

## **PROJECT: movie recommender system**

- **What Is a Recommendation Engine?**

- A recommendation engine is a type of data filtering tool using machine learning algorithms to recommend the most relevant items to a particular user or customer. It operates on the principle of finding patterns in consumer behavior data, which can be collected implicitly or explicitly.
- Netflix uses a recommendation engine to present viewers with movie and show suggestions. Amazon, on the other hand, uses a recommendation engine to present customers with product recommendations. While each uses one for slightly different purposes, both have the same goal: to drive sales, boost engagement and retention, and deliver more personalized customer experiences

- **What are the Types of Recommendation Engines?**

There are main types of recommendation engines: collaborative filtering, content-based filtering – and a hybrid of the two

- ***collaborative filtering***

Collaborative filtering focuses on collecting and analyzing data on user behavior, activities, and preferences, to predict what a person will like, based on their similarity to other users.

To plot and calculate these similarities, collaborative filtering uses a matrix style formula. An advantage of collaborative filtering is that it doesn't need to analyze or understand the content (products, films,

books). It simply picks items to recommend based on what they know about the user

- ***content based filtering***

Content-based filtering works on the principle that if you like a particular item, you will also like this other item. To make recommendations, algorithms use a profile of the customer's preferences and a description of an item (genre, product type, color, word length) to work out the similarity of items using cosine and Euclidean distances.

The downside of content-based filtering is that the system is limited to recommending products or content similar to what the person is already buying or using. It can't go beyond this to recommend other types of products or content. For example, it couldn't recommend products beyond homeware if the customer had only brought homeware

- ***Hybrid model***

A hybrid recommendation engine looks at both the meta (collaborative) data and the transactional (content-based) data. Because of this, it outperforms both.

In a hybrid recommendation engine, natural language processing tags can be generated for each product or item (movie, song), and vector equations used to calculate the similarity of products. A collaborative filtering matrix can then be used to recommend items to users depending on their behaviors, activities, and preferences. Netflix is the perfect example of a hybrid recommendation engine. It takes into account both the interests of the user (collaborative) and the descriptions or features of the movie or show (content-based)

Sql--<https://www.guru99.com/sql-interview-questions-answers.html>

1. Acid properties in dbms:-

2. Structure

3. Pointers

4. Transpose matrix

```
5. import numpy
6. matrix=[[1,2,3],[4,5,6]]
7. print(matrix)
8. print("\n")
9. print(numpy.transpose(matrix))
```

```
import numpy as np
matrix = np.array([[1,2,3],[4,5,6]])
print(matrix)
print("\n")
print(matrix.T)
```

```
A = [ [ 2, 1, 3], [ 3, 1, 5] ]

#initialize B with size of A transpose
B = [[0 for j in range(len(A))] for i in range(len(A[0]))]

for j in range(len(A)):
    for i in range(len(A[0])):
        B[i][j] = A[j][i]

print(' A:', A)
print(' B:', B)
```

```
x = [[12,7],
     [4,5],
```

```

[3, 8]]

result = [[0, 0, 0],
          [0, 0, 0]]

# iterate through rows
for i in range(len(X)):
    # iterate through columns
    for j in range(len(X[0])):
        result[j][i] = X[i][j]

for r in result:
    print(r)

```

10. java vs python
11. why structure and classes are not same?
12. C vs c++
13. Palindrome program
14. Disadvantages of pointers.
15. Ceo of cognizant
16. Difference between iot and iiot
17. Architecture of iot.
18. Various datatypes in c++