CSE 572: Data Mining
HW-3

Name: Dev A Patel

ASU ID: 1229417087

Task 1

Q1: Run K-means clustering with Euclidean, Cosine and Jarcard similarity. Specify K= the number of categorical values of y (the number of classifications). Compare the SSEs of Euclidean-K-means, Cosine-K-means, Jarcard-K-means. Which method is better? (10 points)

	SSE
Euclidian	25323851408.26738
Cosine	687.7671018695032
Jaccard	3729.475436757144

Cosine K-means attains the lowest sum of squared error, so we can consider it better than others. Higher SSE can indicate that clusters are not compact.

Q2: Compare the accuracies of Euclidean-K-means Cosine-K-means, Jarcard-K-means. First, label each cluster using the majority vote label of the data points in that cluster. Later, compute the predictive accuracy of Euclidean-K-means, Cosine-K-means, Jarcard-K-means. Which metric is better? (10 points)

	Accuracy
Euclidian	60.21 %
Cosine	58.34 %
Jaccard	60.38 %

Comparing the accuracies euclidian and jaccard are close enough, but jaccard turns out to be the best. As we are choosing random centroids at start, it might affect the algorithm's accuracy. Also, only 100 iterations were run for this task, so if we run more iterations we may get to see the actaul scenario of accuray.

Q3: Set up the same stop criteria: “when there is no change in centroid position OR when the SSE value increases in the next iteration OR when the maximum preset value (e.g., 500, you can set the preset value by yourself) of iteration is complete”, for Euclidean-K-means, Cosine-Kmeans, Jarcard-K-means. Which method requires more iterations and times to converge? (10 points)

Stop criteria: when there is no change in centroid position

Here I have set a tolerance level of 0.0001. Euclidian K-means require more iterations but takes the least time. On the other hand, jaccard k-means require more time to converge.

	Iterations	Time (sec)
Euclidian	80	36.25
Cosine	70	51.39
Jaccard	74	71.97

Stop criteria: When the sse value increases in the next iteration

Here the max value of iterations is set to 500. Euclidian k-means go upto the maximum value of iterations and hence require the most amount of time.

	Iterations	Time (sec)
Euclidian	500	315.33
Cosine	17	29.04
Jaccard	1	0.97

Stop criteria: Max iterations = 500

For same 500 iterations, jaccard k-means takes most time to converge.

	Iterations	Time (sec)
Euclidian	500	242.48
Cosine	500	378.21
Jaccard	500	793.39

If we observe overall, jaccard would take more time in general if we ignore the increase in sse stop condition. This might be because of the operation to find the max and min point.

Q4: Compare the SSEs of Euclidean-K-means Cosine-K-means, Jarcard-K-means with respect to the following three terminating conditions: (10 points)
• when there is no change in centroid position
• when the SSE value increases in the next iteration
• when the maximum preset value (e.g., 100) of iteration is complete

Euclidian k-means has the highest sum of squared errors for all the three stop criterias.

Stop criteria: when there is no change in centroid position

Here I have set a tolerance level of 0.0001.

	SSE
Euclidian	25477833806.63
Cosine	682.05
Jaccard	3661.28

Stop criteria: When the sse value increases in the next iteration

	SSE
Euclidian	25407782883.54
Cosine	687.01
Jaccard	3994.51

Stop criteria: Max iterations = 500

For same 500 iterations, jaccard k-means takes most time to converge.

	SSE
Euclidian	25455066291.91
Cosine	684.26
Jaccard	3660.46

Q5: What are your summary observations or takeaways based on your algorithmic analysis? (5 points)

- Considering the no-change in centroids stop criteria, it might miss an optimal solution.
- When considering a maximum preset value stop criteria, it might sometime prove computationaly expensive as it takes more time, also there might be no improvement in accuracy.
- Euclidian K-means is generally faster, when time for a single iteration is compared.
- The tolerance level for stop criteria should not be significantly high or algorithm will not converge well.
- Even though euclidian k-means has high sse it get good accuracy, which indicated not well seperated clusters, but majority labels still dominate the clusters.

Task 2

3-c Compute the average MAE and RMSE of the Probabilistic Matrix Factorization (PMF), User based Collaborative Filtering, Item based Collaborative Filtering, under the 5-folds cross-validation (10 points)

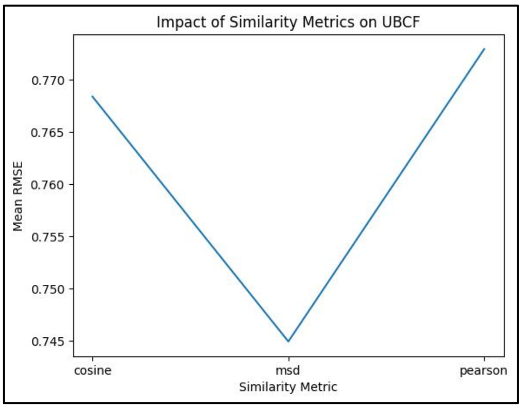
	PMF	UBCF	IBCF
MAE	0.6910	0.7449	0.7205
RMSE	0.8967	0.9688	0.9343

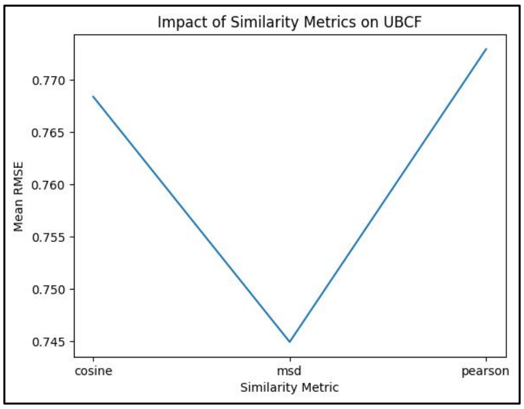
3-d Compare the average (mean) performances of User-based collaborative filtering, item-based collaborative filtering, PMF with respect to RMSE and MAE. Which ML model is the best in the movie rating data? (10 points)

From the above table, we can see that the best ML model in movie rating data is Probalistic Matrix Factorization (PMF) which attains the least MAE and RMSE. Here, I have not provided any specific method for ubcf/ibcf, so it will use the default method: mean squared error. So, the result can be different for cosine or pearson.

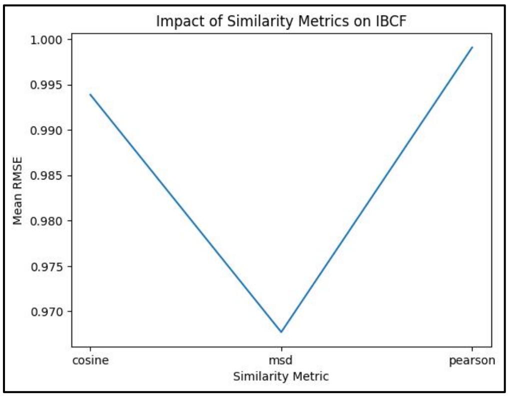
3-e Examine how the cosine, MSD (Mean Squared Difference), and Pearson similarities impact the performances of User based Collaborative Filtering and Item based Collaborative Filtering. Plot your results. Is the impact of the three metrics on User based Collaborative Filtering consistent with the impact of the three metrics on Item based Collaborative Filtering? (10 points)

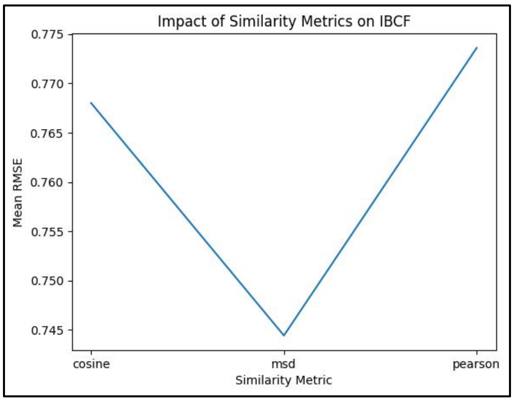
UBCF	cosine	msd	pearson
MAE	0.7684	0.7449	0.7729
RMSE	0.9943	0.9690	0.9984





IBCF	cosine	msd	pearson
MAE	0.7680	0.7444	0.7736
RMSE	0.9939	0.9677	0.9990

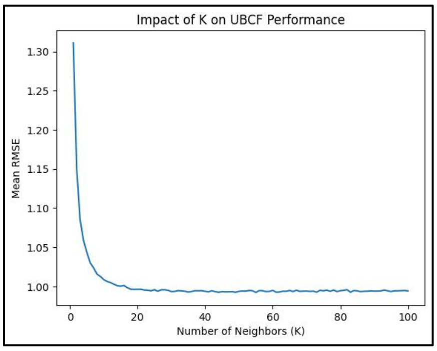


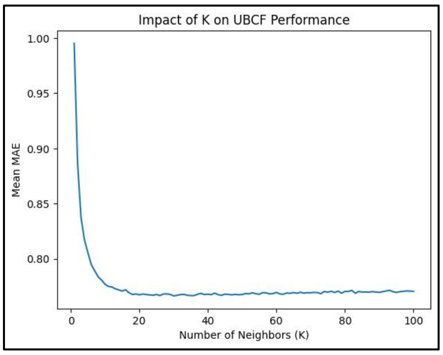


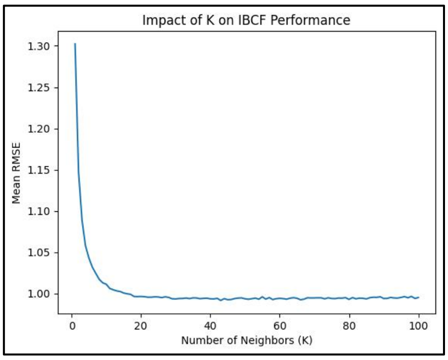
From the results, we observe that the impact of the three metrics on User based Collaborative Filtering is relatively consistent with the impact of the three metrics on Item based Collaborative Filtering, when comparing both the error metrics individually.

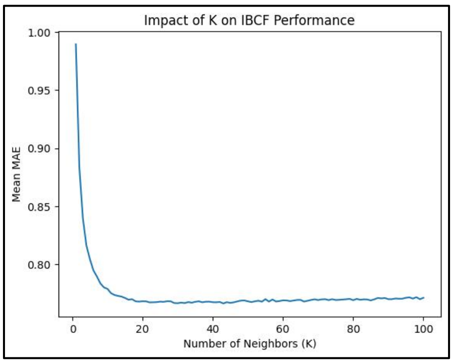
3-f Examine how the number of neighbors impacts the performances of User based Collaborative Filtering and Item based Collaborative Filtering? Plot your results. (10 points)

I have taken 100 values of k from 1 to 100 and calculated the rmse and mae for both UBCF and IBCF methods.









The error value decrease as k increase upto a certain limit but show a minor increase, after the optimal value of k.

3-g Identify the best number of neighbor (denoted by K) for User/Item based collaborative filtering in terms of RMSE. Is the best K of User based collaborative filtering the same with the best K of Item based collaborative filtering? (10 points)

For UBCF:
Minimum RMSE value: 0.9923432219991094
Corresponding k value: 55

Minimum MAE value: 0.7665228386236069
Corresponding k value: 30

For IBCF:
Minimum RMSE value: 0.9915285825388809
Corresponding k value: 43

Minimum MAE value: 0.7665276796265188
Corresponding k value: 43

The best value of k for UBCF and IBCF is different.

Github Links

Task 1:
https://github.com/dpate243/CSE572-Data-Mining/blob/main/hw3_kmeans.ipynb

Task 2:
https://github.com/dpate243/CSE572-Data-Mining/blob/main/hw3_recsys.ipynb