# Recommendation Systems

## MIT-IDSS

Date – 24/11/2024

## Contents / Agenda

- Business Problem and Data Overview
- Exploratory Data Analysis
- Rank Based Model

- User-User Similarity-based Model
- Item-Item Similarity-based Model
- Matrix Factorization based Model
- Conclusion and Recommendations

## Business Problem and Data Overview

- Today, information is growing exponentially with volume, velocity and variety throughout the globe. This has lead to information overload, and too many choices for the consumer of any business. It represents a real dilemma for these consumers and they often turn to denial. Recommender Systems are one of the best tools that help recommending products to consumers while they are browsing online. Providing personalized recommendations which is most relevant for the user is what's most likely to keep them engaged and help business.

- E-commerce websites like Amazon, Walmart, Target and Etsy use different recommendation models to provide personalized suggestions to different users. These companies spend millions of dollars to come up with algorithmic techniques that can provide personalized recommendations to their users.

## Business Problem and Data Overview

- Amazon, for example, is well-known for its accurate selection of recommendations in its online site. Amazon's recommendation system is capable of intelligently analyzing and predicting customers' shopping preferences in order to offer them a list of recommended products. Amazon's recommendation algorithm is therefore a key element in using AI to improve the personalization of its website. For example, one of the baseline recommendation models that Amazon uses is item-to-item collaborative filtering, which scales to massive data sets and produces high-quality recommendations in real-time.

- As a Data Science Manager at Amazon, I have been given the task of building a recommendation system to recommend products to customers based on their previous ratings for other products. I have a collection of labeled data of Amazon reviews of products. The goal is to extract meaningful insights from the data and build a recommendation system that helps in recommending products to online consumers.

## Business Problem and Data Overview

- Data Set :
  - The Amazon dataset contains the following attributes:
    - **userId:** Every user identified with a unique id
    - **productId:** Every product identified with a unique id

- **Rating:** The rating of the corresponding product by the corresponding user
- **timestamp:** Time of the rating. We will not use this column to solve the current problem.

7824482

## Exploratory Data Analysis

- There are total 7,824,482 rows and 3 columns. As this is a very large dataset, it will be computationally impossible to build a model. So we will take the users who given at least 50 ratings, and the products that have at least 5 ratings, as when we shop online we prefer to have some number of ratings of a product.

- After removing users with below 50 ratings and products with below 5 ratings, we created a new data frame which contains 65,290 rows and 3 columns.
- There are 2 columns namely user_id and prod_id which are of object data type and 1 column (ratings) of float data type.
- There are no missing values in the new data frame.

## Exploratory Data Analysis

- After checking summary statistics of ratings, we get minimum rating of 1 and maximum of 5. The average rating is 4.21. Median rating and 75%ile is 5.

- By observing rating plot distribution, we noticed that majority ratings are 5(60%) followed by 4.0 (12). Rest are under 10% of the count with minimum count is of rating 1.0.

- There are 1540 unique users and 5689 unique items in the raw data which have total 65,290 observations.

- Among the top 10 users with most number of ratings, highest number of rating given by a user is 295 followed by 230, which are very less than the actual number of products present in the data.

## Rank Based Model

- First, we build a dataframe with all the products with their average ratings and total count of ratings given to every product, representing the popularity of each product based on the number of interactions.

- Products with a minimum threshold of rating 50 or 100, were excluded to ensure the recommendations given on the basis of wide popularity of the products which will reduce the bias created by products with high rating and low number of interactions.

- Then, we sorted each product on the basis of total count of rating of products so that we prioritize the products with maximum interactions to appear on the top.

- The top 5 products selected from the sorted dataframe to recommend the users on the basis of rank of interactions.
- The most popular product (B0088CJT4U) has rating count of 206 and second most popular product has 184 ratings.

# User-User Similarity-based Model

- First, we initialised KKN basic model using cosine similarity measure and we fitted the model on the training set.
- After that, we computed the metrics (precision, recall, f1 score) to predict the performance of the model.

# User-User Similarity-based Model

- **<u>Observations</u> :**

1. **RMSE(Root mean squared error):**

- The RMSE value is 1.002 which means the average difference between actual and predicted rating is approximately 1 which is not significantly high.
- In this case study where rating scale is 1 to 5, average error of 1 is moderate and we can say that predictions are slightly close to the actual ratings given by users which is reasonable in basic model.

# User-User Similarity-based Model

- We can improve the accuracy by implementing advanced models like matrix factorization or by some improvements in similarity matrics.

**2. Precision :**

- Precision is a metric that measures the fraction of recommended items that are relevant to the user.
- Precision = Number of relevant items recommended / Total number of recommended items

- The basic model gave precision of 0.855 which means 85.5% of recommendations made by the model are relevant to the user.

# User-User Similarity-based Model

- This result indicates a effective recommendations given by the model that the users are likely to interact with.

**3. Recall :**

- It measures the proportion of relevant items recommended by the model out of all the actually relevant items to the user.

- Recall = Number of relevant items recommended / Total actual relevant items to the user

# User-User Similarity-based Model

- Model gave recall of 0.858 which means that 85.8% items recommended by the model are actually relevant to the user which are correctly identified by the model out of total number of relevant items.

- This is a fairly good score. A high recall is necessary to achieve to improve customer satisfaction and user experience. It ensures users get recommendations that they are genuinely interested in and reducing the missed opportunity.

4. **F1 Score :**

- It is a performance metric which is a harmonic mean of precision and recall. It is a trade off between precision and recall to get a balanced performance score.

# User-User Similarity-based Model

- The model gave f1 score of 0.856 which is good score in terms of performance.
- It indicates a strong effectiveness of the model of generating accurate and comprehensive recommendations of relevant items.
- The score of 85% ensures fair balance between precision and recall which are both high in our model.

➔ Below are the result of prediction of rating for a user and an item.

- The model predicted rating of 4.29 and the actual rating was not present in the dataset.

# User-User Similarity-based Model

- However, the model could not confidently generate the prediction due to insufficient data. The error has occurred because there are not enough neighbors or similar users for the user or item to make confident prediction. So the model labeled the prediction as ‘impossible’.

- The rating of 4.29 is not reliable due to the lack of enough similar users in the data. So we can improve the model prediction by tuning the hyperparameters of our KNNBasic model.

# User-User Similarity-based Model

- **<u>After Hyperparameter tuning :</u>**
- **<u>Observations :</u>**

1. **RMSE :**

- The RMSE value has slightly reduced to 0.9527 from previous baseline model(1.002). Lower error showing better predictions and less difference between actual and predicted rating.

- After tuning, model showing better accuracy in predictions of the ratings.

2. **Precision :**

# User-User Similarity-based Model

- After tuning, basic model gave precision of 84.7% which is insignificantly reduced due to trade off to increase recall and overall performance which is acceptable.

**3. Recall :**

- Recall has noticeably improved to 0.894 which indicates that 89% of the relevant items are recommended to the user.

- Recall increased after tuning of hyperparameters, which suggests that now our model is effectively recommending more relevant items to the users,

# User-User Similarity-based Model

- Now, model is more reliable to achieve more relevant recommendations of the items that the users are genuinely interested in.

4. **F1 – Score :**

- After tuning, F1-Score has improved from 85% to 87% which is a significant shift.

- The high F1-Score suggests that now the model is more capable of balancing the precision and recall, both of which are strong in this case.

## Overall Comparison of KNNBasic model before and after tuning :

# User-User Similarity-based Model

- The tuning has successfully reduced the RMSE and improved the recall which is a good sign as it suggest that after optimization, model is now generating more effective recommendations which will ultimately improve user satisfaction.

- Model is  now more able to find similar neighbors or users even though the dataset is sparse, which was the issue occurring in the model before tuning.

- F1-Score has also increased in the later model, and now model is more effective in balancing precision and recall which is necessary to not miss out opportunity or to not  recommending irrelevant items.

# User-User Similarity-based Model

- ✈ Even though precision has slight reduced which is generally required to achieve an outstanding trade off by improving recall which ultimately enhance user experience by focusing more on diverse and relevant suggestions while maintaining a certain level of accuracy in predictions.

- ✈ For our case study of amazon, which is a e-commerce platform, user satisfaction and engagement is more crucial, tuned model is more suitable for implementation.

- While predicting the rating of a certain user and item which the user has not interacted by using optimized model, we achieved the same results as baseline model. This maybe due to insufficient neighbors in the data which lead to fallback prediction like global mean.

# User-User Similarity-based Model

- **After implementing recommendation algorithm based on optimized KNNBasic model,** we predicted the top 5 items recommended to "`A3LDPF5FMB782Z`".

- **We got 4 products which are predicted by the tuned model to the user which all have 5.0 rating.**

# Item-Item Similarity-based Model

1. **RMSE :**

- We got average error of 99.50% which is closer to 1. This score suggest that model is predicting close rating but there needs to be some improvement as this error should be close to 0.
- This RMSE score of item based model is worse than the optimized basic model.

2. **Precision :**

- Precision of 0.838 indicates that approximately 84% of the items recommended are relevant to the user. It has reduced from our previous model which makes it clear that this item based model is predicting slightly irrelevant items.

# Item-Item Similarity-based Model

3. **Recall :**

- Model give a recall of 0.845 which means that model have been successful for identifying 84% of relevant items to the user.
- This score is good at predicting the items that the user rated mostly.
- However, this score is reduced from the previous optimized baseline model(89%), which makes it clear that it is missing a few relevant items from all the possible user relevant items so false negatives are more in this model.

4. **F1- Score :**

# Item-Item Similarity-based Model

- Item item based model generated f1 score of 0.841 which is reasonably good in terms of balance between precision and recall.

- Even though the trade off is overall well in this model, this is slightly lower than the f1 score of optimized user-user based model indicating that user based model was more capable of maintaining trade off between recommending accurate results and not missing out any relevant items.

- **Observations of item based model on user id A3LDPF5FMB782Z and product id 1400501466 :**

# Item-Item Similarity-based Model

- The user interacted with the item only 1 time in which the user gave the rating of 5 to the product. And the model predicted the rating of 4.27, which is slightly lower than the actual rating but its very close.
- Hence, ,model performed really well in identifying almost similar interest of the user for the product.
- The prediction is labelled was_impossible : false, which means model is able to find similar neighbours to make good prediction.

**Observations of item based model on user id A34BZM6S9L7QI4 and product id 1400501466 :**

- This user has not interacted with this item before but model predicted the rating of 4.29.

# Item-Item Similarity-based Model

- The estimated rating suggests that user is likely to show interest in the product and rate the item positively.
- This time the model marked the prediction as impossible because the model could not find enough similar neighbours to make better predictions.
- We can improve the recommendation by gathering more interactions of the user or the product.

- **After hyperparameter tuning :**
- **Observations :**

1. **RMSE :**

# Item-Item Similarity-based Model

- After tuning, the average error between actual and predicted rating reduced to 95.76% from baseline model(99.50%). This improvement is indicative of increased effectiveness of the optimized model.

2. **Precision :**

- Precision score of 83.9% shows that the model is capturing most of the relevant items from all the true and false positive.
- However, precision has not improved even after tuning, this can be acceptable as this can be due to the trade off to increase recall.

3. **Recall :**

# Item-Item Similarity-based Model

- The optimized model gave recall of 0.88 which has remarkably improved from the untuned baseline model which showed recall of 84%.
- High recall shows that model is now more efficient in recognizing user preference out of actual user likings.
- This will enhance customer engagement and is less likely to miss possibility of generating more revenue.

**4. F1 – Score :**

- The performance metric has moderately increased to 85.9% which suggest that tuned item based model is more reliable to give strong results.

# Item-Item Similarity-based Model

- **Observations of item based model on user id A3LDPF5FMB782Z and product id 1400501466 :**
- This user has interacted with the product before and gave a rating of 5. The optimized model gave a prediction of 4.67 which is quite close to the actual rating.

# Matrix Factorization based Model

1. **RMSE :**

- This model give an average error of 88.82% which is comparatively low than previous models. It effectively captures latent patterns in the user user matrix.

2. **Precision :**

- The model achieve the precision of 85.3% which is quite good in terms of finding out relevant items out of the recommended items.

3. **Recall :**

- The recall is relatively high as compared with precision. The score of 88% is significantly a good score in achieving relevancy.

# Matrix Factorization based Model

4. **F1 – Score :**

- The f1 score of 0.866 is overall well and its capable in maintaining balance between performance metrics.

# Matrix Factorization based Model

- Compared to other baseline models, matrix factorization model is better in providing accuracy and coverage.

✞ **Observation of matrix model on user id A3LDPF5FMB782Z and product id 1400501466**:

- The user has interacted before with this item and gave a rating of 5 but the model predicted the rating of 4.08 which is on the positive side. The prediction is close to the actual rating.

✞ **Observation of matrix model on user id A34BZM6S9L7QI4 and product id 1400501466:**

- This user has never interacted with this item but the model provided the rating of 4.40 which means model is suggesting that the user is likely to engage with the product and prefer it if its recommended.
- The model marked this prediction as not impossible because it could successfully found similar users or items to the user which was a challenge for the previous basic models.

# Matrix Factorization based Model

- **After Hyperparameter Tuning :**

1. **RMSE :**

- After tuning, model's error has been slight reduced to 88.0% which indicates a better accuracy.

2. **Precision :**

- The optimization really improved the precision (85.4%), it means its not doing the trade off overly.

3. **Recall :**

- Recall remained high (87.8%) which means model is performing well before and after tuning.

4. **F1 – Score :**

- The optimized model gave f1 score of 86.6% which suggest that model is good at maintaining balance and robustness of the outcomes.

✞ **Observation of optimized matrix based model on user id A3LDPF5FMB782Z and product id 1400501466 :**

- The user has interacted before and given the rating of 5 to the item and optimized model gave rating of 4.13 which is quite close and suggesting the item to the user.

✞ **Observation of optimized matrix based model on user id A34BZM6S9L7QI4 and product id 1400501466 :**

# Matrix Factorization based Model

- This user never interacted before with the item but model gave prediction of 4.22 which again recommending to this user as well which indicates that model is capable of identifying similar neighbours.

# Conclusion and Recommendations

- **The table below summarizes the performance evaluation metrics of all the preformed models:**

| Model | RMSE | Precision | Recall | F1 - score |
|---|---|---|---|---|
| User based KNN(baseline) | 1.0012 | 0.855 | 0.858 | 0.856 |
| User based KNN(tuned) | 0.9527 | 0.847 | 0.894 | 0.870 |
| Item based KNN (baseline) | 0.9950 | 0.838 | 0.845 | 0.841 |
| Item based KNN (tuned) | 0.9576 | 0.839 | 0.880 | 0.859 |
| Matrix Factorization(baseline) | 0.8882 | 0.853 | 0.880 | 0.866 |
| Matrix Factorize(tuned) | 0.8808 | 0.854 | 0.878 | 0.866 |

## Conclusion and Recommendations

- After analyzing performance of all the models, we came to the conclusion that matrix factorization(tuned) had the lowest rmse and more balanced precision, recall and f1 score.
- However, we can observe that both user user and item item similarity models improved their performance significantly after hyperparameter tuning. But the completeness and balance of matrix factorization model achieved the best scores for each metrics.
- Matrix factorization model assumes that items and users are present in some low dimensional space describing their properties and recommend an item based on its proximity to the user in the latent space, implying it accounts for latent factors as well.

- By comparing the recall and f1 score of user user(tuned) model with matrix factorization(tuned), we can see that both metrics are better of the previous one but rmse of user user is way worse than the later. So the balance and robustness of matrix model is considerably reliable.

## Conclusion and Recommendations

✞ **Recommendations for business :**

- Implement the best model and apply the predictions to personalize the suggestions for users.
- Gathering more detailed information(e.g. location, age group) of the users or the products can help to modify and filter the recommendation.

- Recommendations can be better categorised for users in the same geographical location. Also, age and other demographical information about the users can help filter out preferable items for same age people.
- Conduct regular feedback or ratings from the users to cross check the model's performance and

# APPENDIX

to evaluate user satisfaction.
- Recommend premium products on the top to generate the possibility of more revenue.
- Regularly monitoring of performance metrics and implementing personalized approaches to address user behaviour pattern.

# Happy Learning !