ASSIGNMENT 3: RECOMMENDER SYSTEMS – DATA SCIENCE

COSC2670/COSC2738
Ridge Tagala
S3934367

TASK 1 SUMMARY

- At the time I executed the data set, we get a MAE of 0.747 and a RMSE of 0.921. **Note that the values change every time the program is executed.

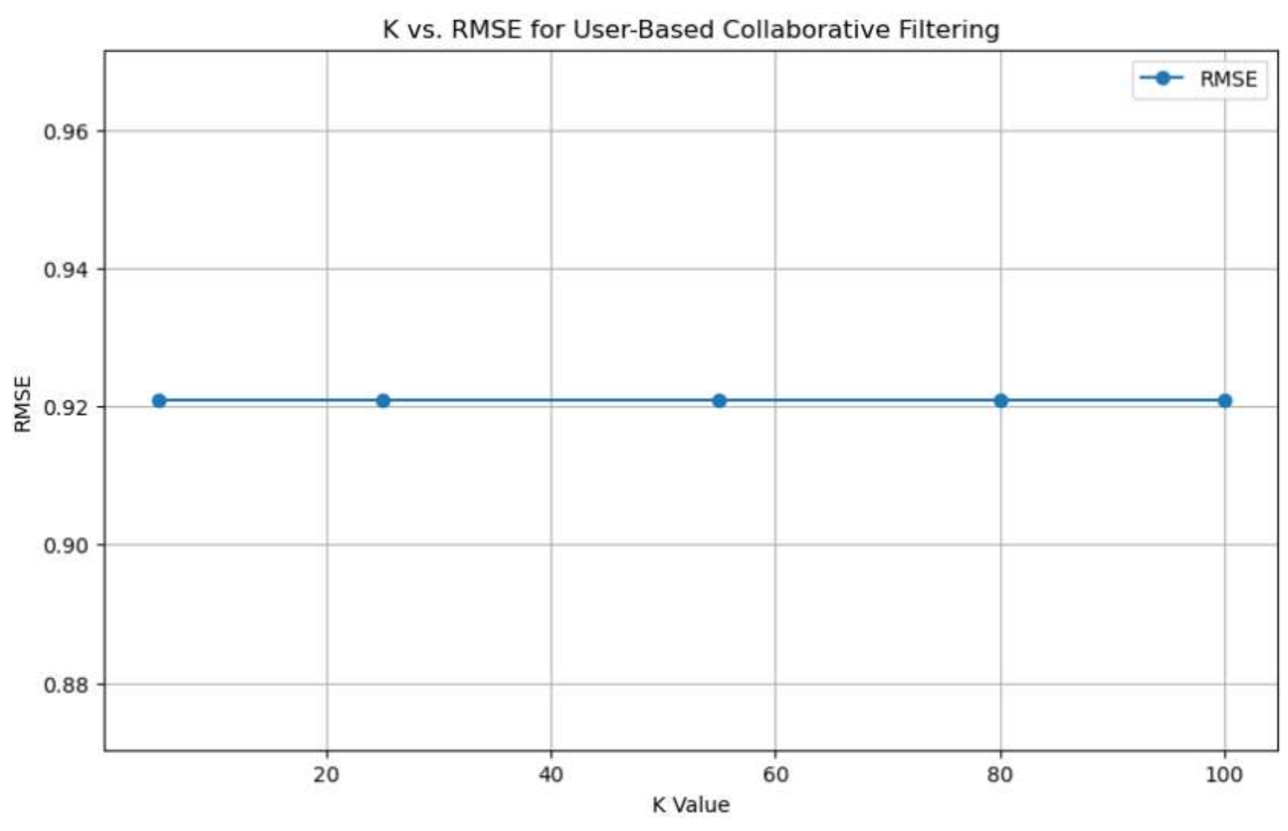
```
MAE on Tesing set (User-based): 0.7473684210869806
```

```
RMSE on Tesing set (User-based): 0.9209774406399865
```

- The MAE measures the average magnitude of errors in prediction. With a MAE of 0.747, the kNN models predicted ratings differ from the actual ratings by around 0.747 points on average.
- RMSE also measures the same thing – however, it gives more weight to larger errors due to the differences being squared.
- With the RMSE being slightly higher than the MAE, it indicates a low average prediction error. However, there are still occasional larger errors which are well captured by RMSE.
- In terms of accuracy, both metrics suggest that the kNN-based CF is doing a fitting job at predicting ratings, but there is some room for improvement. One improvement that can be done is the optimisation of the number of neighbours (k-value) in the algorithm.

RMSE VS K-VALUE

- We compare the RMSE and K-value and determine the impact of the K-value on RMSE. I have generated a graph to show this.
- As shown in the graph to the right, we can conclude that the RMSE has no direct impact on the k-Value, as the graph shows a consistent, linear line.
- This can be due to possible overfitting/underfitting, lack of dense data, or the insensitivity to the k-value.



PEARSON CORRELATION COEFFICIENT

- There are multiple types of similarity metrics that can be used to measure kNN collaborative filtering, however I have chosen the Pearson correlation coefficient for this specific task.
- The Pearson correlation coefficient measures the relationship linearly between 2 given variables, being adjustable by the mean rating. It considers both direction and magnitude which makes it useful when users have different rating scales.
- The Pearson Coefficient captures any relative rating patterns in our data, allowing it to determine if users rate movies similarly or below their own average.
- It also works well alongside user-based collaborative filtering, which helps identify users with a similar taste, its also less sensitive to individual users ratings – thus it ignores any variation between the users ratings and focuses on the shape of the users rating profile.

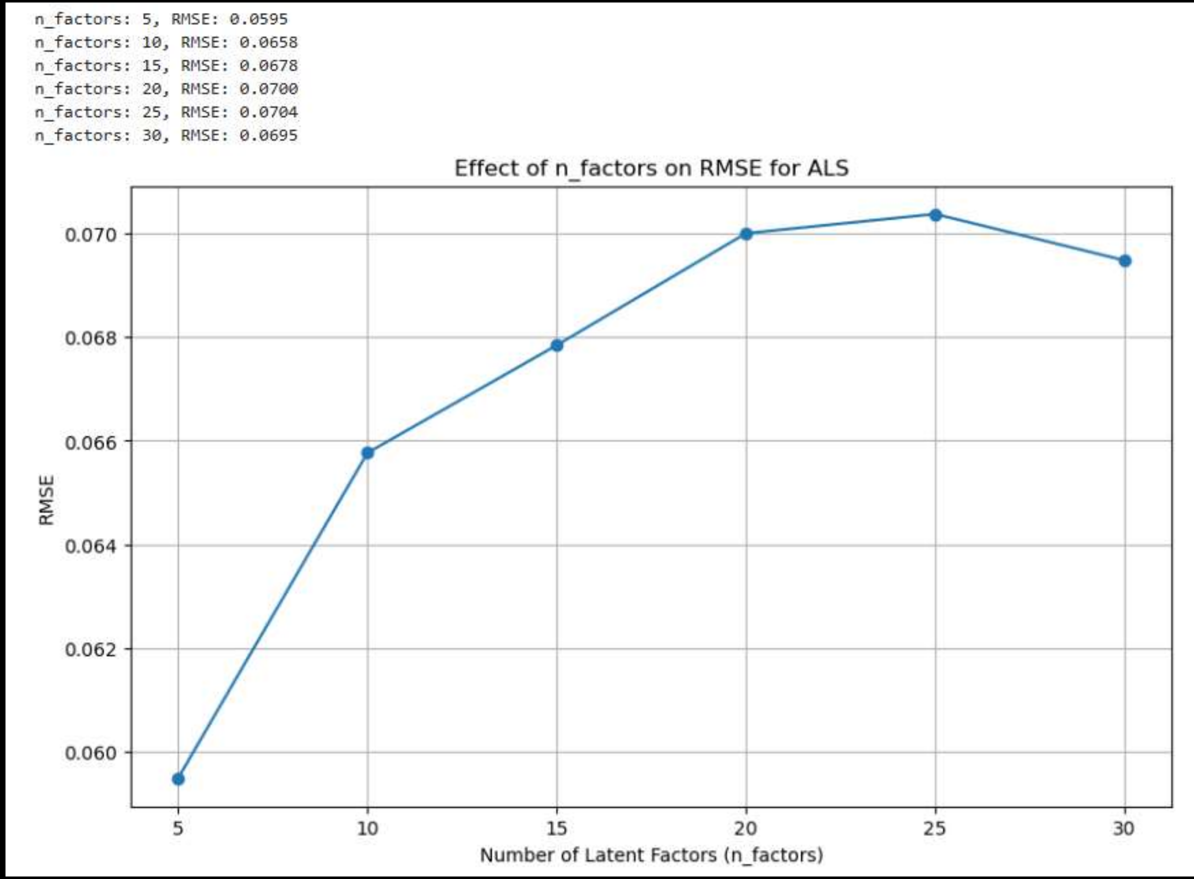
```
Weighted Pearson correlation between user 0 and user 0: 1.0000
Weighted Pearson correlation between user 0 and user 1: 0.1735
Weighted Pearson correlation between user 0 and user 2: 0.0935
Weighted Pearson correlation between user 0 and user 3: -0.1608
Weighted Pearson correlation between user 0 and user 4: 0.0994
Weighted Pearson correlation between user 1 and user 0: 0.1735
Weighted Pearson correlation between user 1 and user 1: 1.0000
Weighted Pearson correlation between user 1 and user 2: -0.0024
Weighted Pearson correlation between user 1 and user 3: 0.4001
Weighted Pearson correlation between user 1 and user 4: -0.0436
Weighted Pearson correlation between user 2 and user 0: 0.0935
Weighted Pearson correlation between user 2 and user 1: -0.0024
Weighted Pearson correlation between user 2 and user 2: 1.0000
Weighted Pearson correlation between user 2 and user 3: 0.2396
Weighted Pearson correlation between user 2 and user 4: -0.0240
Weighted Pearson correlation between user 3 and user 0: -0.1608
Weighted Pearson correlation between user 3 and user 1: 0.4001
Weighted Pearson correlation between user 3 and user 2: 0.2396
Weighted Pearson correlation between user 3 and user 3: 1.0000
Weighted Pearson correlation between user 3 and user 4: -0.0483
Weighted Pearson correlation between user 4 and user 0: 0.0994
Weighted Pearson correlation between user 4 and user 1: -0.0436
Weighted Pearson correlation between user 4 and user 2: -0.0240
Weighted Pearson correlation between user 4 and user 3: -0.0483
Weighted Pearson correlation between user 4 and user 4: 1.0000

]: array([[ 1.        ,  0.17353083,  0.09350981, -0.16076058,  0.09938514],
       [ 0.17353083,  1.        , -0.00242681,  0.40012858, -0.04361459],
       [ 0.09350981, -0.00242681,  1.        ,  0.23964733, -0.0240151 ],
       [-0.16076058,  0.40012858,  0.23964733,  1.        , -0.04828538],
       [ 0.09938514, -0.04361459, -0.0240151 , -0.04828538,  1.        ]])
```

TASK 2 SUMMARY

- The RMSE is at its lowest at 0.0595 when n_factors = 5, reaching its peak around n_factors = 25. This tells us that using fewer factors returns the best results, suggesting that the model doesn't need a high number of factors to detect patterns in the dataset
- The ALS Recommender system performs well when recommending movies to users – making accurate predictions for user ratings.
- The lower RMSE at fewer n_factors indicate that simpler models may have a better generalisation for this dataset.
- Compared to other recommender systems such as SVD, ALS thrives in implicit feedback datasets, such as movie ratings compared to SVD, and it also performs better in capturing relationships in larger data sets, helping provide personalised recommendations based on these patterns. It also handles overfitting and underfitting better, including explicit regularisation parameters and tuning flexibility to help balance over and underfitting in comparison to SVD, which is more prone to such errors.

AFFECT OF N-FACTORS ON RMSE FOR ALS

- I have chosen 6 common numbers from 1 – 30 to figure out the optimal value for n_factors.
- As you can see by the graph to the right, the RMSE is at its lowest when n_factors = 5. This indicates that the algorithm is more accurate at predicting the users ratings when there are less factors.
- As seen in the graph, the higher the value of n_factors, the higher the RMSE. This indicates that the value of n_factors affects the performance of the RMSE, thus affecting accuracy.



ALS AND RMSE

- As we can see from the image (right), we have fairly small RMSE, indicating that the ALS algorithm is fairly accurate in its predictions and is close to its actual values.

```
all 20 Iterations complete.

RMSE for ALS: 0.0536
```

- The recommendation model is deemed high quality due to the low RMSE, thus the ALS model is more effective at capturing users preferences compared to others, such as SVD and NMF.
- I have chosen ALS as the recommender system due to ALS being more preferable in this context to other factorisation techniques.
  - Our dataset is really sparse, thus another system such as SVD wouldn't perform as well due to it requiring a dense dataset.
  - ALS is also easily adaptable for implicit and explicit feedback - in this context, implicit feedback for movie ratings can be useful, as those who don't provide ratings can still have interaction data.
  - ALS also works better in larger data sets like this one, where we have multiple data files, and chosen data sets greater than 100.

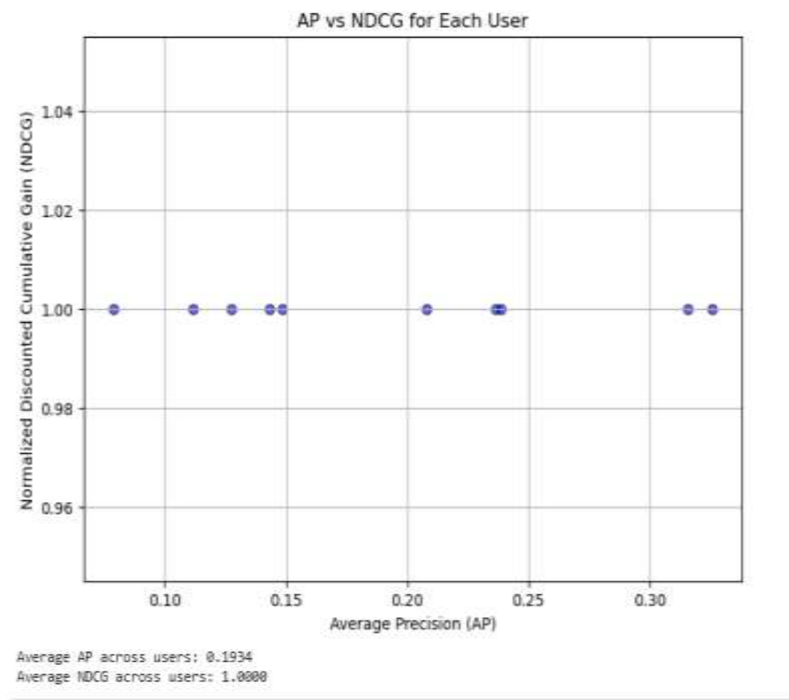
TASK 3: SUMMARY

- In this task, we observe the results in Task 1 and 2 and compare them with KNNCF and IMFR.
- We observe that IMFR achieves better performance for rating prediction accuracy (RMSE) compared to task 1, with a lower AP due to possible factors such as the sparsity of the data set, or the number of latent factors.
- However, IMFR scores a perfect NDCG score, indicating that IMFR has a slight edge, and indicating that the model has exceptional performance for the test data, having a high accuracy in ranking by relevance.
- KNNCF performs better in AP, indicating that it ranks better compared to IMFR. This indicates that the KNNCF is more precise and can be more interpretable, thus it achieves a greater immediate relevance in its recommendations.

IMFR AND KNNCF

- As seen with the image to the right, the kNNCF algorithm in task 3 has an avg AP of 0.690 and avg. NDCG of 0.861. This suggests a high level of accuracy, meaning that the system is returning relevant recommendations to the users preferences.
- For IMFR, we see a consistently high score for each user, thus the score of 1 suggests that the ranking of recommendations are ideal for the users, and implies that the relevant items to the users appear at the top of the list. The moderate AP score however, suggests that overall relevance across all recommended items may not be as high as desired.
- IMFR offers better performance due to its ability to better capture implied user preferences based on sparse data.

```
kNN-based Collaborative Filtering Results
Average AP: 0.6900000000000001
Average NDCG: 0.860789452165718
```



REFERENCES:

Internal References:

RMIT Canvas Links:

RMIT University. (n.d.-a). *Week 9 slides*. Canvas. https://rmit.instructure.com/courses/125162/files/41220806?module_item_id=6661186

RMIT University. (n.d.-b). *Week 9 practical activity*. Canvas. https://rmit.instructure.com/courses/125162/files/39724460?module_item_id=6451699

RMIT University. (n.d.-c). *Week 10 slides*. Canvas. https://rmit.instructure.com/courses/125162/files/41331874?module_item_id=6672004

RMIT University. (n.d.-d). *Week 10 practical activity*. Canvas. https://rmit.instructure.com/courses/125162/files/39724434?module_item_id=6451709

External References:

Pearson Correlation Coefficient: Scribbr. (n.d.). *Pearson correlation coefficient*. Retrieved from https://www.scribbr.com/statistics/pearson-correlation-coefficient/#:~:text=The%20Pearson%20correlation%20coefficient%20(r,the%20relationship%20between%20two%20variables.&text=When%20one%20variable%20changes%2C%20the,changes%20in%20the%20same%20direction.

Relation Between SVD and ALS: StackExchange. (2018). *What is the relation between SVD and ALS?* Retrieved from https://stats.stackexchange.com/questions/354355/what-is-the-relation-between-svd-and-als

Performance Comparison of SVD and ALS: Banik, N. S., Gupta, A., & Tiwari, M. (2021). *Performance comparison and analysis of SVD and ALS in recommendation system*. ResearchGate. Retrieved from https://www.researchgate.net/publication/379180704_Performance_comparison_and_analysis_of_SVD_and_ALS_in_recommendation_system#:~:text=Specifically%2C%20the%20SVD%20algorithm%20demonstrates,ALS%20by%20a%20considerable%20margin.

NDCG Metric: Evidently AI. (n.d.). *Normalized Discounted Cumulative Gain (NDCG)*. Retrieved from https://www.evidentlyai.com/ranking-metrics/ndcg-metric#:~:text=Normalized%20Discounted%20Cumulative%20Gain%20(NDCG)%20is%20a%20ranking%20quality%20metric,DCG%20representing%20a%20perfect%20ranking.

Matrix Factorization with Collaborative Filtering: Google Developers. (n.d.). *Matrix factorization for recommendation*. Retrieved from https://developers.google.com/machine-learning/recommendation/collaborative/matrix

RMSE Overview: Discovery. (n.d.). *Root mean square error (RMSE)*. Retrieved from https://discovery.cs.illinois.edu/guides/Statistics-with-Python/rmse/#:~:text=What%20is%20the%20RMSE%3F,and%20observed%20by%20a%20model.

RMSE and MAE Explanation: Kumar, S. (2019, August 20). *What are RMSE and MAE?* Medium. Retrieved from https://towardsdatascience.com/what-are-rmse-and-mae-e405ce230383

Matrix Factorization Technique: Malik, H. (2020, March 20). *Recommendation system: Matrix factorization*. Towards Data Science. Retrieved from https://towardsdatascience.com/recommendation-system-matrix-factorization-d61978660b4b

Pandas Factorize Documentation: pandas. (n.d.). *pandas.factorize*. Retrieved from https://pandas.pydata.org/docs/reference/api/pandas.factorize.html

Mean Absolute Error: C3 AI. (n.d.). *Mean absolute error (MAE)*. Retrieved from https://c3.ai/glossary/data-science/mean-absolute-error/

Implementing Matrix Factorization: Patel, R. (2020, March 23). *Implementing matrix factorization technique for recommender systems from scratch*. Medium. Retrieved from https://medium.com/@rebirth4vali/implementing-matrix-factorization-technique-for-recommender-systems-from-scratch-7828c9166d3c