ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

PROJECT MILESTONE 1: BASELINE RECOMMENDATION WITH GLOBAL AVERAGING

CS-449 SYSTEMS FOR DATA SCIENCE

Nour Ghribi, Sciper: 250232

19th March 2021

For making the code more modular there is package object called utils (can be found in src/main/scala/utils/utils.scala) that contains all the functions used and relevant methods. In addition, the case class Rating is defined in this package so that it is a global type seen by all the three scala files:
src/main/scala/stats/Analyzer.scala
src/main/scala/predict/Predictor.scala
src/main/scala/recommend/Recommender.scala.

All the functions defined in utils have their respective description that can be generated as scala doc by the doc command of sbt. "sbt:m1_250232> doc"

CHAPTER 1

MILESTONE 1

1.1 ANSWERS:

Q.3.1.1

On average the ratings do not coincide with the middle of the $(1, 2, 3, 4, 5)$ scale, the average rating is approximately 3.53 so on average ratings are higher by 0.53 than the middle of the scale.

Q.3.1.2

We can see that there is quite a disparity in users average ratings. Some are quite low compared to the global average rating and some tend to have a larger average than the global average thus rate more positively.
The average of the average ratings per user is ≈ 3.59, the minimum is ≈ 1.5 and the maximum is ≈ 4.87 .
Considering ratings that deviate with less than 0.5 from the global average close: The ratio of users that are close to the global average is: ≈ 0.75 that is, we have a $3 : 4$ ratio approximately of users average ratings that are close to the global rating.

Q.3.1.3

Looking at the data grouped by items, we have an average of the average ratings per item ≈ 3.08 which is quite close to the middle of the scale. However, not all items are close to the global average. We have a ratio of ≈ 0.49 items whose average ratings are close to the global average -> almost half of the items have an average close to the global average.

Q.3.1.4

Comparing averages predictions to the baseline:

Model	MAE
Global Average	0.9680
Users Averages	0.8501
Items Averages	0.8275
Baseline	0.7669

- We can see that the baseline model outperforms all of the other prediction models based on the averages.

- The highest MAE can be seen using the global average method and it is logical as the global average can not be generalized on all the users ratings patterns or the success of some items.
- Using the user ratings averages we improve our prediction but still suffer an important loss. Some users tend to have a disparity in their ratings therefore this method that introduces a bias into the predictions doe not work so well. In addition, some items are quite successful that they will have a good rating regardless of the user average. These items will maybe have a positive deviation from the user's average rating.
- Proceeding with the items rating averages as predictors, we discover an improvement from the users rating averages method. We can interpret this result as: an item's average is more logical to have as predictor than the user's average, as this will have the item's bias instead of the user's and thus will give more information about the item in question.
- The baseline model is the best predictor so far. Based on the items average deviations that incorporates the users biases, this method takes into account the success of each item compared to the different rating patterns of users. It is only logical that it would perform the best out of the 4 methods.

Q.3.1.5

Technical specifications:
- MacbookPro (13-inch, 2016, Four Thunderbolt 3 Ports)
- 2,9 GHz Intel Core i5 double cœur
- 8 Go 2133 MHz LPDDR3
- MacOS Big Sur 11.2.3
- sbt 1.4.7 (JetBrains s.r.o Java 11.0.6)

Execution times in microseconds				
Model	Minimum	Maximum	Average	Standard Deviation
Global Average	201905.09	884357.39	482547.5	253169.03
Users Averages	521201.12	3336583.46	1320295.57	971568.55
Items Averages	494299.17	862203.12	703579.93	100270.5
Baseline	2635555.89	4101214.77	3353147.37	437269.39

- Looking at the average execution times, we can see that the most expensive method is the baseline model. It follows from the fact that it uses the items average deviations and user averages.
- The fastest model is the global average method followed by the users averages and then the items averages model.
- We have a ratio of ≈ 6.95, that is the global average method is approximately 7 times faster than the baseline model. This result shows the consensus one has to make between performance times and correctness.

Q.4.1.1

Adding personal ratings on movies already seen in personal.csv, and fitting the model on the expanded data, some of the recommended movies might interest me to watch. However, these results show that these recommendations might come from the fact that these movies are not rated that much but have a full score of 5.0 on only one rating.

The recommendations are :

1. 814, "Great Day in Harlem", 5.0
2. 1122, "They Made Me a Criminal (1939)", 5.0
3. 1189, "Prefontaine (1997)", 5.0
4. 1201, "Marlene Dietrich: Shadow and Light (1996)", 5.0
5. 1293, "Star Kid (1997)", 5.0

Q.4.1.2

To favour more popular movies, it seemed logical to think about a mapping for "being famous" and a weight on the predictions. The solution is to consider a sigmoid on the count of number of ratings of movies with a "famous criteria" call it f_c. The new model would be:

$$p_{u,i} = w_i * (\bar{r}_{u,\bullet} + \bar{\hat{r}}_{\bullet,i} * scale((\bar{r}_{u,\bullet} + \bar{\hat{r}}_{\bullet,i}), \bar{r}_{u,\bullet}) - 1) + 1$$

with

$$w_i = sigmoid(count_i - f_c),\ sigmoid(x) = \frac{1}{1 + \exp(-x)}$$

and $count$ the number of ratings of item i.

The idea is to shift the sigmoid on the "famous criteria" f_c, that is the count of which we will consider a movie famous and associate this value to how important the prediction is. It follows that multiplying the baseline by this value gives a prediction that takes into account the popularity. This model will favor more famous movies to have almost the same prediction and penalizes the less popular movies. Some scaling by constants was necessary to keep the predictions in the range $[1, 5]$.

1.2 APPENDIX

1.3 SIGMOID FUNCTION:

FIGURE 1.1
Y=Sigmoid(X-100)