recommendation system.

Result Format

1. Save the prediction results in a text file. The result is ordered by **user_id** and **business_id** in ascending order.

Example Format:
$user_1, business_2, prediction_{12}$
$user_1, business_3, prediction_{13}$
$\dots$
$user_n, business_k, prediction_{nk}$

2. **Print the accuracy information** in terminal, and **copy this value** in your description file.

>=0 and <1: 12345
>=1 and <2: 123
>=2 and <3: 1234
>=3 and <4: 1234
>=4: 12
RMSE: 1.23456789
Time: 123 sec

Description File

Please include the following content in your description file:
1. Mention the Spark version and Python version
2. Describe how to run your program for both tasks
3. The precision and recall.
4. Same baseline table as mentioned in task 2 to record your accuracy and run time of programs in task 2

5. If you make any improvement in your recommendation system, please also describe it in your description file.

Submission Details

Your submission must be a .zip file with name: Firstname_Lastname_hw3.zip
Please include all the files in the right directory as following:
1. A description file: Firstname_Lastname_desription.pdf
2. All Scala scripts:
Firstname_Lastname_competition.scala
3. A jar package for all Scala file: Firstname_Lastname_competition.jar
If you use Scala, please make all *.scala file into ONLY ONE Firstname_Lastname_competition.jar file and strictly follow the class name mentioned above. And DO NOT include any data or unrelated libraries into your jar.
4. If you use Python, then all python scripts:
Firstname_Lastname_competition.py
5. Required result files for competition:
Firstname_Lastname_competition.txt

Ranking Criteria

Every week, Yuanbin will check the submission of the competition and post the result on the discussion board, the format will like a Ladder.

We will rank the competition based on your RMSE accuracy. After the last day of the competition, the submission having the highest accuracy will receive a 4% bonus on their final grade. The 2nd highest accuracy will receive a 3% bonus on their final grade. The 3rd highest accuracy will receive a 2% bonus on their final grade. Others will receive a 1% bonus on their final grade. In addition, you need to beat Yuanbin's system to be able to receive the bonus. Yuanbin will continuously improve her system and will announce her accuracy every Friday along with the rankings for the previous week.