

Major Project Report

Name: Anshul Thakur, Dudhani Omkar Sanjay, Anupam Verma (B21EE007)

Roll Number: B21CS085, B21EE, B21EE007

Implementation #1(Item based Collaborative Filtering)

In this implementation of recommendation systems, instead of finding the relationships between users , movies are compared with each other.

In user based recommendation systems, habits of users can be changed. This situation makes it hard to recommend. However, in item based recommendation systems, movies or stuff does not change. Therefore recommendation is easier.

It has a few advantages over user based recommendation as there are many people in the input dataset. Comparing people increases the computational load. However, if items in this case movies are compared, less computational power is required.

In item based recommendation systems, we need to make user vs items/ movies matrices. In which we calculate similarity between columns that are movies or items.

Let us say there are two movies called A and B. Now there may be similarities between the two as they both are liked by three different people. This is the similarity point between these two movies.

If this similarity is high enough, we can recommend A to other people who have watched B or vice versa.

```
[47] movie01 = return_item_based_recommended("Toy Story (1995)", 10)
```

movie01

title	
Toy Story (1995)	1.000000
My Mother's Courage (Mutters Courage) (1995)	0.647453
Halfmoon (Paul Bowles - Halbmond) (1995)	0.556726
New York Cop (Nyû Yôku no koppu) (1993)	0.545606
Windows (1980)	0.541223
Gospa (1995)	0.516264
Shadows (Cienie) (1988)	0.508859
Happiness Is in the Field (Bonheur est dans le pré, Le) (1995)	0.495360
Eye of Vichy, The (Oeil de Vichy, L') (1993)	0.490778
Girl in the Cadillac (1995)	0.477885

dtype: float64

Implementation #2

Implementation #2(Item based using KNN)

In this implementation we first reduced the data set as KNN could not be applied on a dataset so big.

We then created The Csr matrix to reduce the sparsity of data because we filled the data with 0s.

Then we simply took the movie name as an input in the recommend method of our RecommendationSystem2 Class, which we searched in the data set. We then trained the KNN model on our csr matrix that has been created from data. And find the n_movies nearest neighbors.

```
[26] RS2.recommend('Toy Story')

['Jumanji (1995)',
 'Sabrina (1995)',
 'Father of the Bride Part II (1995)',
 'Babe (1995)',
 'Grumpier Old Men (1995)',
 'Heat (1995)',
 'Ace Ventura: When Nature Calls (1995)',
 'Clueless (1995)',
 'Leaving Las Vegas (1995)']
```

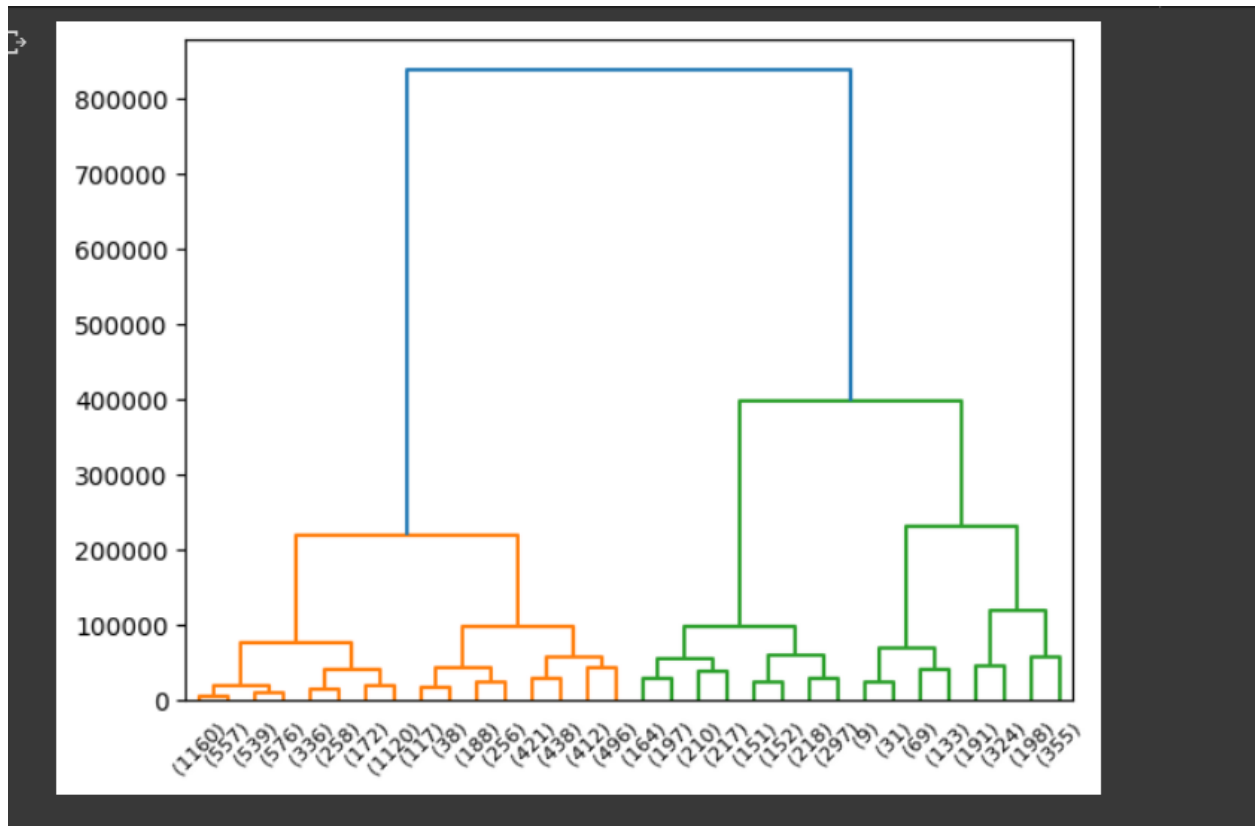
Implementation #3(User based using Agglomerative Clustering)

In This section we are clustering users based on their similarity using bottom - up agglomerative clustering.

The number of clusters is determined by the dendrogram analysis.

In our case the optimal count of clusters was 6.

Dendrogram :



Using this we predict in which cluster the new user will belong and hence give recommendation based upon the cluster members.

Contributions:

Anshul Thakur (B21CS085):

Made the Second Recommendation System Model (KNN)

Dudhani Omkar Sanjay (B21EE017):

Made the third recommendation system model (Agglomerative Clustering)

Anupam Verma:

Made the first recommendation system model (Item based Collaborative)