Miner user: lalvarez

Rank: 31

RMSE: 1,00

RECOMMENDER SYSTEM

Introduction

In this project I have developed a recommender system that predicts the user's ranking for a movie.

My model is based on **low-dimensional factor models**, with a matrix factorization based algorithm. The model, creates a rating matrix than contains the ratings from the train file. The unknown results are computed as the average of the known ratings of each user for all the movies. After applying SVD to the matrix, the initial matrix is decomposed into 3 matrix. The columns of the U matrix are called the left-singular vectors of A, and will represent the users, and the vectors of V are called the right-singular vectors of A, the movies for this problem. The middle matrix (diagonal) is reduced from 1223 to 12 values, this matrix contains the singular values, with meaningful information of the ratings. The rating matrix is replicated again but using the new reduced matrix of singular values.

Predictions

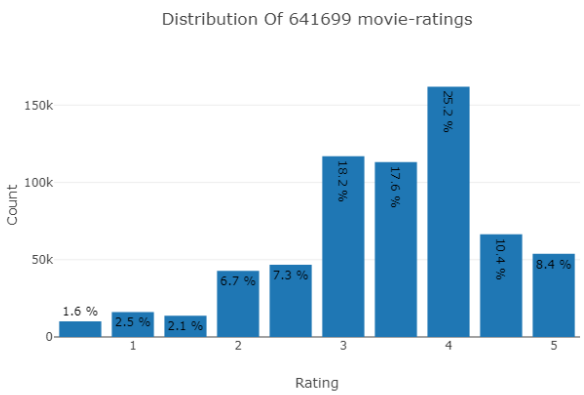
Initial matrix values:

The unknown values of the matrix are filled with the average grade for each user, this approach is not computationally expensive. The mean value is not powerful but can handle 600.000 instances of data. Due to the huge amount of records we have on both the training and test set this option was the most optimal for keeping the complexity of the algorithm O(n2).

Non-catalogued movies:

Movies or users that don't have an entry in the rating matrix are given a random value between 3 and 4. This is due to the distribution of the data:

The right graph shows that the higher percentage of ratings in the training data are between 3 and 4.

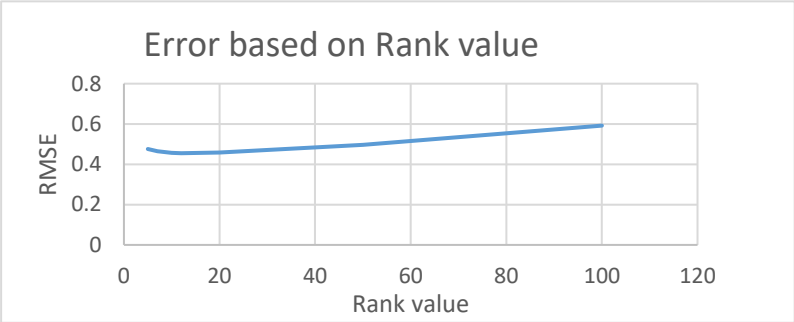


Model evaluation procedure

To analyze the performance of my classifiers, and set the rank value, number of singular values, I used the mean squared error rate, splitting my train data into train and test data and comparing the performance of the different rank values.

In the next graph, we can appreciate how the different values affect the performance of the classifier.

The error decreases to a minimum value when rank value is close to twelve and then the function increases towards infinity.



KNN failure

As I said before, the mean value for choosing the initial values of the unknown values it is not the most effective way to solve the problem, but it is efficient. This could be improved with a more powerful algorithm such as KNN to choose the initial values. KNN can find the nearest neighbor, the most similar user, and its rate to the test movie and give the same rating. The computational time for this was huge so I discarded this solution.