Name: Vigneshwaarar CR

CWID: A20392185

Collaborative Filtering

User-Based Collaborative Filtering:

- In Collaborative filtering, we don’t learn much about the data. We just need user, item and the rating.
- In user-based collaborative filtering, in training data, the user’s ratings for all the items are converted into a vector.
- Similarly, we form user vector in test dataset and for each user vector in test dataset, we find top K nearest neighbors in training dataset. We use cosine similarity as measure for finding the nearest neighbor.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2}\sqrt{\sum\limits_{i=1}^{n} B_i^2}}$$

- Once, we get the similarity measure and the top K nearest neighbor, we can use the following formula to calculate the predicted rating.

$$P_{u,i} = \frac{\sum_{\text{all similar items, N}}(s_{i,N} * R_{u,N})}{\sum_{\text{all similar items, N}}(|s_{i,N}|)}$$

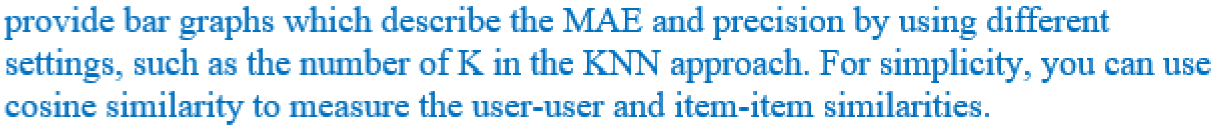
- Difference of Predicted rating for each item by the user to the actual rating given by the user for that item and we find mean of this error is our Mean Absolute Error value of our model.
- Now, we can evaluate the metric using Precision as well. We predict the top N recommendations, and the number of relevant items we recommend for the user based on his rating in test data, to the total number of ratings we recommend is precision.
- For each data in testing dataset, we find the distance to each of the training dataset. Let’s assume if K is 3, then top 3 training data with lesser distance value will be selected and most occurring value of the classification feature which we are predicting, will be the predicted value for the test data. Similarly, we predict for all the data in test dataset. As we already know the truth, we compare it with the predicted values and find accuracy. We repeat this process for different K values
- Similarly, we can build a model based on Item-Based collaborative filtering and here, instead of using User vector, we consider Item vector for our calculations.
- Now, we find K nearest neighbors using cosine similarity as shown above formula and we predict the ratings as shown above.
- Even here we can evaluate using metrics like Mean Absolute Error and precision by recommending the top N recommendations.

Execution steps:

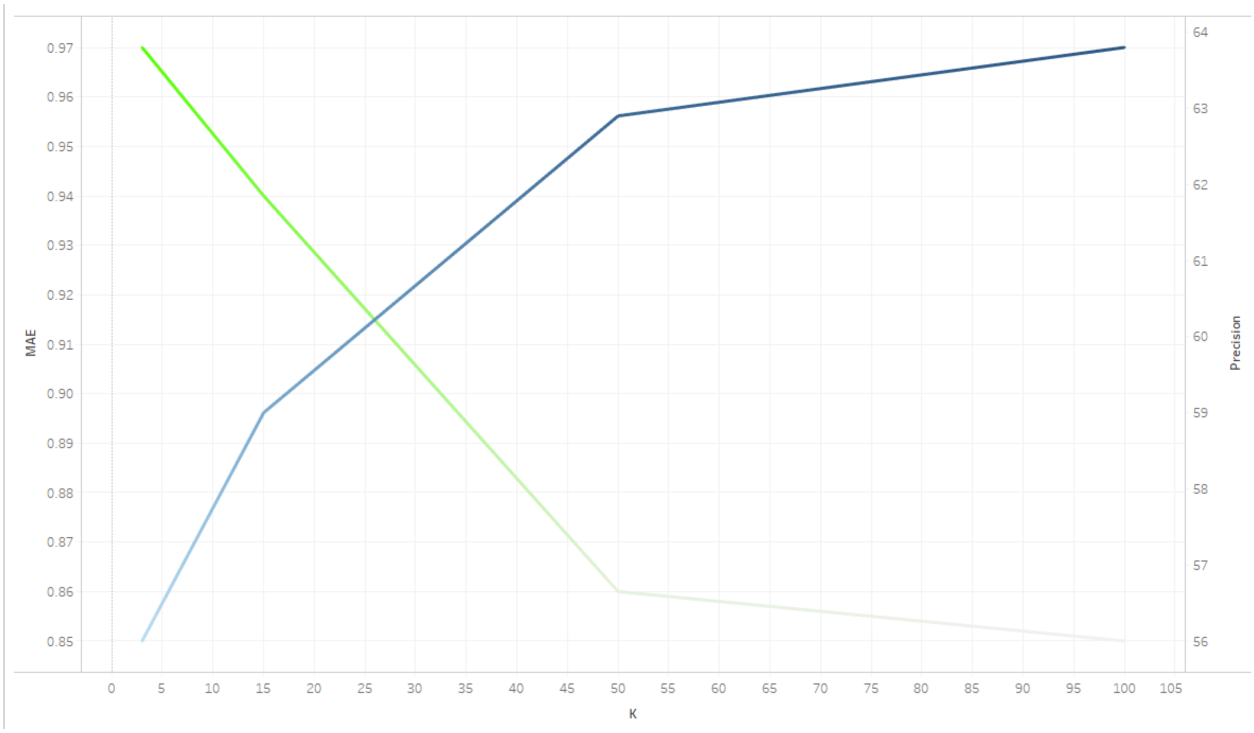
NOTE: PLACE THE INPUT FILES NAMED “train.txt” and “test.txt” IN THE SAME FOLDER AS THE .jar.

- Run the .jar file as following command format
  Java -CF.jar -u 0 -k 3 -n 1
- Enter the user based as 0 or Item based 1
- Enter k value next, which can be any integer
- Next inter n value, Evaluation metrics i.e., MAE as 0 and Precision as 1
- The predicted output file is saved to the folder path where the .jar exists as “out.txt”

MAE and Precision trend for different K-values:



provide bar graphs which describe the MAE and precision by using different settings, such as the number of K in the KNN approach. For simplicity, you can use cosine similarity to measure the user-user and item-item similarities.


MAE
Precision
K