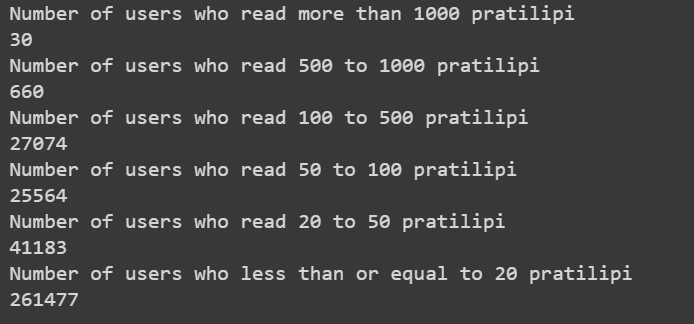
Questions and Answers

**Q1. Documentation explaining why you choose this model, explain alternatives and how the chosen approach is better than the alternatives**

Following is the distribution of Pratilipis read by a user

(Information from data\_exploration.ipynb)



There are 27,704 users who have read more than 100 Pratilipis. These users are suitable for collaborative filtering based Recommendation System as they have few features.

66,747 users who have read between 20 and 100 Pratilipis, don’t have many datapoints, thus content based recommendation system are suitable for them.

3,02,660 users who have read below 20 Pratilipis are suitable for recommending mixture of most popular content.

I have two solutions:

1. Content-based Recommendation System:
   1. Recommend old users based on similar content they have watched
   2. Built Pratilipi features composing of category and read time
   3. Omitted author in Pratilipi features as it would have been too sparse, but it can be included in future
   4. Built user feature as average( read\_percent \* Pratilipi feature for all Pratilipis read)
   5. Built score matrix which is user matrix \* Pratilipi feature matrix
   6. Score matrix was not fitting memory, thus used pickle to store it in external memory and process user feature one at a time
   7. Recommends most popular content if a new user is encountered
2. Collaborative filtering-based Recommendation System:
   1. Recommends Pratilipis to old users based on similarity to other users
   2. A user is likely to like a content if other user with similar taste also likes the content
   3. Built user-based recommendation system
   4. Built pivot matrix and stored it in pickle for reading and processing in parts
   5. Built similarity and stored it in pickle for reading and processing in parts
   6. Found ‘n’ users closest to a given user and recommended their top movies to the user

Following answer covers alternative solutions

**Q 2. Improvements to the model built**

1. The threshold to decide which user to be recommended Pratilipis from collaborative filtering or content based can be experimented
2. Mixture and ratio of collaborative filtering and content based recommendation can also be given
3. Mixture and ratio of above solutions and popular Pratilipis can also be given.
4. Popular Pratilipis can be Trending Pratilipis in world(last week), Trending by region, most popular Pratilipis in world, most popular Pratilipis by region
5. Item-based collaborative filtering can be explored. It can be better than user-based collaborative filtering as user may have variety of taste but Pratilipis are less variant ( like a Pratilipi will have 2-3 genre categories, user can have 5-10 genre likings)
6. Parameters like score threshold, number of Pratilipis to recommend can be explored.
7. Different similarity calculating methods like SVD, truncated SVD, Pearson, Kendall, Spearman can be explored.
8. Recommendation based upon time of the day
9. Recommendations based upon special occasions like festival
10. Recommendations based upon recent news

**Q3. Technical challenges faced**

1. **Out of memory issue**: Due to huge amount of data, it was not possible to process huge matrix like pivot matrix, similarity matrix(collaborative) score matrix (content).

I tried the following ways to deal with it:

1. SQL: was unable to store matrix as number of columns are limited
2. CSV: was able to store matrix but it was too slow to read and write
3. Pickle: was able to store matrix in separate vectors. Variables where stored as Python objects so no need to convert it unlike in SQL and CSV
4. HDF5: still to be explored
5. **Huge processing time:** I was unable to calculate accuracy on the whole dataset due to this reason. I have set variable in config to limit the train and test data size. Given enough time, it will be able to process and calculate accuracy without any bugs.

I was able to optimize to a degree by loading chunks of matrix (row-wise) for matrix multiplication rather than single rows one by one.

**Q4. Accuracy**

As I mentioned in above point, due to Huge processing time**,** I was unable to calculate accuracy on the whole dataset due to this reason. I have set variable in config to limit the train and test data size.

**Given enough time**, it will be able to process and calculate accuracy **without any bugs**.

For limited data, I found,

(content-based recommendation for old user + popular items for new users)

to work better than

(user-based collaborative filtering for old users + popular items for new users)

This can be because the range of read date is small ( between 18 March 22 to 23 March, ie 5 days) and we don’t have long term interaction data points.

This can also indicate we don’t have enough data points for collaborative filtering and user similarity is not properly found.