CAPSTONE PROJECT
MUSIC RECOMMENDATION SYSTEMS
MILESTONE 2 REPORT

- **Refined Insights -** What are the most meaningful insights from the data relevant to the problem?
    1. For performance evaluation of these models, precision@k and recall@k are used. Using these two metrics, the F_1 score is calculated for each working model.
    2. As we are trying to find the best model for this task, the F_1 scores for the models we assessed with the precision_recall_at_k function are
        - Baseline User-User Similarity Based Collaborative Filtering - 0.504
        - Optimized User-User Similarity Based Collaborative Filtering - 0.498
        - Baseline Item-Item Similarity Based Collaborative Filtering - 0.397
        - Optimized Item-Item Similarity Based Collaborative Filtering - 0.498
        - Baseline Model-Based Collaborative Filtering (Matrix Factorization) - 0.498
        - Optimized Model-Based Collaborative Filtering – Matrix Factorization - 0.502
        - Baseline Cluster-Based Recommendation System - 0.472
        - Optimized Cluster-Based Recommendation System - 0.462
    3. There is a lot of room for improvement of the hyperparameters using gridsearch. As suggested earlier, with grid search, we could train the model using different combinations from a wider range of these hyperparameters and select the combination that produces the best performance on the validation set.
- **Comparison of various techniques and their relative performance -**
    1. How do different techniques perform?
        - The F1 score is a metric that is used to evaluate the performance of a classification model. It is the harmonic mean of the model's precision and recall. A high F1 score indicates that the model has a good balance of precision and recall, while a low F1 score indicates that the model may be lacking in one of these areas. As the F_1 scores indicate, all the models are average at best. None of the models stand out.
    2. Which one is performing relatively better?
        - The best one is the baseline user-user similarity-based collaborative filtering model with an F_1 score of 0.504. Interestingly, this result aligns with Spotify's model design.
    3. Is there scope to improve the performance further?
        - While the optimized user-user similarity-based collaborative filtering model didn't yield a higher F_1 score, we can tune the hyperparameters with a computationally more expensive grid search with a wider parameter range. This will surely yield a better F_1 score and a better model.
- **Proposal for the final solution design -**

  What model do you propose to be adopted? And why is this the best solution to adopt?

    - The user-user similarity-based collaborative filtering model should be adopted as it has the best F_1 score. For hyperparameter tuning, one approach is, of course, testing a wider range of parameters with cross-validation. Another approach is to use domain knowledge or prior experience with similar problems to guide the selection of parameters to test. For example, if you have experience with KNN and know that certain parameter values tend to work well in similar scenarios, you can use this knowledge to inform your choice of parameters to test. In case this approach doesn't yield a satisfying model, we can try hybrid models, combining the user-user similarity-based collaborative filtering with optimized Model-Based Collaborative Filtering, which yielded the second-best F_1 score.

Questions –

1. Which metric should be used for this problem to compare different models?

Ans. In the present case, precision and recall both need to be optimized as the service provider would like to minimize both the losses discussed above. Hence, the correct performance measure is the F_1 score, as it is the harmonic mean of the two measures.

2. In the function precision_recall_at_k above, the threshold value used is 1.5. How are precision and recall affected by changing the threshold? What is the intuition behind using the threshold value of 1.5?

Ans. Relevant item: An item (product in this case) that is rated higher than the threshold rating (here 1.5) is relevant. If the actual rating is below the threshold, then it is a non-relevant item.

Recommended item: An item that's predicted rating is higher than the threshold (here 1.5) is a recommended item. If the predicted rating is below the threshold, then that product will not be recommended to the user.

Recall: It is the fraction of relevant items that are recommended to the user.

Precision: It is the fraction of recommended items that are relevant.

The intuition behind using the threshold value of 1.5 is that we want the recommended and relevant sets to be useful while also not being too diluted or restricted.

3. How changing the test size would change the results and outputs?

The size of the testing set can affect the F1 score in a few different ways. For example, a larger testing set can provide a more accurate representation of the performance of a model on unseen data, which can lead to a more reliable F1 score. Additionally, a larger testing set can provide a more fine-grained breakdown of the performance of a model on different classes or categories, which can be useful for identifying potential problems or areas for improvement.

However, it's important to note that simply increasing the size of the testing set without careful consideration of the data being used can lead to overfitting and other problems. It's generally best to use a testing set that is large enough to provide reliable results but not so large that it becomes unwieldy or difficult to work with.

4. Along with making predictions on listened and unknown songs, can we get 5 nearest neighbours (most similar) to a certain song?

We can find out similar users to a given user or its nearest neighbours based on this KNNBasic algorithm. Below, we are finding the 5 most similar users to the first user in the list with internal id 0, based on the msd distance metric.

5. In the above function, to correct the predicted play_count a quantity 1/np.sqrt(n) is subtracted. What is the intuition behind it? Is it also possible to add this quantity instead of subtracting?

Higher play freq means more accurate predictions. This has been proven empirically.

In the above-corrected rating formula, we can add the quantity 1/np.sqrt(n) instead of subtracting it to get more optimistic predictions. But here, by subtracting this quantity, we are being more stringent by demanding our predicted ratings to be supported by more data.

6. (Optimized Item-Item Similarity Based Collaborative Filtering) How do the parameters affect the performance of the model? Can we improve the performance of the model further? Check the list of hyperparameters here. (item-item)

- k (int) – The (max) number of neighbours to take into account for aggregation (see this note). The default is 40.
- min_k (int) – The minimum number of neighbours to take into account for aggregation. If there are not enough neighbours, the prediction is set to the global mean of all ratings. The default is 1.
- sim_options (dict) – A dictionary of options for the similarity measure. See Similarity measure configuration for accepted options.
- Verbose (bool) – Whether to print trace messages of bias estimation, similarity, etc. The default is True.

With grid search, we could train the model using different combinations from a wider range of these hyperparameters and select the combination that produces the best performance on the validation set.

7. (Optimized Model-Based Collaborative Filtering) How do the parameters affect the performance of the model? Can we improve the performance of the model further? Check the available hyperparameters here.

- n_epochs – The number of iterations of the scholastic gradient descent procedure. The default is 20.
- lr_all – The learning rate for all parameters. The default is 0.005.
- reg_all – The regularization term for all parameters. The default is 0.02.

With grid search, we could train the model using different combinations from a wider range of these hyperparameters and select the combination that produces the best performance on the validation set.

8. (Optimized Cluster-Based Recommendation System)How do the parameters affect the performance of the model? Can we improve the performance of the model further? Check the available hyperparameters here.

- n_cltr_u (int) – Number of user clusters. The default is 3.
- n_cltr_i (int) – Number of item clusters. The default is 3.
- n_epochs (int) – Number of iterations of the optimization loop. The default is 20.

With grid search, we could train the model using different combinations from a wider range of these hyperparameters and select the combination that produces the best performance on the validation set.

9. So far, we have only used the play_count of songs to find recommendations, but we have other information/features on songs as well. Can we take those song features into account?

Yes, we can do this by computing similarities between the texts of the other features.