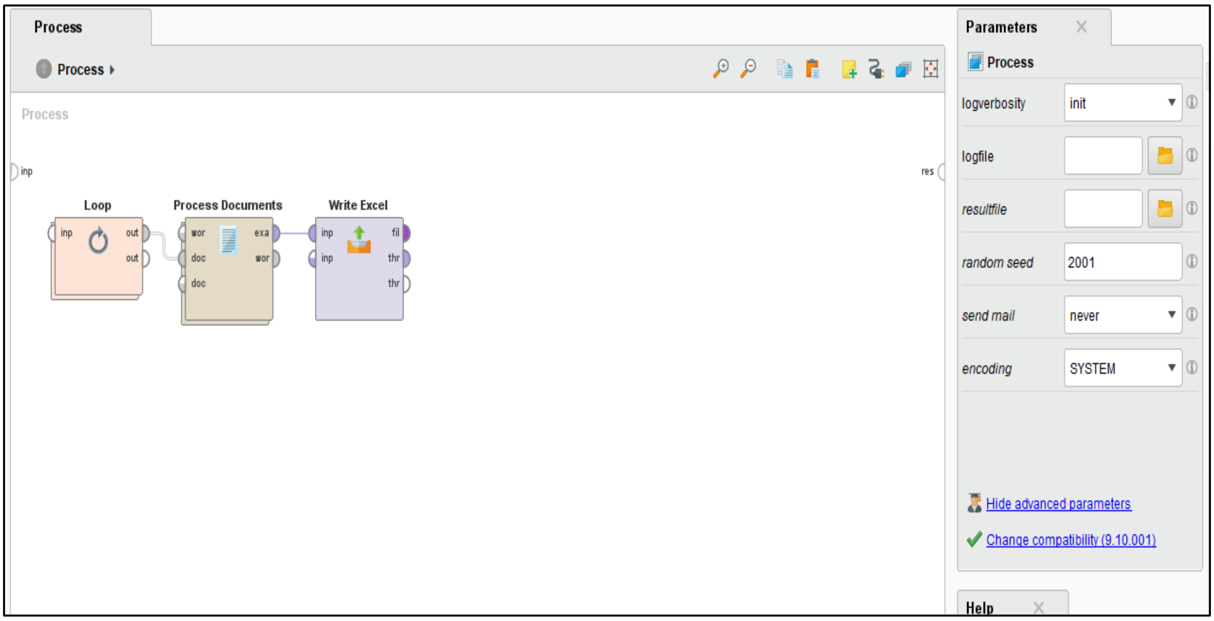
Information Retrieval Systems
Part 2 – Recommender Systems

Table of contents

Stage 1A: Crawl the web to retrieve a set of web pages

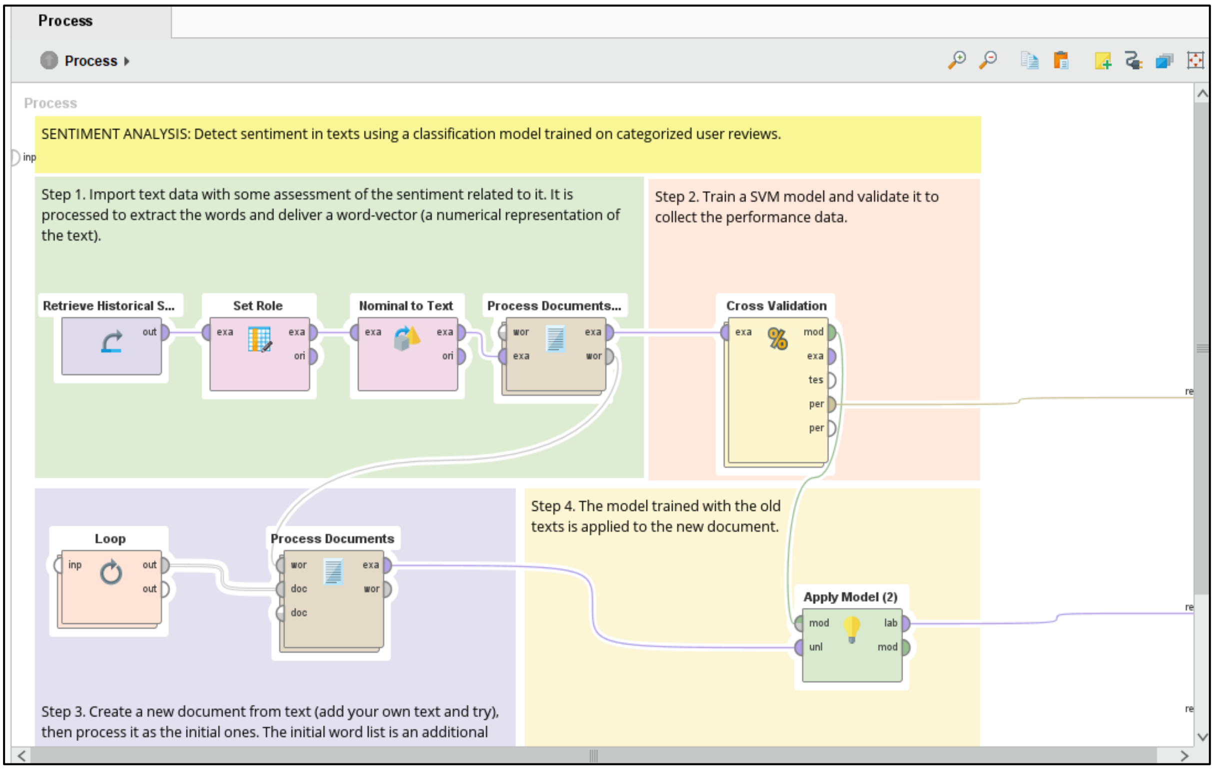
Process:



Please refer to Crawl.rmp file

Stage 1B: Sentiment Analysis

Process:

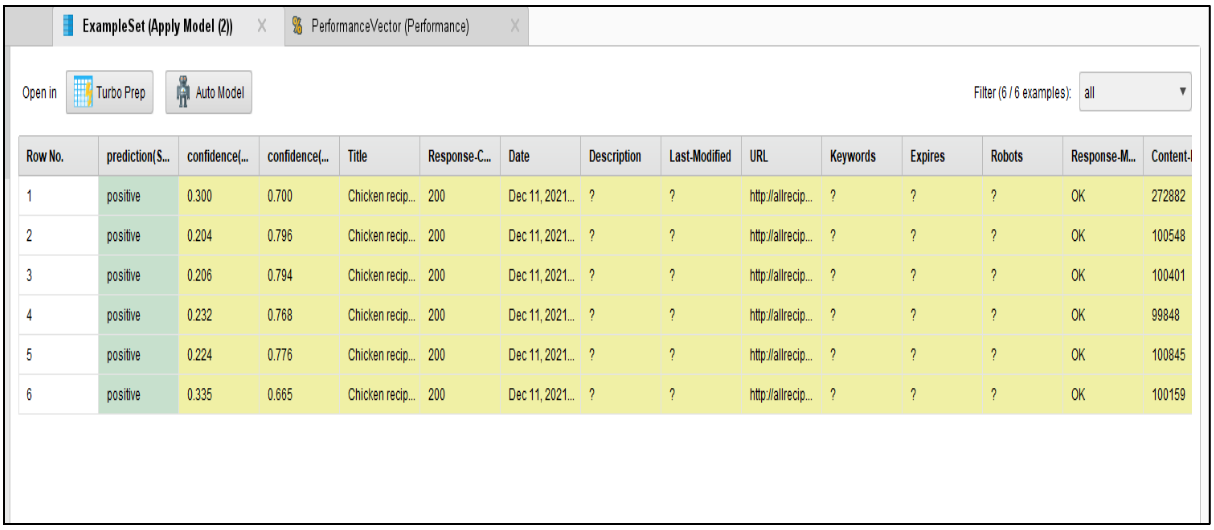


Please refer to Sentiment.rmp file

Question 1

"Sentiment analysis is a natural language processing (NLP) task in which a given text is classified into predefined classes (e.g., positive, neutral, and negative)" (Choi, Oh and Kim, 2020). It consists of several different tasks which are usually combined to get some knowledge about the opinions found in the text (Mejova, 2021).

Analysing results obtained in the above process:



Row No.	prediction(S...	confidence(...	confidence(...	Title	Response-C...	Date	Description	Last-Modified	URL	Keywords	Expires	Robots	Response-M...	Content-l
1	positive	0.300	0.700	Chicken recip...	200	Dec 11, 2021...	?	?	http://allrecip...	?	?	?	OK	272882
2	positive	0.204	0.796	Chicken recip...	200	Dec 11, 2021...	?	?	http://allrecip...	?	?	?	OK	100548
3	positive	0.206	0.794	Chicken recip...	200	Dec 11, 2021...	?	?	http://allrecip...	?	?	?	OK	100401
4	positive	0.232	0.768	Chicken recip...	200	Dec 11, 2021...	?	?	http://allrecip...	?	?	?	OK	99848
5	positive	0.224	0.776	Chicken recip...	200	Dec 11, 2021...	?	?	http://allrecip...	?	?	?	OK	100845
6	positive	0.335	0.665	Chicken recip...	200	Dec 11, 2021...	?	?	http://allrecip...	?	?	?	OK	100159

In the above figure, all documents in the term document matrix were returned having a positive polarity. We may say that all six documents have a general positive rating.

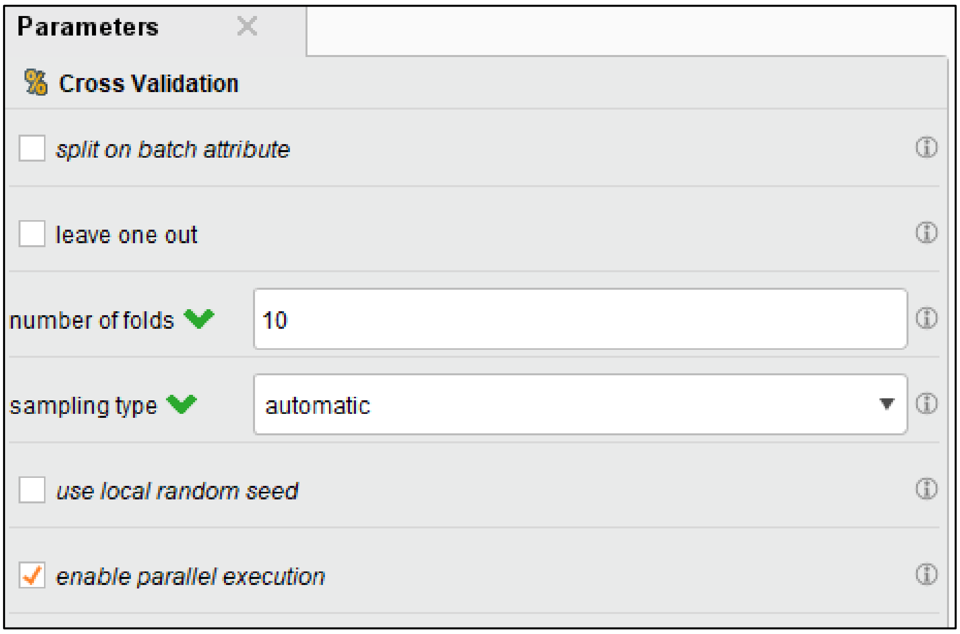


Figure 1

PerformanceVector

```
PerformanceVector:
accuracy: 64.00% +/- 9.37% (micro average: 64.00%)
ConfusionMatrix:
True:   negative        positive
negative:       44      21
positive:       51      84
```

With the above parameters in figure 1, the accuracy achieved for this model is at 64%. There were a total of 72 instances that were misclassified, with 21 negatives classified as positive and 51 positives classified as negatives. However, there were 44 negatives and 84 positives correctly classified which gives a total of 128 correct classifications.

By changing the parameters i.e., number of folds from 10 to 15 and, the sampling type to stratified sampling we can achieve a better accuracy, as seen below.

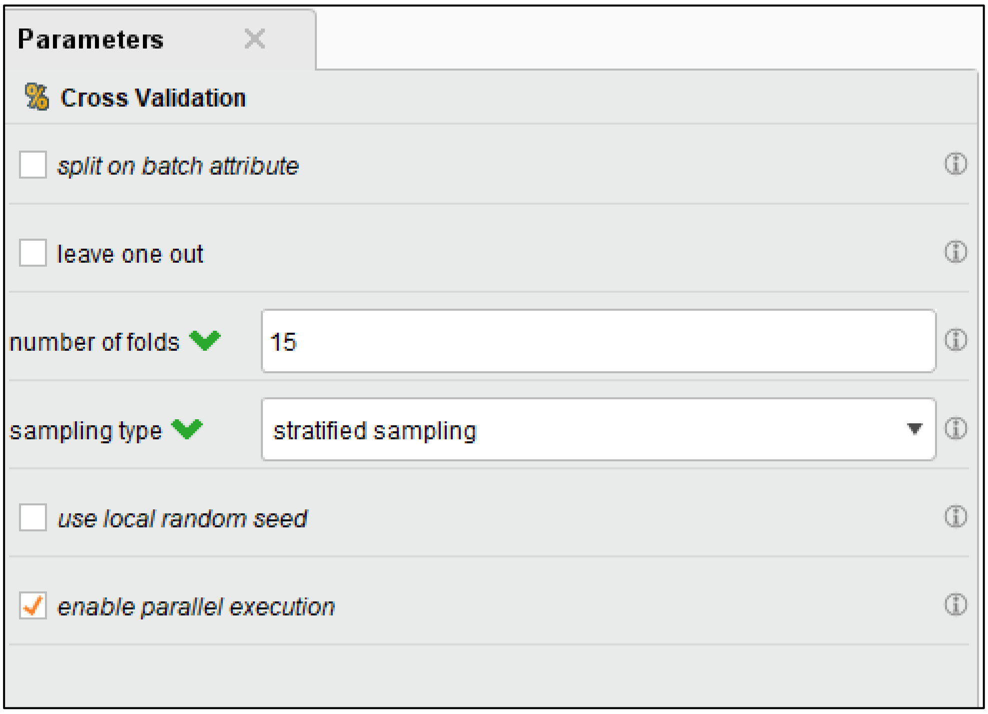


Figure 2

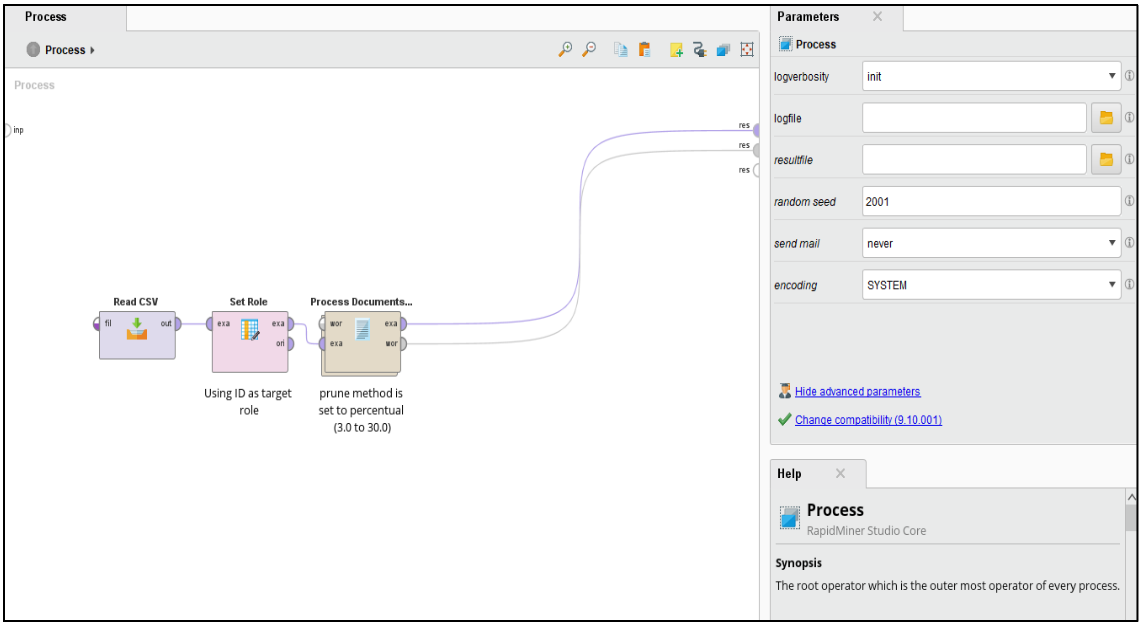
PerformanceVector

```
PerformanceVector:
accuracy: 67.07% +/- 13.58% (micro average: 67.00%)
ConfusionMatrix:
True:   negative        positive
negative:       44      15
positive:       51      90
```

The number of incorrect classification instances for class negative have reduced to 15 and consequently number of correctly classified instances for class positive have increased to 90, hence overall improvement in the performance.

Stage 2A: Processing Documents from Data

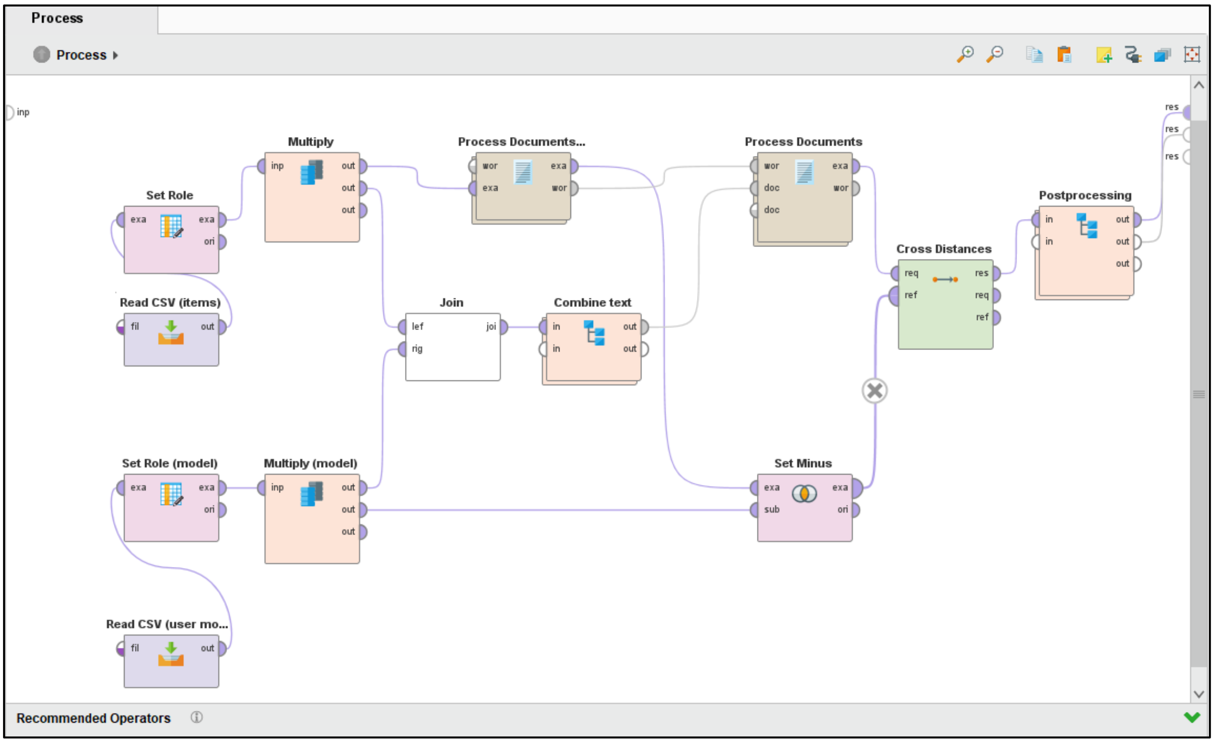
<u>Process</u>



Please refer to Processing.rmp file

Stage 2B: Building a Content Based Recommender

Process:



Please refer to contentRecommend.rmp file.

Question 2

Content-based recommenders

A representation of the product is created based on the properties of the product itself. For example, in a news article, the text contained in the article will be used to build the model. Standard IR techniques are used to create a vector space model of the news article and use that as a representation.

Advantages:

- We do not need any other data from other users
- No ratings of other users means sparsity is not a problem.
- There is no problem of a cold start because there are no opinions of others
- It has the ability to recommend users with unique or distinct taste because the model is based on the product itself.
- Not having to find similar users implies, the model is able to recommend new and unpopular items and overcome popularity bias.
- Because properties of product are used the model can provide explanations for the recommendation. For example, if a phone is being recommended we can give an

explanation of the features, like the camera picture quality, the memory etc. rather than basing the recommendation on just other people liking it.

Disadvantages:

- The data has to be in a structured format
- Not using other peoples opinion becomes a limitation here, because we are also unable to use quality judgements from other users for products that depend on it (Massie, 2021).

Collaborative Filtering

This approach uses ratings of other users as a representation and to make a recommendation. We find other users that are like us. This is a method of performing automatic filtering about user interests by collecting preferences from many other users (Massie, 2021).

Advantages:

- This approach does not require any knowledge about the features/attributes of the product or item (Massie, 2021).
- The model helps users discover new interests. The machine learning system may not know the user is interested in a given item, but it still recommends it because other similar users are interested in that item (Collaborative Filtering Advantages & Disadvantages | Recommendation Systems | Google Developers, 2021).

Disadvantages:

- Cold start problem, where either we have a new user who has not rated any item yet or a new item which has not been rated by any user. Such an item cannot be recommend because it has no ratings.
- Sparsity in the user/rating matrix. If there are multiple items to be recommended, the user/rating matrix is sparse and it is hard to find the users who have rated the same item and it becomes difficult to make recommendations.
- This model tends to recommend only popular items, hence there is a popularity bias (Massie, 2021).

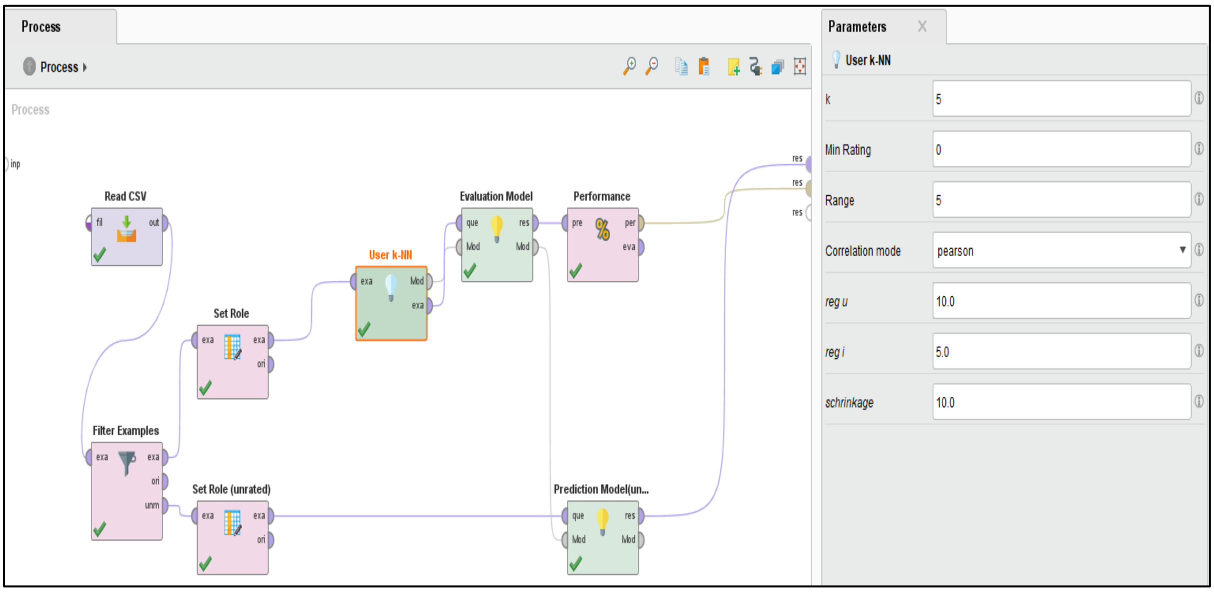
Conclusion

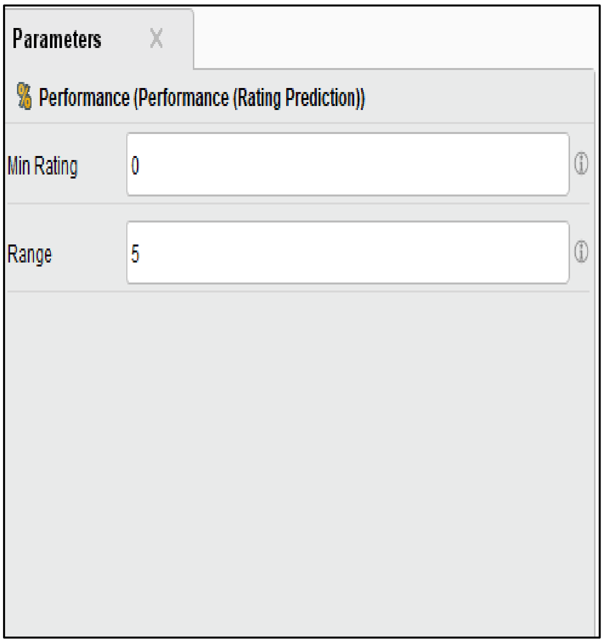
In the task above, we have about 9000 abstracts out of which only 2 abstracts were read. Thus, sparsity and cold start become an issue if considering collaborative filtering approach. The user/item rating matrix will be very sparse, considering the number of read documents and also that there was no rating provided for them. A new abstract newly submitted, would have been read and rated only few times and cannot be recognized easily from so many abstracts and recommended to the right student/s. On the other hand, in the content-based recommender systems, we use content analysis to represent all the abstracts and compute the similarity between abstracts and user profile, overcoming the new abstract problem (Bai, Wang, Lee, Yang, Kong & Xia, 2019).

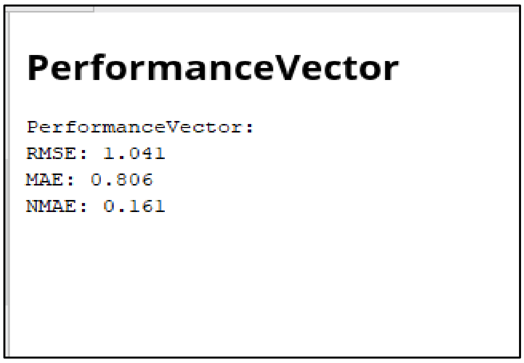
Hence, we may say that for academic abstract recommendation purposes, content based models are likely to give better recommendations because, this model uses the text contained in the abstracts itself to build the model as opposed to collaborative filtering which uses ratings of other users as a representation to make recommendations.

Stage 3: Building a Collaborative Filtering Recommender

In this task we build a model using collaborative filtering approach and measured the performance by Normalised Mean Average Error (NMAE) on the predictions made for known ratings.







Normalized Mean Average Error is calculated as

NMAE= MAE / Rating (max) – Rating (min), where MAE is the mean average error, calculated as

$MAE = (1 / n) * (\Sigma\ |y - \hat{y}|)$

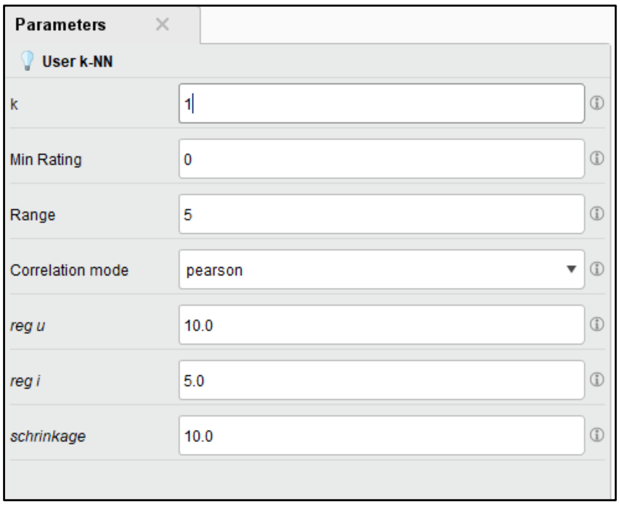
The error rates are as in the figure above.

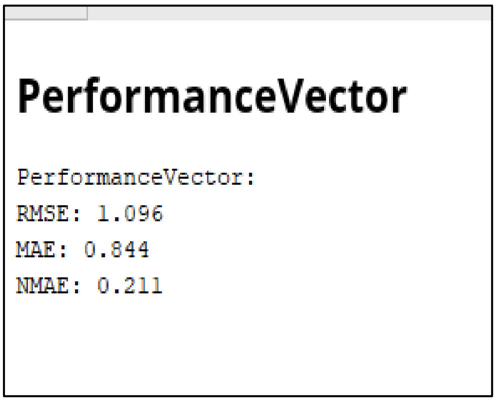
Question 3A

In this task we experiment with different parameters and designs to see if these changes can improve the performance.

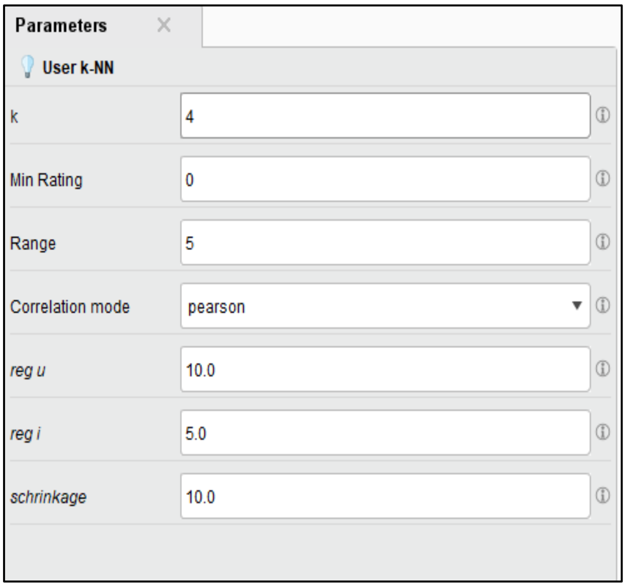
First we try different k values,

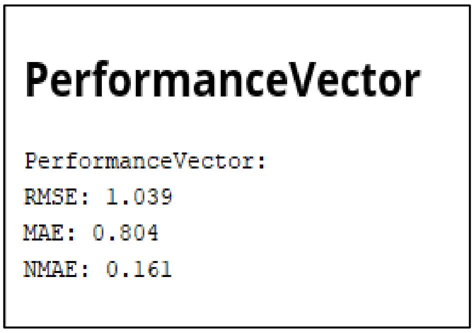
With k =1





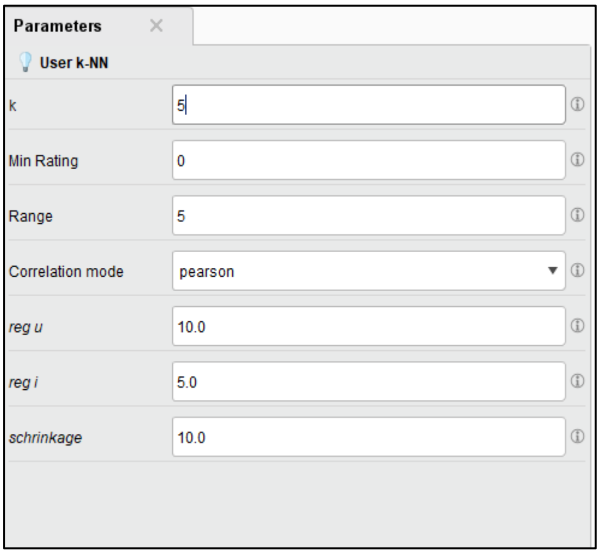
We can see the NMAE error rate increases when the value of k is decreases from 3 to 1 which is not useful. So, we now increase the value of k from 3 to 4 and see its impact on performance.

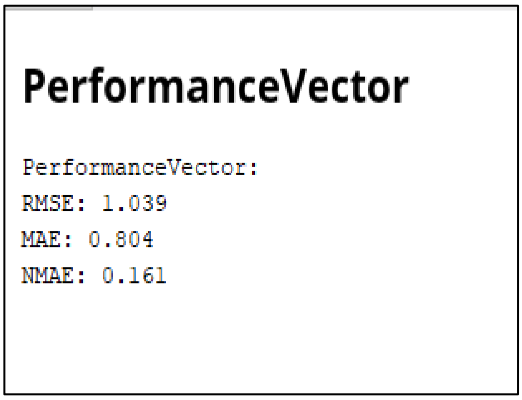




We notice that the NMAE error rate remains the same and there is no further improvement in performance.

With k = 5, the error rate still remains constant.





At k = 3 the error rate drops to 0.161 and remains constant after that until further trials with values up to 80. Hence, we may consider best k = 3.

We now experiment by changing the correlation coefficient from Pearson to Cosine for k =3, to see any changes or improvement in performance.

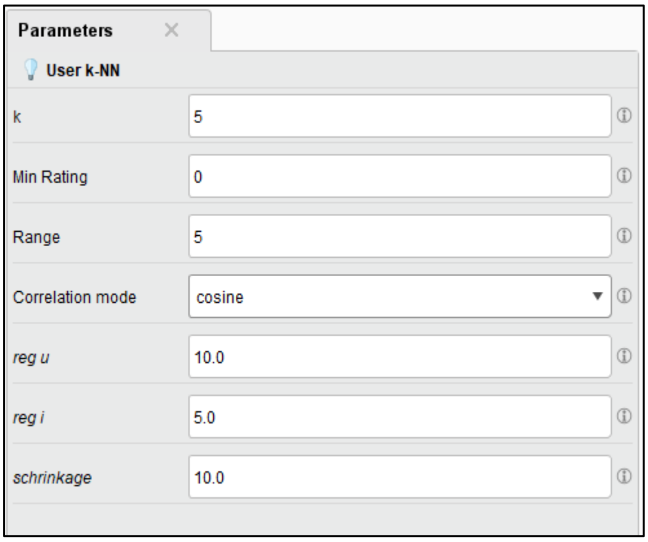
Pearson coefficient is measure using the formula below.

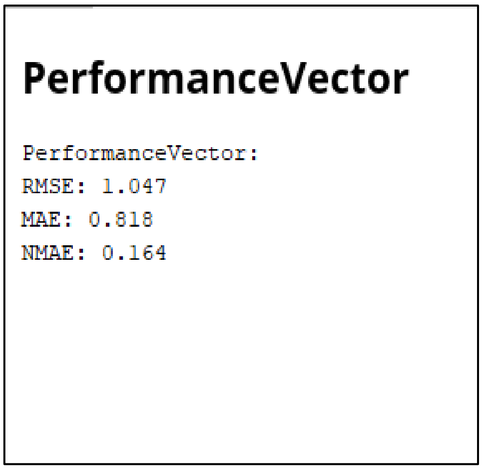
$$w_p(a,i) = \frac{\sum_j (x_{aj} - n_a)(x_{ij} - n_i)}{\sqrt{\sum_j (x_{aj} - n_a)^2 \sum_j (x_{ij} - n_i)^2}}$$

And Cosine is measured using the formula below.

$$w_c(a,i) = \frac{\sum_{j=1}^{n} x_{aj} x_{ij}}{\sqrt{\sum_{j=1}^{n} x_{aj}^{\ 2} \sum_{j=1}^{n} x_{ij}^{\ 2}}}$$

The NMAE error rate slightly increases with the Cosine Coefficient.



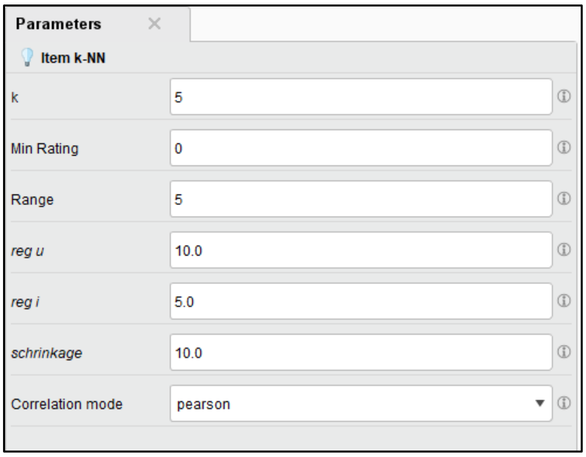


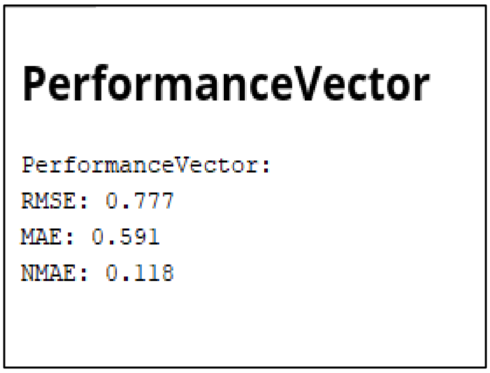
Hence, we may say that Pearson Coefficient Correlation works better for this model.

Now, we experiment with different operators to see if we can achieve an improved performance.

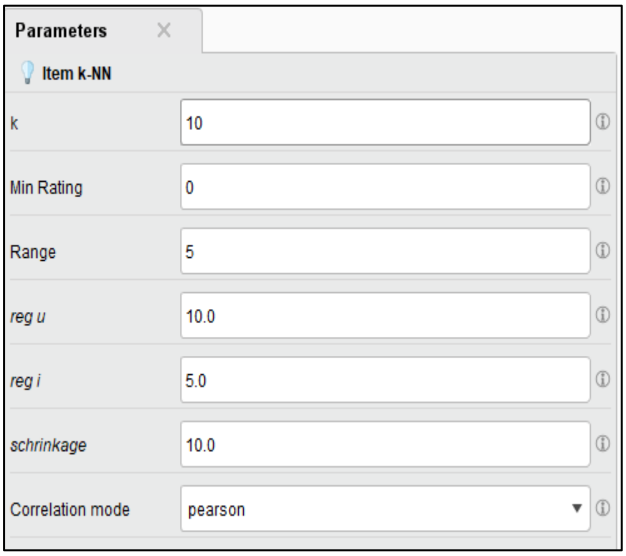
Using Item k-NN with different k values to see if we can achieve an improved performance.

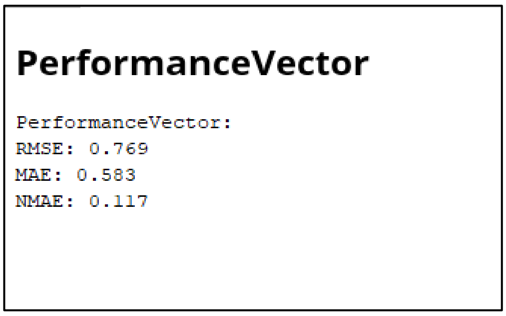
With k- 5 the NMAE value is 0.118.



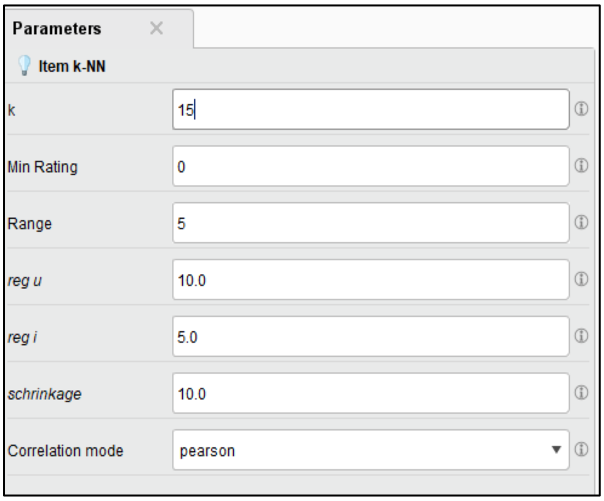


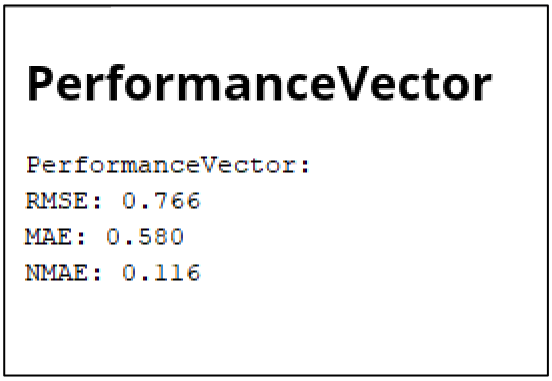
With k=10





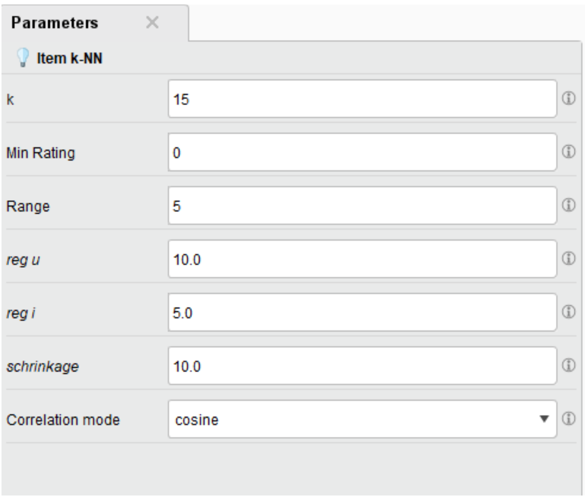
With k =15

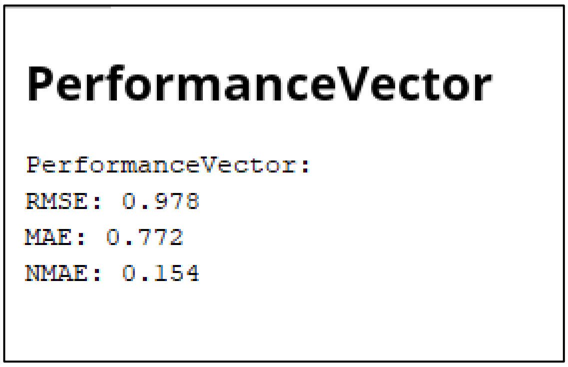




From the results achieved above, we can say that with increase in the k value the error rate decreases.

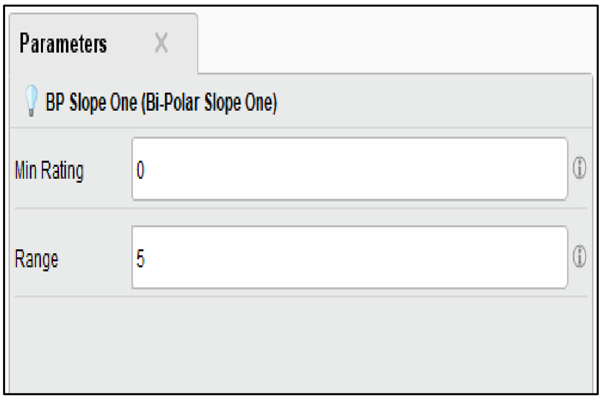
Changing the correlation coefficient from Pearson to Cosine and seeing the effect on the error rate.

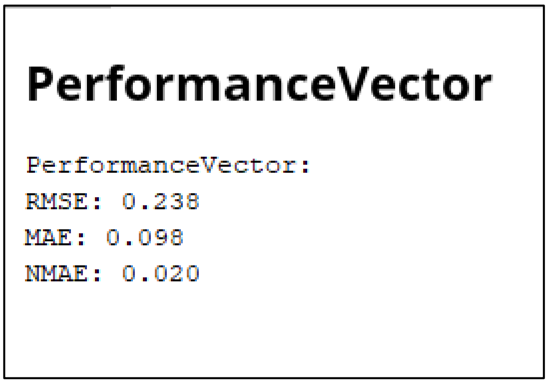




We can see that at k=15 with Cosine correlation Coefficient the NMAE error rate is 0.154, and with Pearson correlation coefficient NMAE is 0.116. Hence, we may say that with Item k-NN operator at k=15 and the correlation coefficient as Pearson Coefficient we achieve a lower error rate, hence an improved performance.

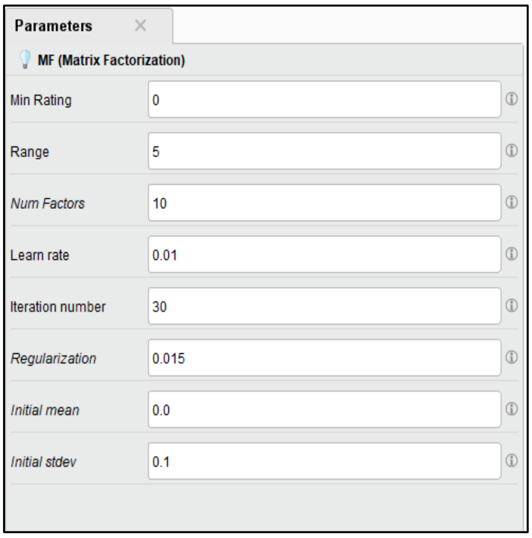
The operator is now changed to Bi-Polar Slope One and the performance vector is as below.

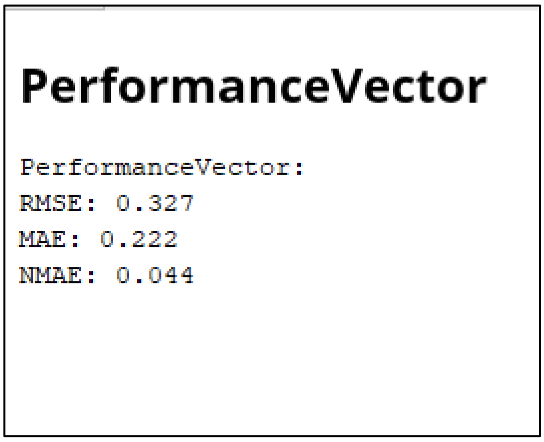




NMAE value is 0.020.

The operator is now changed to Matrix Factorization and the performance vector is as below.





NMAE value is 0.044

Observation

For k = 15 and the correlation coefficient as Pearson coefficient, using different operators, we can see the results below in the table.

Operators	**NMAE**
User k-NN, k=15, Pearson correlation coefficient	0.161
Item k-NN,k=15, Pearson correlation coefficient	0.116
BP Slope One	0.02
Matrix factorization	0.044

From the results above for this model in the above task, Bi-Polar Slope One operator gives the lowest NMAE value of 0.02. We may conclude that it gives a better performance for our model.

Question 3B

There is an assumption in most of the recommendation systems that the number of users are larger than the number of items. Under this assumption the recommendation algorithms can run effectively. However, this is not true and even the most popular items may have very few ratings. This makes the user/rating matrix very sparse (Bai, Wang, Lee, Yang, Kong & Xia, 2019).

Here, with the abstracts provided in the coursework we have about 9000 abstracts out of which only 2 abstracts were read. The user/item rating matrix will be very sparse, considering the number of read documents and also that there was no rating provided for them. A new abstract newly submitted, would have been read and rated only few times and cannot be recognized easily from so many abstracts and recommended to the right user.

Sparsity seems to be a problem in collaborative filtering approach because it uses usage statistics. If there are not enough statistics/ratings the system cannot make recommendations. One of the ways to address sparsity is to deal with the missing values/ratings.

In order to improve accuracy of the recommendation results, some of the paper recommender systems combine two or more recommendation techniques to make recommendations. Such an approach or method is called the Hybrid method. One of the advantages of this method is that it can use combinations of different recommendation techniques and the information from many sources (Bai, Wang, Lee, Yang, Kong & Xia, 2019).

One such hybrid method/system is a combination of content-based and collaborative filtering methods. Both these methods have their own benefits and limitations. The hybrid method uses both content based techniques and collaborative filtering techniques. The content based techniques help in building a user's profile by capturing any previous interests of the user, on the other hand collaborative filtering techniques help in identifying the user's ratings. By combining the two methods, the system is able to perform much better than the traditional recommendation systems. Some studies have used this combination with different forms to make better paper recommendations, which also addresses the problem of sparsity (Bai, Wang, Lee, Yang, Kong & Xia, 2019).

References

- Choi, G., Oh, S. and Kim, H., 2020. Improving Document-Level Sentiment Classification Using Importance of Sentences. *Entropy*, 22(12), p.1336.
- Mejova, Y., 2021 Sentiment Analysis: An Overview.
- Bai, X., Wang, M., Lee, I., Yang, Z., Kong, X., Xia, F. 2019. Scientific Paper Recommendation: A Survey. IEEE Access. PP. 1-1. 10.1109/ACCESS.2018.2890388.
- Massie, S., 2021. *Content-based Approach*.
- Massie, S., 2021. *Collaborative Filtering*.
- Google Developers. 2021. *Collaborative Filtering Advantages & Disadvantages | Recommendation Systems | Google Developers*. [online] Available at: <https://developers.google.com/machine-learning/recommendation/collaborative/summary> [Accessed 13 December 2021].