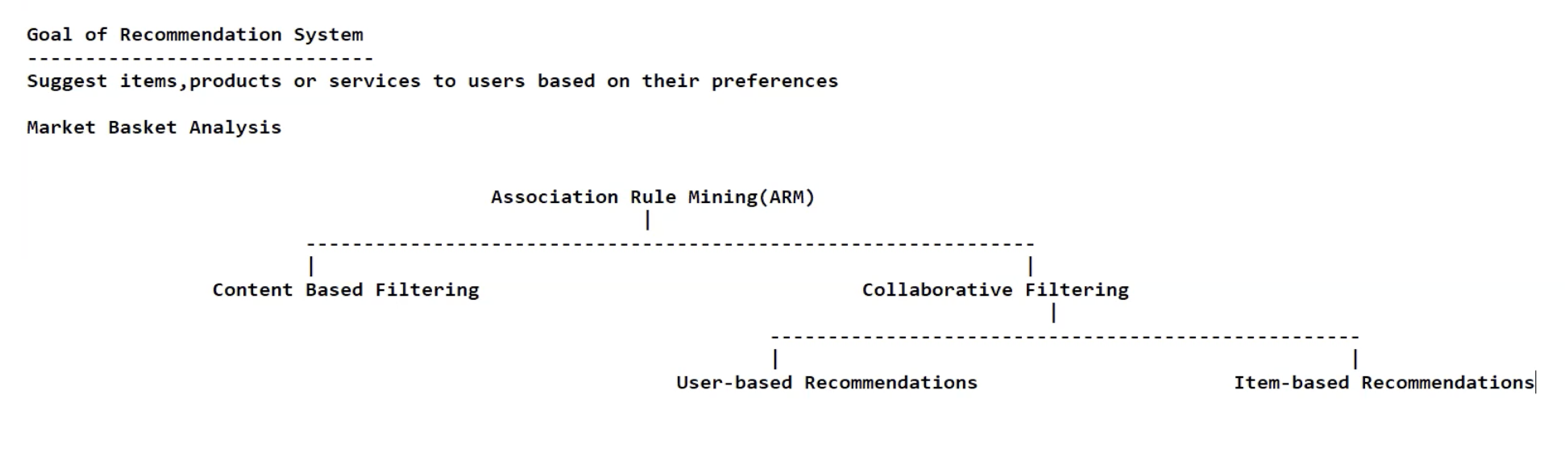
Introduction to Recommendation System

=================================

Association Rule Mining(ARM)



When we are talking about Content Based Filtering, is purely domain based approach. Anyone who is an expert in a domain details will suggest product replacement strategy.

User-Based Recommendation:

You will be using a panda dataframe as a model. The row index is movie\_id and the column index is User\_Id and the content will be the ratings. This dataframe called as Association Matrix.

Create a market basket data analysis:

Row Index: Products

Column Index: Users

Content: any entity.

To identify the similar products: 🡺 we use as pairwise\_distances formula.

Create a model with pairwise similarity:

from sklearn.metrics import pairwise\_distances

movie\_similarity = 1 - pairwise\_distances(np.matrix(learningMatrix.values) , metric= cosine)

#FinalKnowledgeBase

np.fill\_diagonal(movie\_similarity, 0)

ratings\_matrix = pd.DataFrame(movie\_similarity)

ratings\_matrix.head()

|  | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **...** | **1672** | **1673** | **1674** | **1675** | **1676** | **1677** | **1678** | **1679** | **1680** | **1681** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0.000000 | 0.402382 | 0.330245 | 0.454938 | 0.286714 | 0.116344 | 0.620979 | 0.481114 | 0.496288 | 0.273935 | ... | 0.035387 | 0.0 | 0.000000 | 0.000000 | 0.035387 | 0.0 | 0.0 | 0.0 | 0.047183 | 0.047183 |
| 1 | 0.402382 | 0.000000 | 0.273069 | 0.502571 | 0.318836 | 0.083563 | 0.383403 | 0.337002 | 0.255252 | 0.171082 | ... | 0.000000 | 0.0 | 0.000000 | 0.000000 | 0.000000 | 0.0 | 0.0 | 0.0 | 0.078299 | 0.078299 |
| 2 | 0.330245 | 0.273069 | 0.000000 | 0.324866 | 0.212957 | 0.106722 | 0.372921 | 0.200794 | 0.273669 | 0.158104 | ... | 0.000000 | 0.0 | 0.000000 | 0.000000 | 0.032292 | 0.0 | 0.0 | 0.0 | 0.000000 | 0.096875 |
| 3 | 0.454938 | 0.502571 | 0.324866 | 0.000000 | 0.334239 | 0.090308 | 0.489283 | 0.490236 | 0.419044 | 0.252561 | ... | 0.000000 | 0.0 | 0.094022 | 0.094022 | 0.037609 | 0.0 | 0.0 | 0.0 | 0.056413 | 0.075218 |
| 4 | 0.286714 | 0.318836 | 0.212957 | 0.334239 | 0.000000 | 0.037299 | 0.334769 | 0.259161 | 0.272448 | 0.055453 | ... | 0.000000 | 0.0 | 0.000000 | 0.000000 | 0.000000 | 0.0 | 0.0 | 0.0 | 0.000000 | 0.094211 |

How to use the model:

try:

userInput = input("Enter Movie Name: ")

movieIndex = movie[movie['movie\_title'] == userInput].index.tolist()[0]

movie['similarity'] = ratings\_matrix.iloc[movieIndex]

print("Recommended Top 10 Movies: ", movie.sort\_values(['similarity'], ascending=True)[1:10])

except:

print(movie.head(10))

Enter Movie Name: ra one

movie\_id movie\_title release date \

0 1 Toy Story (1995) 01-Jan-1995

1 2 GoldenEye (1995) 01-Jan-1995

2 3 Four Rooms (1995) 01-Jan-1995

3 4 Get Shorty (1995) 01-Jan-1995

4 5 Copycat (1995) 01-Jan-1995

5 6 Shanghai Triad (Yao a yao yao dao waipo qiao) ... 01-Jan-1995

6 7 Twelve Monkeys (1995) 01-Jan-1995

7 8 Babe (1995) 01-Jan-1995

8 9 Dead Man Walking (1995) 01-Jan-1995

9 10 Richard III (1995) 22-Jan-1996

video\_release\_date IMDb URL \

0 NaN [http://us.imdb.com/M/title-exact?Toy%20Story%2](http://us.imdb.com/M/title-exact?Toy%20Story%252)...

1 NaN <http://us.imdb.com/M/title-exact?GoldenEye%20(>...

2 NaN [http://us.imdb.com/M/title-exact?Four%20Rooms%](http://us.imdb.com/M/title-exact?Four%20Rooms%25)...

3 NaN [http://us.imdb.com/M/title-exact?Get%20Shorty%](http://us.imdb.com/M/title-exact?Get%20Shorty%25)...

4 NaN <http://us.imdb.com/M/title-exact?Copycat%20(1995>)

5 NaN <http://us.imdb.com/Title?Yao+a+yao+yao+dao+wai>...

6 NaN <http://us.imdb.com/M/title-exact?Twelve%20Monk>...

7 NaN <http://us.imdb.com/M/title-exact?Babe%20(1995>)

8 NaN <http://us.imdb.com/M/title-exact?Dead%20Man%20>...

9 NaN <http://us.imdb.com/M/title-exact?Richard%20III>...

unknown Action Adventure Animation Children ... Film-Noir Horror \

...

8 0 0 0 0 0 0 0 0.272448

9 0 0 0 0 0 1 0 0.055453

[10 rows x 25 columns]