=> To predict something and recomend it

Types of Recommendation Engine:

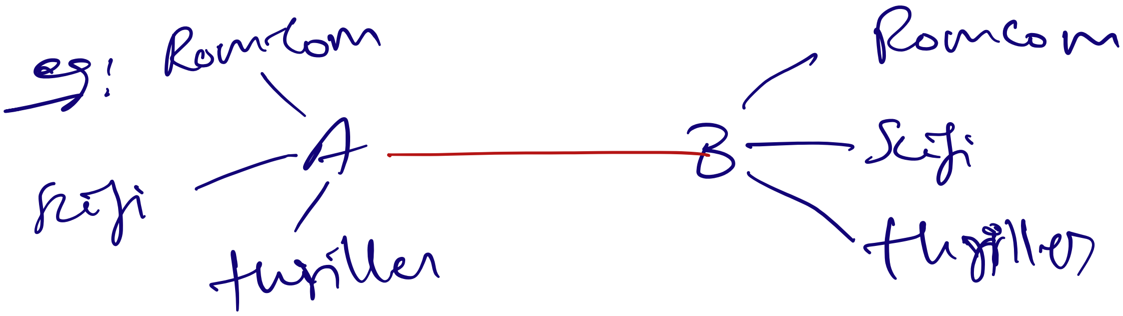
- Collaborative Filtering
- Content-Based Filtering
- Hybrid Recommendation Systems.

eg: Netflix

Collaborative filtering:

Filtering items from a large set of alternatives is done collaboratively by users preferences.

eg:



=> as they both share same interest, in future, they will be recommended similar movies...

⇒ If user A watch a new movie, then user (B) is also recommended same movie.

Colab filtering

→ User based colab filtering

① Identify similar users based on similar user preferences

② Recommend new items to an active user based on the items not rated by the active user.

→ item based colab filtering

① Calculate the item similarity based on the item preferences

② Find the top similar items to the non-rated items by active user and recommend them.

⇒ how do we measure the similarity?

Pearson Correlation

$$u_{ik} = \frac{\sum_{j} (v_{ij} - v_i)(v_{kj} - v_k)}{\sqrt{\sum_{j} (v_{ij} - v_i)^2 \sum_{j} (v_{kj} - v_k)^2}}$$

Cosine similarity

$$\cos(u_i, u_j) = \frac{\sum_{k=1}^{m} v_{ik} v_{jk}}{\sqrt{\sum_{k=1}^{m} v_{ik}^2 \sum_{k=1}^{m} v_{jk}^2}}$$

$$\boxed{v_{ij}^{*} = k \sum_{v_{kJ} \neq ?} u_{jk} v_{kj}}$$

Weighted Avg Movie Rating: !!

$$W = \frac{Rv + Cm}{v + m}$$

W= Weighted Rating

R= average for the movie as a number from 0 to 10 (mean) = (Rating)

V = number of votes for the movie = (votes)

m= min votes required to be listed in the Top 250 (currently 3000)

C = the mean vote across the whole report (currently 6.9)

MinMax Scaler:

→ Transform features by scaling each feature to a given range.

→ This estimator scales and translates each feature individually such that it is in the given range on the training set.

eg 0 - 1

1.

$$X\_std = (X - X.min(axis=0)) / (X.max(axis=0) - X.min(axis=0))$$

2.

$$X\_scaled = X\_std * (max - min) + min$$

where min, max = feature_range.

note!

this transformation is always used as an alternative to

mean=0,
Variance=1

note! Flow of project.

1. Analyze data
2. to merge bothe files use id as Common column
3. drop few unnecessary columns
4. Calculalte weighted average
5. check for popularity
6. Take 50% wt Avg and 50% pop
7. Sort them in desending order
8. it will recommend the movies acc to this filter.

Collab

Recommendation using basic python:

1. we dont use any formula to find the best movie.
2. on the basis of Correlation, we can observe the movies highly rated with respect to no.of votings.
3. This gives us correlation table and we then we can sort the movies in descending order basing on

the correlation values.

Recommendation Using KNN: !!

1. Nearest Neighbors item based on Collaborative filtering.
2. we have two files movies and rating. so, we merge them using Id
3. drop the null rows
4. we create a pivot matrix for all movies
5. we make a matrix called csr_matrix
6. importing nearest neighbors, we apply Cosine rule
7. we check the angle and we find the similarity b/w the movies and recommend them accordingly.

$$\mathrm{sim}(A,B) = \cos(\theta) = \frac{A \cdot B}{\left|\,|A|\,|B|\,\right|}$$

Book Recommendation System using pearson Correlation!

1. we have '3' csv files Books, Users, Ratings
2. first create a recommendation using ratings column by grouping 'ISBN' and 'bookRating'
3. sort the resultant
4. try doing with the help of Correlation.

⇒ we use pearson's Correlation Coefficient to measure the linear correlation b/w two variables, in our case, the ratings of two books

⇒ first, find out the avg rating and the no. of ratings each book received.