**UNSUPERVISED ML -**

**BOOK RECOMMENDATION SYSTEM**



**By : Kaiwalya Zankar**

**Data Science Trainee**

**at Almabetter**

## 

**Abstract**

Recommender systems are one of the most applied methods in machine learning and find applications in many areas, ranging from economics to the Internet of things Using clustering can address several known issues in recommendation systems, including increasing the diversity, consistency, and reliability of recommendations; the data sparsity of user-preference matrices; and changes in user preferences over time.

Suggestion of items, according to user preferences are most important, so suggestion according to similarities provides suitable recommendation. This method of recommendation system works as suggestion, customization,

learning, administration and this all provides user for the items suggestion and decision making.

**Introduction**

Book recommendation is created and deployed in this approach of work, which helps in recommending books. Recommendation achieved by the users feedbacks and rating, this is the online which analyze the ratings, comments and reviews of user, negative positive nature of comments using opinion mining. User searching for the interested book will be displayed in top list and also can read feedback given by people about the book or any searched items. Whenever user search for any book from the large data available, he gets confused from the number of displayed item, which one to choose. In that case recommendation helps and displays on the interested items. This is the trustworthy approach, which is used in this paper where selection is based on the services of cloud using collaborative filtering. Set of user requirements are searched in cloud services. From the targeted user it computes the similar preferences to provide the service of recommendation to each user. Sorting of services are done based on the degree of recommendation for each user. In recent years use of recommendation is increased because of the use of some techniques. This techniques displays the selected and preferred items on the top list, it helps in suggesting user, the relevant items which is the need of requirement to user.

# **Problem Statement**

During the last few decades, with the rise of Youtube, Amazon, Netflix, and many other such web services, recommender systems have taken more and more place in our lives. From e-commerce (suggest to buyers articles that could interest them) to online advertisement (suggest to users the right contents, matching their preferences), recommender systems are today unavoidable in our daily online journeys.

In a very general way, recommender systems are algorithms aimed at suggesting relevant items to users (items being movies to watch, text to read, products to buy, or anything else depending on industries).

Recommender systems are really critical in some industries as they can generate a huge amount of income when they are efficient or also be a way to stand out significantly from competitors. The main objective is to create a book recommendation system for users

**Objective**

The project's main goal is to create a model that can perform Clustering on comparable material by matching text-based attributes.

**Data Description**

**The Book-Crossing dataset comprises 3 files.**

**● Users :**

**Contains the users. Note that user IDs (User-ID) have been anonymized and map to integers. Demographic data is provided (Location, Age) if available. Otherwise, these fields contain NULL values.**

**● Books :**

**Books are identified by their respective ISBN. Invalid ISBNs have already been removed from the dataset. Moreover, some content-based information is given (Book-Title, Book-Author, Year-Of-Publication, Publisher), obtained from Amazon Web Services. Note that in the case of several authors, only the first is provided. URLs linking to cover images are also given, appearing in three different flavors (Image-URL-S, Image-URL-M, Image-URL-L), i.e., small, medium, large. These URLs point to the Amazon website.**

**● Ratings :**

**Contains the book rating information. Ratings (Book-Rating) are either explicit, expressed on a scale from 1-10 (higher values denoting higher appreciation), or implicit, expressed by 0**

**Challenges Faced**

The following are the challenges faced in the data analysis:

* Conversion of Datetime features, categorical features.
* Feature engineering
* Model Implementation

**Approach**

As we have to work on te past beaviour of te dataset and as I know at such condition collaborative filltering works .

So further analysis is done using item-item collaborative filtering , and user – item collaborative filltering

**Tools Used**

The whole project was done using python, in google colaboratory. Following libraries were used for analyzing the data and visualizing it and to build the model to predict the bike count required at each hour for the stable supply of rental bikes.

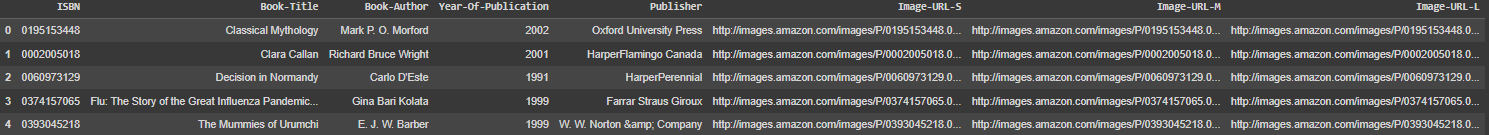
* Pandas: Extensively used to load and wrangle with the dataset.
* Sys: The sys module in Python provides various functions and variables that are used to manipulate different parts of the Python runtime environment
* Numpy: For some math operations in predictions.
* Matplotlib: Used for visualization.
* Seaborn: Used for visualization.
* Random: Python Random module is an in-built module of Python which is used to generate random numbers.
* Warnings: For filtering and ignoring the warnings.

The below table shows the dataset in the form of Pandas DataFrame

1. **User data-**

|  |  |  |  |
| --- | --- | --- | --- |
|  | User id | Location | age |
| **0** | 1 | nyc, new york, usa | NaN |
| **1** | 2 | stockton, california, usa | 18.0 |
| **2** | 3 | moscow, yukon territory, russia | NaN |
| **3** | 4 | porto, v.n.gaia, portugal | 17.0 |
| **4** | 5 | farnborough, hants, united kingdom | NaN |

1. **Books data-**

****

1. **Ratings Data-**

|  |  |  |  |
| --- | --- | --- | --- |
|  | User id | ISBN | Book - rating |
| **0** | 276725 | 034545104X | 0 |
| **1** | 276726 | 0155061224 | 5 |
| **2** | 276727 | 0446520802 | 0 |
| **3** | 276729 | 052165615X | 3 |
| **4** | 276729 | 0521795028 | 6 |

Users data consist of 278858 observations with 3 columns,

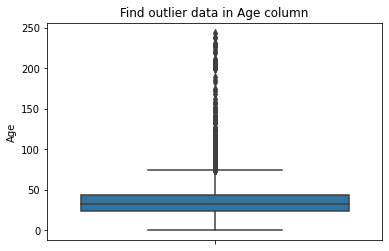
Books data consist of 271360 observations with 8 column, and

Rating data consist of 1149780 observations with 3 columns

1. User data set

After some processing I get to know that in user data set around 39% data does not contain age

Outliers in age column

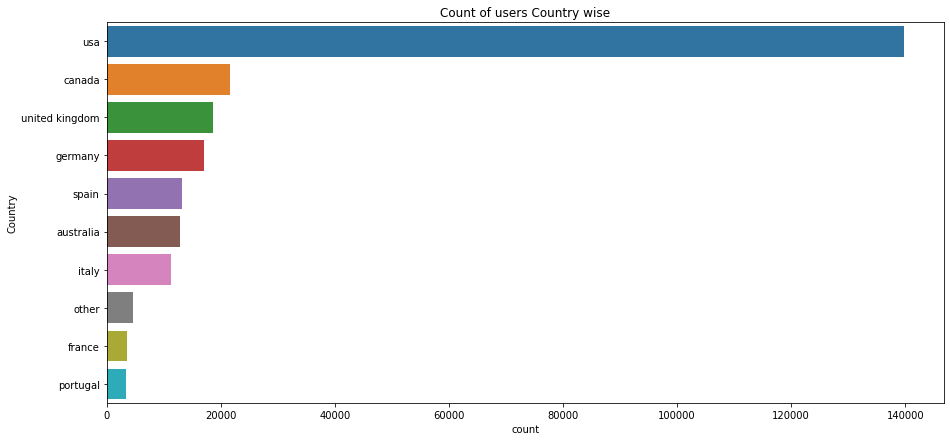


The dots represent the outliers here the maximum age of the user is 244 as it is not possible so the user data contain the outliers.

User data on basis of location

The user data consist of users around 529 countries.

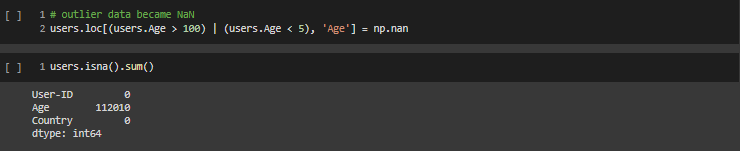
Most number of users from top 10 countries



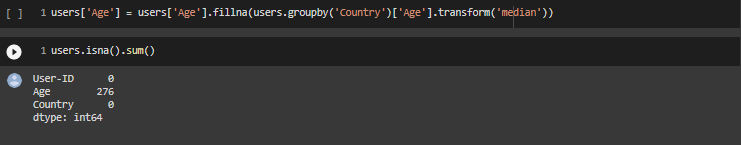
The most user are from the USA.

Treating outliers in age

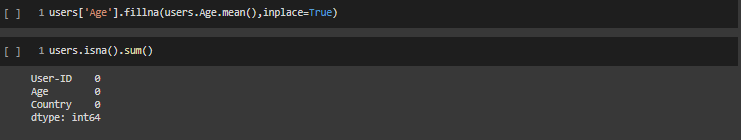
As age below 5 and above 100 does not make sense… hence replacing them by NaNs



Age has positive Skewness so I can use median to fill NaN values but for this I don't like to fill Nan value just for one range of age. To handle this I have used country column to fill Nan.

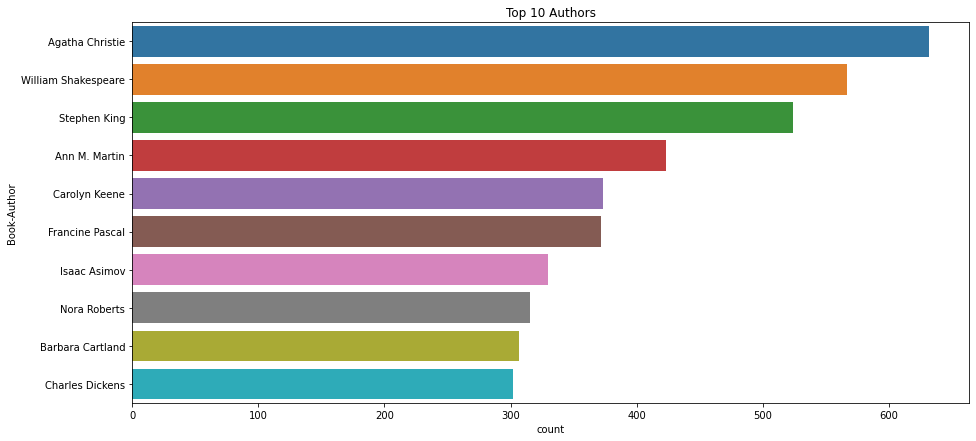


Still I have 276 Nan values .so I fill them with mean

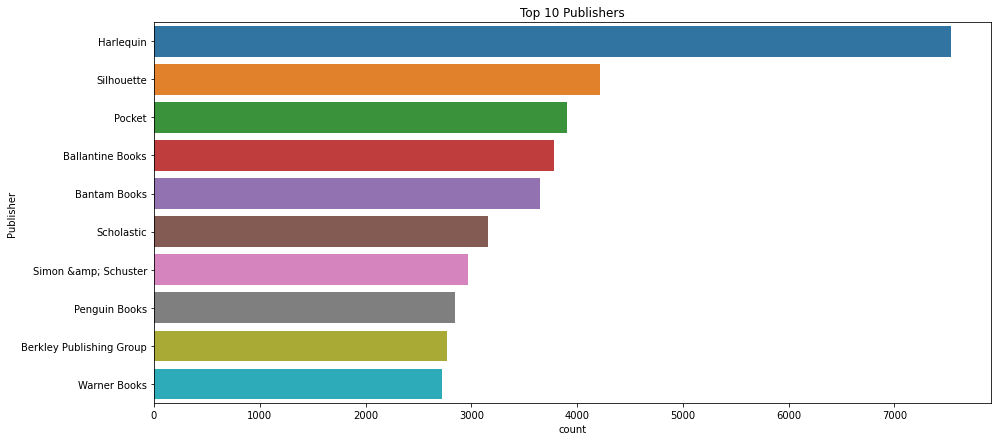


1. Books Dataset

Top Authors which have written the most books

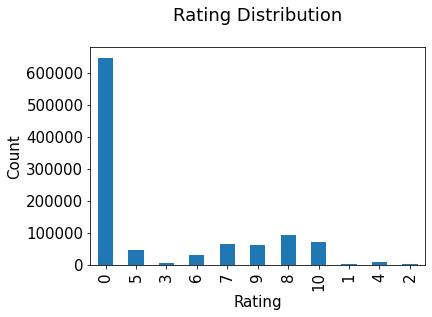


Top publisher which publish most books.



1. Rating Dataset

Rating distribution



The ratings are very unevenly distributed, and the vast majority of ratings are 0

Most rated books

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | ISBN | Book Title | Book Author | Year of publication | Publisher |
| **0** | 0316666343 | The Lovely Bones: A Novel | Alice Sebold | 2002.0 | Little, Brown |
| **1** | 0971880107 | Wild Animus | Rich Shapero | 2004.0 | Too Far |
| **2** | 0385504209 | The Da Vinci Code | Dan Brown | 2003.0 | Doubleday |
| **3** | 0312195516 | The Red Tent (Bestselling Backlist) | Anita Diamant | 1998.0 | Picador USA |
| **4** | 0060928336 | Divine Secrets of the Ya-Ya Sisterhood: A Novel | Rebecca Wells | 1997.0 | Perennial |

Merging All Dataset



final dataset consists of 383842 observations with 11 columns.

**Popularity Based Filtering**

As the name suggests Popularity based recommendation system works with the trend. It basically uses the items which are in trend right now. For example, if any book which is usually bought by every new user then there are chances that it may suggest that book to the user who just signed up.

Book weighted avg formula:

**Weighted Rating(WR)=[vR/(v+m)]+[mC/(v+m)]**

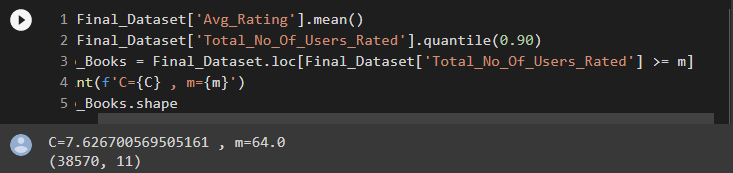
where,

v is the number of votes for the books;

m is the minimum votes required to be listed in the chart;

R is the average rating of the book; and

C is the mean vote across the whole report.



# 

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Book Title | Total number of user | Avg rating | Score |
| **0** | Harry Potter and the Goblet of Fire (Book 4) | 137 | 9.262774 | 8.741835 |
| **1** | Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback)) | 313 | 8.939297 | 8.716469 |
| **2** | Harry Potter and the Order of the Phoenix (Book 5) | 206 | 9.033981 | 8.700403 |
| **3** | To Kill a Mockingbird | 214 | 8.943925 | 8.640679 |
| **4** | Harry Potter and the Prisoner of Azkaban (Book 3) | 133 | 9.082707 | 8.609690 |
| **5** | The Return of the King (The Lord of the Rings, Part 3) | 77 | 9.402597 | 8.596517 |
| **6** | Harry Potter and the Prisoner of Azkaban (Book 3) | 141 | 9.035461 | 8.595653 |
| **7** | Harry Potter and the Sorcerer's Stone (Book 1) | 119 | 8.983193 | 8.508791 |
| **8** | Harry Potter and the Chamber of Secrets (Book 2) | 189 | 8.783069 | 8.490549 |
| **9** | Harry Potter and the Chamber of Secrets (Book 2) | 126 | 8.920635 | 8.484783 |
| **10** | The Two Towers (The Lord of the Rings, Part 2) | 83 | 9.120482 | 8.470128 |
| **11** | Harry Potter and the Goblet of Fire (Book 4) | 110 | 8.954545 | 8.466143 |
| **12** | The Fellowship of the Ring (The Lord of the Rings, Part 1) | 131 | 8.839695 | 8.441584 |
| **13** | The Hobbit : The Enchanting Prelude to The Lord of the Rings | 161 | 8.739130 | 8.422706 |
| **14** | Ender's Game (Ender Wiggins Saga (Paperback)) | 117 | 8.837607 | 8.409441 |
| **15** | Tuesdays with Morrie: An Old Man, a Young Man, and Life's Greatest Lesson | 200 | 8.615000 | 8.375412 |
| **16** | Charlotte's Web (Trophy Newbery) | 68 | 9.073529 | 8.372037 |
| **17** | Dune (Remembering Tomorrow) | 75 | 8.973333 | 8.353301 |
| **18** | A Prayer for Owen Meany | 181 | 8.607735 | 8.351465 |
| **19** | Fahrenheit 451 | 164 | 8.628049 | 8.346969 |

# **The Popularity based recommender provide a general chart of recommended books to all the users. They are not sensitive to the interests and tastes of a particular user.**

# **Model Based Collaborative Filtering Recommender.**

The goal of the recommender system is to predict user preference for a set of items based on the past experience.

Two the most popular approaches are Content-Based and Collaborative Filtering.

Collaborative filtering is a technique used by websites like Amazon, YouTube, and Netflix. It filters out items that a user might like on the basis of reactions of similar users. There are two categories of collaborative filtering algorithms: memory based and model based.

Model based approach involves building machine learning algorithms to predict user's ratings. They involve dimensionality reduction methods that reduce high dimensional matrix containing abundant number of missing values with a much smaller matrix in lower-dimensional space.

The goal of this section is to compare SVD and NMF algorithms, try different configurations of parameters and explore obtained results

**SVD and NMF models comparison**

Singular Value Decomposition (SVD) and Non-negative Matrix Factorization (NMF) are matrix factorization techniques used for dimensionality reduction. Surprise package provides implementation of those algorithms.

SVD

|  |  |
| --- | --- |
| Test\_rmse | 1.601489 |
| Test\_mae | 1.239510 |
| Fit\_time | 12.735846 |
| Test\_time | 0.979896 |
| Dtype: float64 |  |

NMF

|  |  |
| --- | --- |
| Test\_rmse | 2.618516 |
| Test\_mae | 2.236896 |
| Fit\_time | 17.139331 |
| Test\_time | 0.736626 |
| Dtype: float64 |  |

Optimization of SVD algorithm

Grid Search Cross Validation computes accuracy metrics for an algorithm on various combinations of parameters, over a cross-validation procedure. It's useful for finding the best configuration of parameters.

It is used to find the best setting of parameters:

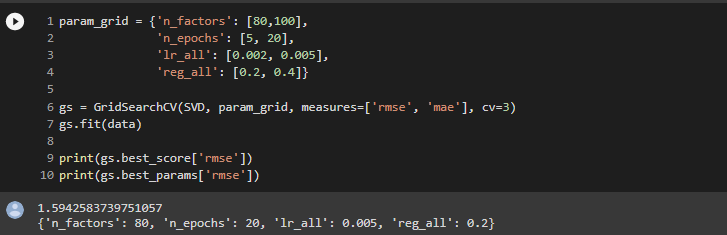
n\_factors - the number of factors

n\_epochs - the number of iteration of the SGD procedure

lr\_all - the learning rate for all parameters

reg\_all - the regularization term for all parameters

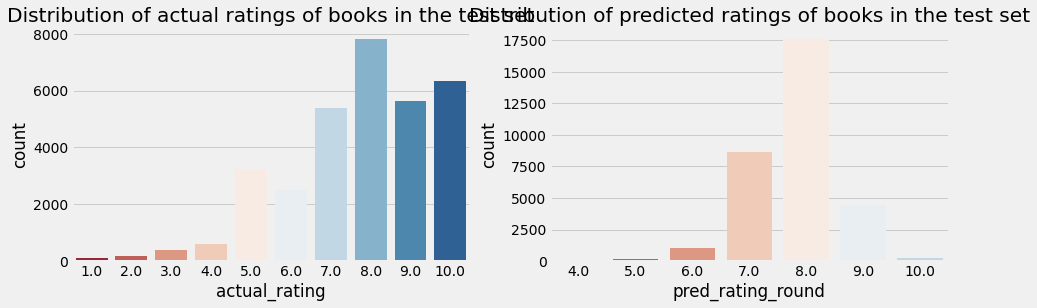
As a result, regarding the majority of parameters, the default setting is the most optimal one. The improvement obtained with Grid Search is very small.



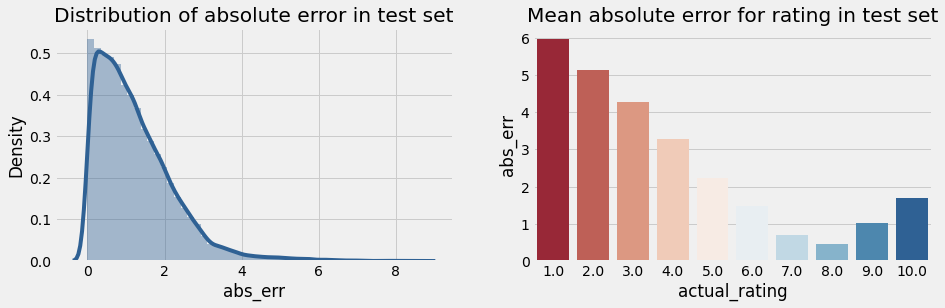
Analysis of Collaborative Filtering model Results

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | User id | isbn | Actual rating | Pred rating | Impossible | Pred rating round | Abs err |
| **30456** | 152645 | 038072152X | 6.0 | 6.599161 | False | 7.0 | 0.599161 |
| **17899** | 99347 | 0671693816 | 9.0 | 8.210509 | False | 8.0 | 0.789491 |
| **1231** | 148590 | 0679441557 | 9.0 | 7.509133 | False | 8.0 | 1.490867 |
| **28855** | 166922 | 0140620427 | 9.0 | 7.734341 | False | 8.0 | 1.265659 |
| **12493** | 149763 | 034541215X | 9.0 | 8.160448 | False | 8.0 | 0.839552 |

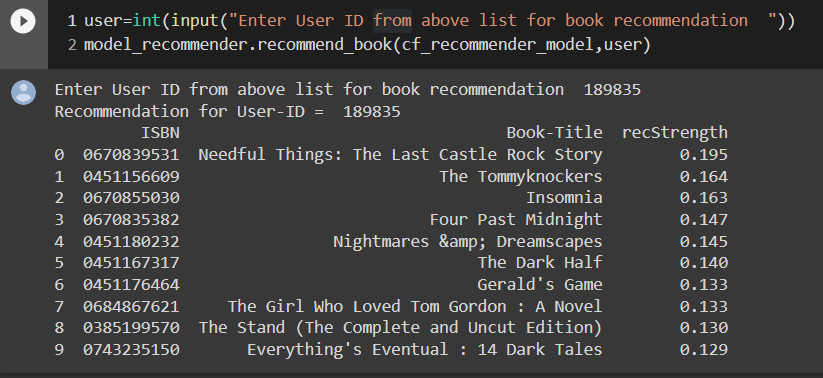
Distribution of actual ratings of books in the test set and predicted ratings of books in the test set



distribution of absolute error with density and actual rating with mean absolute error



**lets recommend book to user-id 189835**

****

**Evaluating the Model**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | hits@5 count | hits@10 count | Interacted count | recall@5 | recall@10 | User id |
| **10** | 252 | 338 | 1389 | 0.181 | 0.243 | 11676 |
| **31** | 185 | 241 | 1138 | 0.163 | 0.212 | 98391 |
| **45** | 21 | 28 | 380 | 0.055 | 0.074 | 189835 |
| **30** | 81 | 101 | 369 | 0.220 | 0.274 | 153662 |
| **70** | 30 | 34 | 236 | 0.127 | 0.144 | 23902 |
| **7** | 29 | 49 | 204 | 0.142 | 0.240 | 235105 |
| **47** | 26 | 31 | 203 | 0.128 | 0.153 | 76499 |
| **50** | 23 | 34 | 193 | 0.119 | 0.176 | 171118 |
| **42** | 61 | 71 | 192 | 0.318 | 0.370 | 16795 |
| **43** | 22 | 32 | 188 | 0.117 | 0.170 | 248718 |

# 

# 

# **Conclusion-**

In EDA, the Top-10 most rated books were essentially novels. Books like The Lovely Bone and The Secret Life of Bees were very well perceived.

Majority of the readers were of the age bracket 20-35 and most of them came from North American and European countries namely USA, Canada, UK, Germany and Spain.

If we look at the ratings distribution, most of the books have high ratings with maximum books being rated 8. Ratings below 5 are few in number.

Author with the most books was Agatha Christie, William Shakespeare and Stephen King.

For modelling, it was observed that for model based collaborative filtering SVD technique worked way better than NMF with lower Mean Absolute Error (MAE) .

Amongst the memory based approach, item-item CF performed better than user-user CF because of lower computation .

**Thank you..!!**