Movie Recommender System

- Gunjan Gupta - 2019111035
- Pratishtha Abrol - 2020121002

Introduction :

The recommender systems are employed to help users to find their items based on their preferences. They produce individualized recommendations as output or have the effect of guiding the user in a personalized way to find interesting or useful items in a large amount of other items. The importance of recommendation systems is increasing day by day due to the massive amount of data and information-overloaded arising from the internet.

Pre-Processing :

Apart from as asked by the assignment, we also transformed the dataframe to come into a True/False matrix as required by the Apriori algorithm. We dropped non-contributing columns. And used a 10% random sample of the data for visualisation.

Apriori :

We are using the Apriori algorithm to extract the set of all association rules of form $X \rightarrow Y$, where X contains a single movie and Y contains the set of movies from the training set.
It is used to mine all frequent itemsets in the database. The algorithm makes many searches in the database to find frequent item sets whereas; k-item sets are used to generate k+1-itemsets. Each k-item set must be greater than or equal to minimum support threshold frequency. Otherwise, it is called candidate item sets. We used Apriori algorithm in generating association rules to find frequency of 1-itemsets that contain only one item by counting each item in the dataset. The frequency of 1-itemsets is used to find the item sets in 2-itemsets which in turn is used to find 3-itemsets and so on until there are not any more k-item sets. If an item set is not frequent, any large subset from it is also non-frequent.

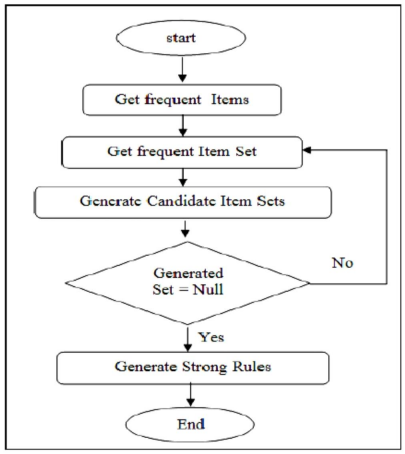


Figure 2: Flowchart of Apriori algorithm

Why Apriori is a good choice?

- This is the most simple and easy-to-understand algorithm among association rule learning algorithms
- The resulting rules are intuitive and easy to communicate to an end user
- The apriori algorithm doesn't scale well and thus, should be used with smaller datasets
- It doesn't require labeled data as it is fully unsupervised; as a result, you can use it in many different situations because unlabeled data is often more accessible
- Many extensions were proposed for different use cases based on this implementation—for example, there are association learning algorithms that take into account the ordering of items, their number, and associated timestamps
- The algorithm is exhaustive, so it finds all the rules with the specified support and confidence

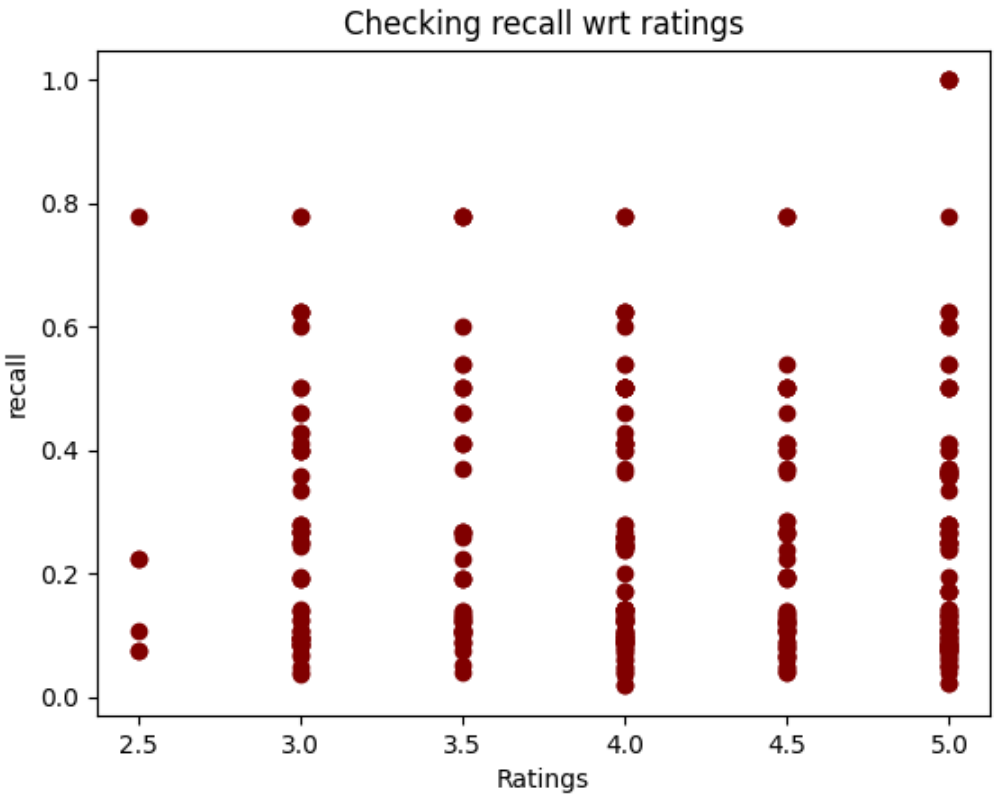
How did we build the recommender?

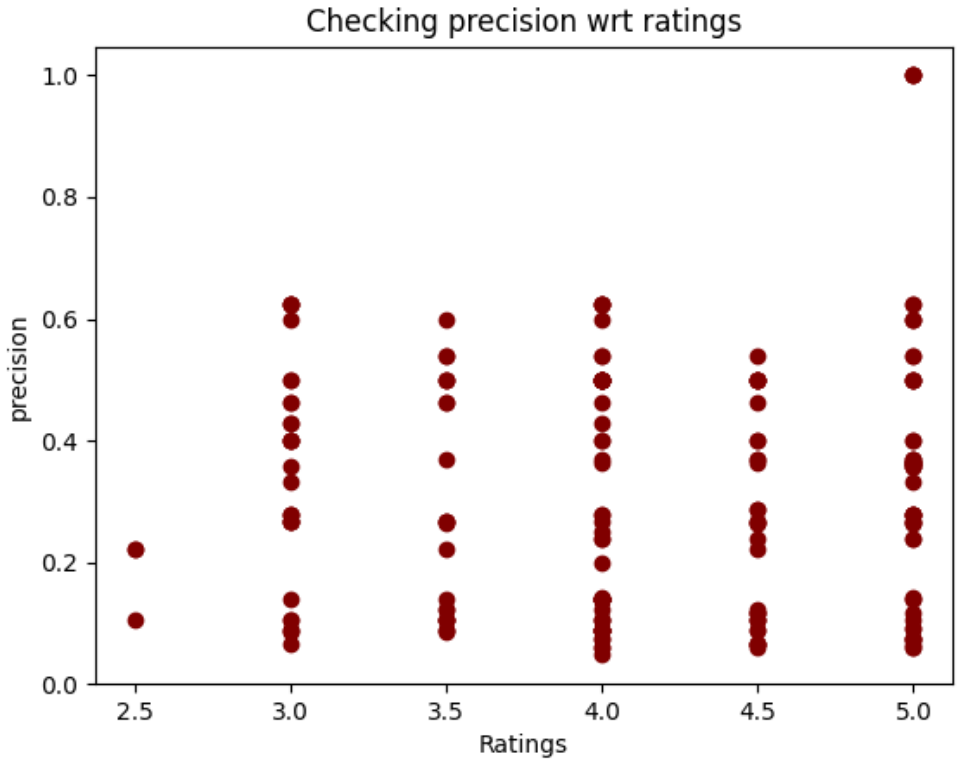
We started off by sorting the entire set of association rules by confidence and support. Then for each user, for each movie of each user in the train dataset, we find the association rules involving it. The consequents of these association rules are put in a union set for each user.

Alongside, we also cumulate the test dataset into user, and movie list. For each user, taking the intersection of the union set and the test movie list gives us the hit set.
For each user, we also calculate the precision and recall as per the formulas in analysisrecsys.pdf.
The average precision and recall are calculated as average over these.

The following graphs display the rating and recall and rating and precision distributions respectively.

The Average Precision came out to be 0.20417 and the recall is 0.11027





Optimisations:

Apriori is one of the slowest algorithms for frequent itemset mining. Using an FP growth algorithm or others such as Eclat or LCM could have made it 1000 times faster than Apriori. There also exist some fundamental problems in apriori, and so it doesn't make sense to try and optimize the algorithm. But, we did a few simple optimizations:

- We removed attributes that were not contributing to the algorithm, such as movie-title, tags and timestamp.
- The dataframe generated by Apriori will always keep a min-support value to get the frequent itemset.
- The k-th dataframe is made by joining two itemsets that belong to the (k-1)th frequent itemsets.
- Although it would have been better to keep a max length in the number of association rules generated, we did not do so, firstly, because it was a small dataset, which would be able to support apriori because it doesn't scale well, and also because we wanted to explore at most relationships as possible.