**Capstone Project**

# **Book Recommendation System**

**Het Kothari, Kamya Malhotra,**

**Saurabh Daund, Amir Khan,**

**Mouleena Jaiswal.**

**Contents:**

1. **Abstract**
2. **Introduction**
3. **Problem Statement**
4. **Steps Involved**
5. **Data Visualization**
6. **Model Training and Evaluation**
7. **Conclusion**
8. **References**

**Abstract:**

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).

Recommendation 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.

Content:

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.

**Introduction:**

Recommending books is a very subjective matter and people often don't follow a certain trend in reading books and always want to try something new. We try to use a technique which can help with that as well.

**Problem Statement:**

This project is aimed at creating an efficient book recommending model which will recommend books to customers based on previous history and other factors. We will tackle the problem statement in the following steps:

Step 1: Data Overview

Step 2: Data preprocessing

Step 3: Visualizing the features

Step 4: Implementing Collaborative filtering using kNN

Step 5: Implementing Collaborative filtering using SVD

Step 6: Evaluating using Top-N accuracy based metrics

Step 7: Conclusion

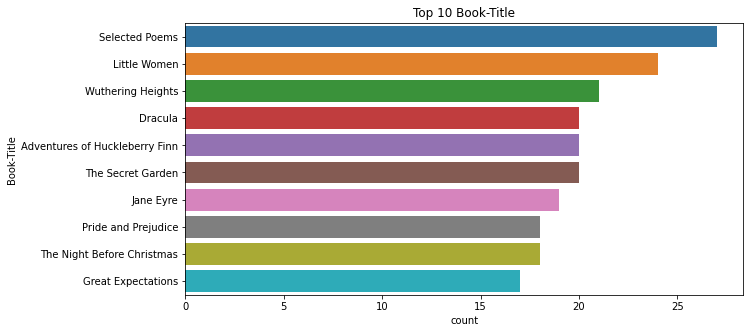
**Steps Involved:**

1. Importing Packages: Here, we import inbuilt libraries like NumPy, Pandas to prepare the data and Seaborn and Matplotlib for graphical representations of data or data visualizations. We also import various libraries for creating our unsupervised models.
2. Data Overview: We use various python and pandas functions to understand the features of the data and how it’s shaped and its data types.
3. Data preprocessing: We have three tables for this project. We drop unnecessary rows and handle null values. We also remove outliers and finally combine tables to our benefit.
4. Visualizing the features: Using seaborn library with matplotlib we plot various graphs of individual variables and combined graphs of various variables to understand the spread of the data and detect outliers.
5. Implementing Collaborative filtering using kNN: We convert our table to a 2D matrix, and fill the missing values with zeros (since we will calculate distances between rating vectors). We then transform the values(ratings) of the matrix data frame into a scipy sparse matrix for more efficient calculations. Finding the Nearest Neighbors We use unsupervised algorithms with sklearn.neighbors. The algorithm we use to compute the nearest neighbors is “brute”, and we specify “metric=cosine”. Finally, we fit the model.
6. Implementing Collaborative filtering using SVD: The Singular Value Decomposition (SVD), a method from linear algebra that has been generally used as a dimensionality reduction technique in machine learning. SVD is a matrix factorisation technique, which reduces the number of features of a dataset by reducing the space dimension from N-dimension to K-dimension (where K<N). In the context of the recommender system, the SVD is used as a collaborative filtering technique.from scipy.sparse.linalg we use svds for model training.
7. Evaluating using Top-N accuracy based metrics: We choose to work with Top-N accuracy metrics, which evaluates the accuracy of the top recommendations provided to a user, compared to the items the user has actually interacted with in the test set.
8. Conclusion: We finally understand our dataset and our results implemented using kNN and SVD for collaborative filtering.

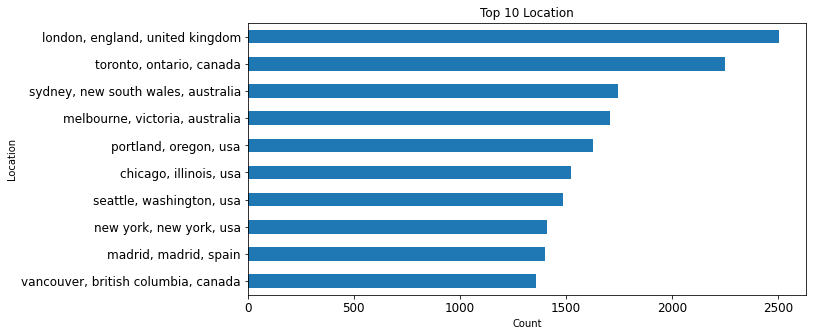
**Data Visualization:**

Let’s understand various relation among target and other variables:

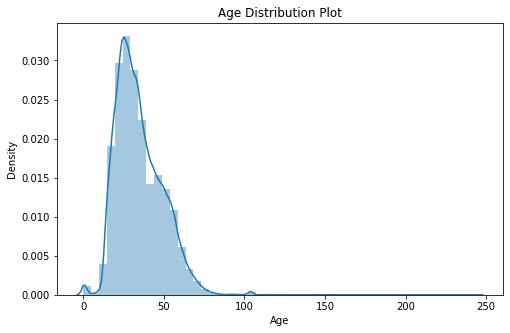
1. **We see that Selected poems is the highest read book in our dataset.**

****

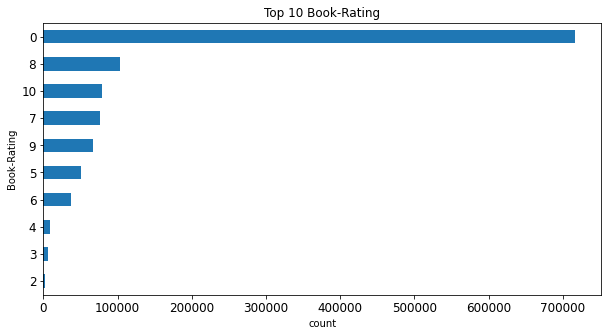
1. **We can see most of the books published were in London.**

****

1. **As we can see, the highest readers are around the age of 25.**

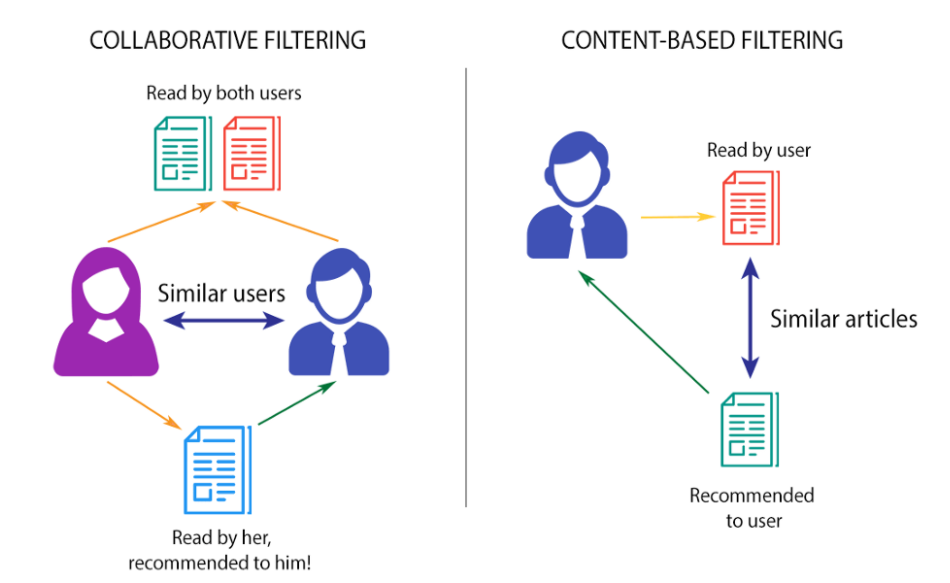
****

1. **Again we can see that most books haven’t received any rating.**

****

**Model Training and Evaluation:**

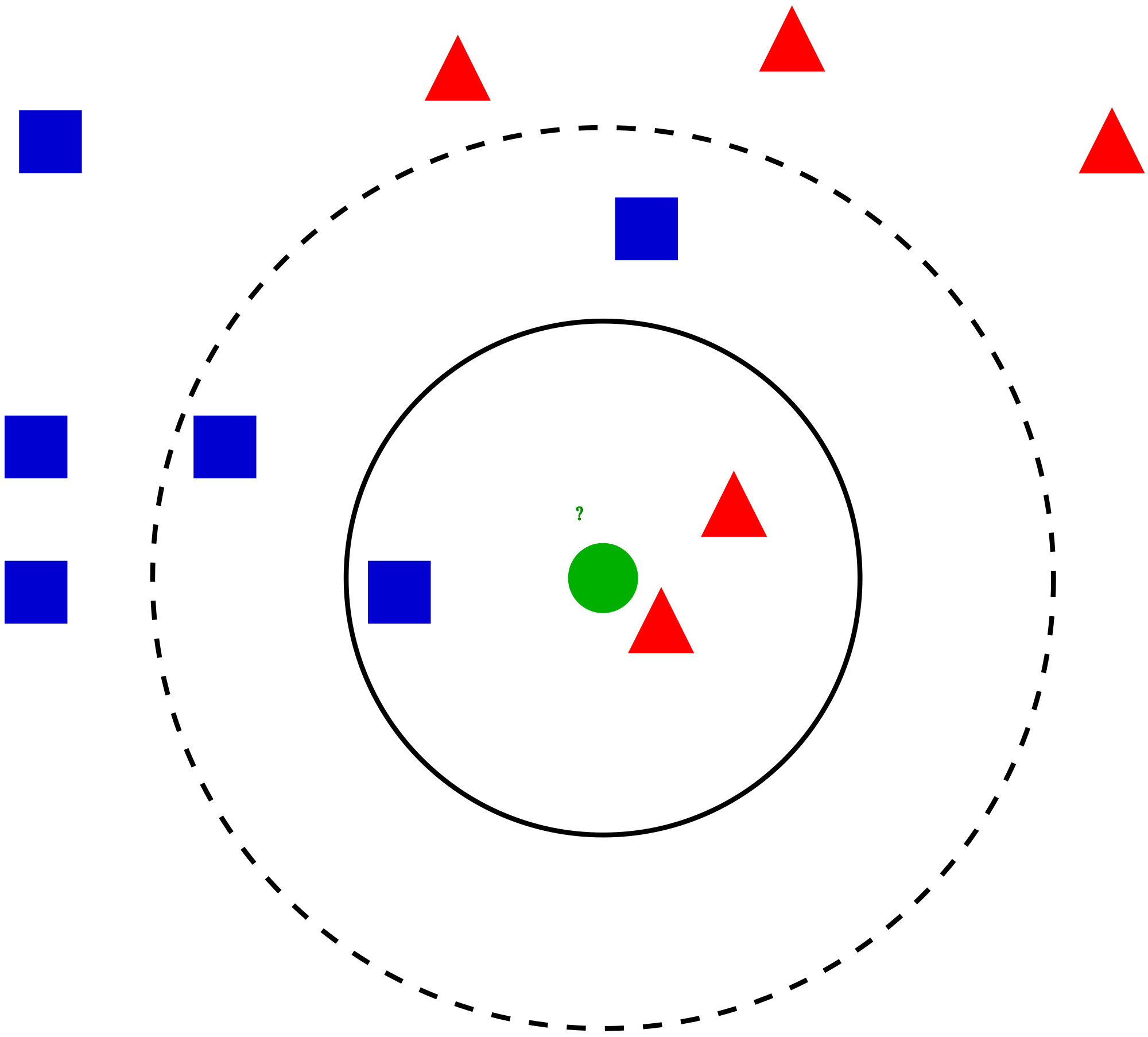
**Collaborative Filtering:**



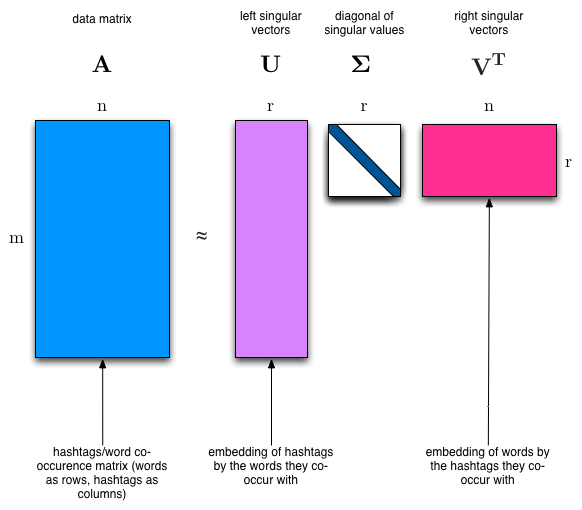
This filtering method is usually based on collecting and analyzing information on user’s behaviors, their activities or preferences, and predicting what they will like based on the similarity with other users. A key advantage of the collaborative filtering approach is that it does not rely on machine analyzable content and thus it is capable of accurately recommending complex items such as books without requiring an “understanding” of the item itself.

For collaborative filtering wwe use two methods:

1. Nearest Neighbors: [NearestNeighbors](https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.NearestNeighbors.html#sklearn.neighbors.NearestNeighbors) implements unsupervised nearest neighbors learning. It acts as a uniform interface to three different nearest neighbors algorithms: [BallTree](https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.BallTree.html#sklearn.neighbors.BallTree), [KDTree](https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KDTree.html#sklearn.neighbors.KDTree), and a brute-force algorithm based on routines in [sklearn.metrics.pairwise](https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics.pairwise). The choice of the neighbor's search algorithm is controlled through the keyword 'algorithm', which must be one of ['auto', 'ball\_tree', 'kd\_tree', 'brute']. When the default value 'auto' is passed, the algorithm attempts to determine the best approach from the training data. For our data set we used the brute force method.Fast computation of nearest neighbors is an active area of research in machine learning. This can be achieved by brute force.



1. Singular Value Decomposition: SVD is a classical method from linear algebra is getting popular in the field of data science and machine learning. This popularity is because of its application in developing recommender systems. There are a lot of online user-centric applications such as video players, music players, e-commerce applications, etc., where users are recommended with further items to engage with.



**Model evaluation:**

**Since this is an unsupervised learning problem common metrics cannot be used here.**

**We check the working of the kNN model by practically implementing it and seeing the results.**

**A example:**

Recommendations for The Unbearable Lightness of Being:

1: Red Square, with distance of 22.58317958127243:

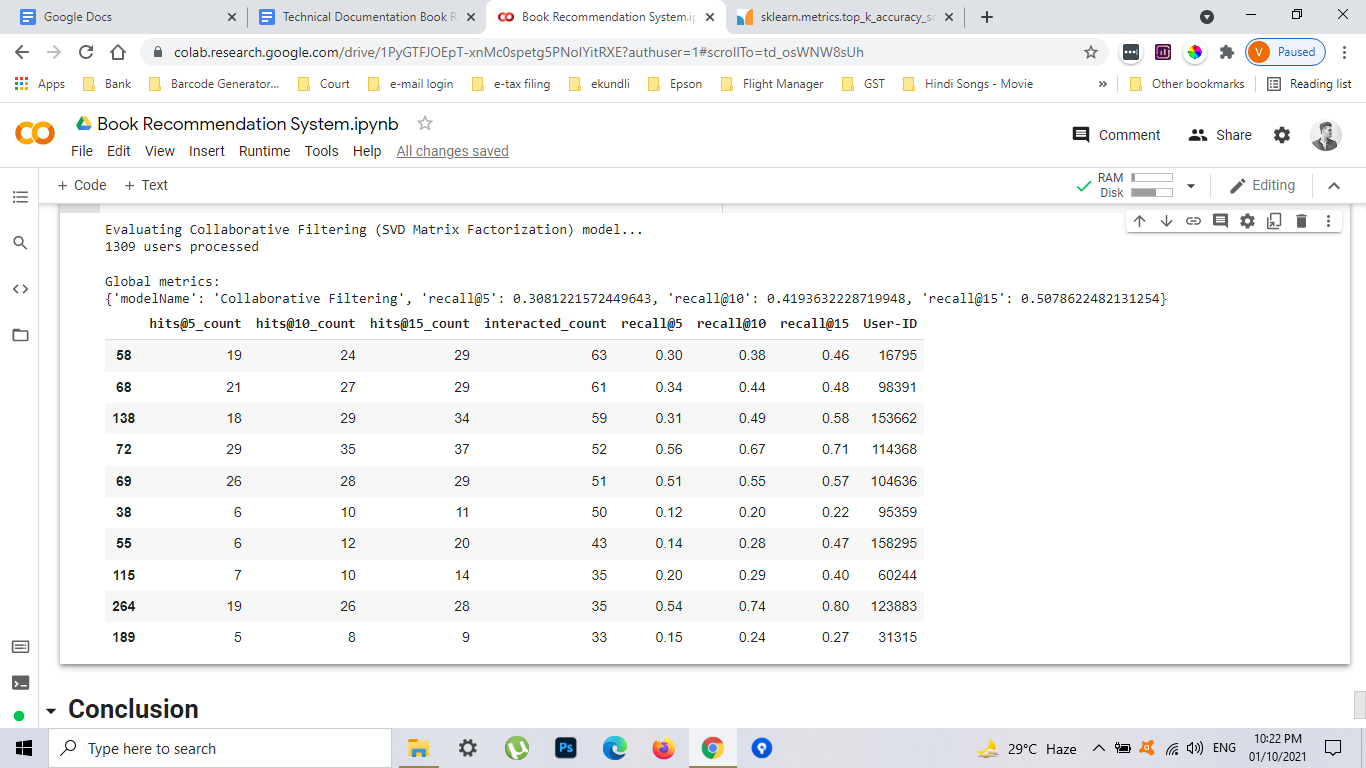
2: Random Winds, with distance of 23.811761799581316:

3: House, with distance of 24.779023386727733:

4: Privileged Information, with distance of 24.819347291981714:

5: Acts of Love, with distance of 24.95996794869737:

**After implementing SVD we use a common way of evaluating using Top-N accuracy metric.This metric computes the number of times where the correct label is among the top k labels predicted (ranked by predicted scores).**



**As we can see, we received a score for around 50 for recall@15.**

**Conclusion:**

As we can see, after implementing Collaborative Filtering and evaluating it using SVD matrix we are satisfied with the results. A recall rate of around 50 for hit@15 is fair enough for such a large dataset. Also since it is an unsupervised learning algorithm trying to find good books that users will like which in itself is a very vast and complicated study.

**References:**

1. GeeksforGeeks
2. Stackoverflow
3. Almabetter
4. Youtube
5. Github