**Programming for Data Analytics – 2019/2020**

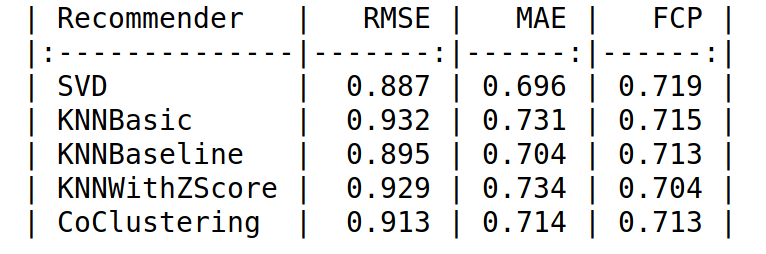
**Project Competition Report**

The goal of this report is to outline the procedures and techniques used to generate my submissions to the project competition of the “Programming for Data Analytics” class of 2019/2020.

In order to generate the predictions required by this project, I have used two different libraries, surprise and Graphlab, running then some experiments with different algorithms and systems available in the libraries. Given that the experiments based on the same library share most of the code, I have divided this report in two main sections, one per library. Every section outlines first the general procedure, comparing then the results of different experiments.

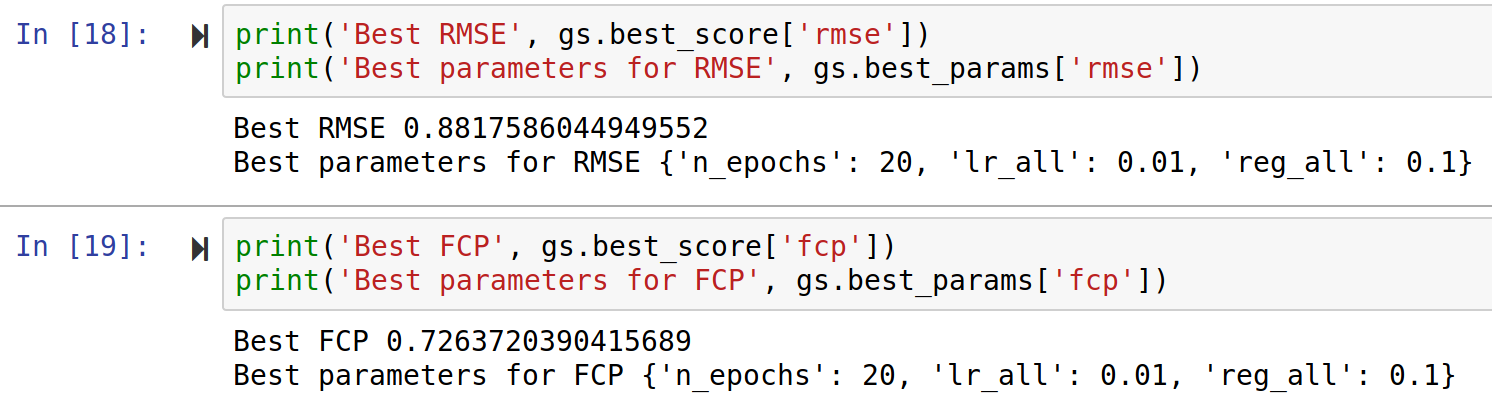
**The surprise library**

The first experiments I have decided to run were based on the surprise library, given that it has been previously used in class for different exercises. The two experiments based on this library used the SVD and SVD++ algorithms, since an in-class comparison on the competition’s training data set showed how they were the most suitable ones for our scenario, as shown also by the results summarized in the following table:



In both experiments, the first step is to read the input training, test and contents data sets, operation performed using the Pandas library. From the training data set, the full dense matrix is generating, pivoting the original data set on the userID and itemID columns. This operation is needed to understand which movies have not been seen by the different users, so that we can avoid predicting unuseful ratings.

We then extract, from the test data set, the list of users we are interested into for our predictions. Next, we generate the train set, using the reader object, with the rating scale set from 1 to 5, and loading and building the full train set, as shown in class.

We can then create the recommender system, based on the SVD or SVD++ algorithm, using the generated train set. For the SVD experiment, the most suitable parameters have been found applying the grid search procedure executed as part of the in-class quiz, with the following results:

Unfortunately, for the SVD++ I was not able to complete the grid search procedure due to some hardware limitations and for this reason I used the default parameters set by the library.

After creating the recommender object, the model is then trained using the generated train set.

Next, we loop through the list of test user and, for each of them, we predict the rating for all movies that he/she has not watched yet. Those movies are represented by a NA in the dense matrix rows, with the movie id corresponding to the column. The predicted ratings are then added in a dictionary having as key the corresponding movie id.