# Experiment 8

## Aim:

Recommendation system using Machine Learning.

## About Dataset:

The dataset used for building the recommendation system is anime dataset.
This data set contains information on user preference data from 73,516 users on 12,294 anime. Each user is able to add anime to their completed list and give it a rating and this data set is a compilation of those ratings. There are various features such as user_id, anime_id, rating, name, genre etc.

## Theory:

What are Recommendation Systems?
A recommendation system is a subclass of information filtering system that seeks to predict the "rating" or "preference" a user would give to an item. Recommender systems are utilised in a variety of areas, with commonly recognized examples taking the form of playlist generators for video and music services, product recommenders for online stores, or content recommenders for social media platforms and open web content recommenders. These systems can operate using a single input, like music, or multiple inputs within and across platforms like news, books, and search queries.
There are generally two types of recommendation systems:

Content-based recommendation systems: These systems recommend items similar to what a user has liked in the past. The system uses the characteristics or features of the items to make recommendations. For example, if a user has liked action movies in the past, the system will recommend action movies that share similar characteristics.

Collaborative filtering recommendation systems: These systems recommend items based on the past behaviour of similar users. The system uses the history of user-item interactions to identify patterns and similarities among users, and then uses this information to make recommendations. For example, if a user has similar preferences as another user who has liked a particular item, the system will recommend that item to the user.

In Data Mining, similarity measure refers to distance with dimensions representing features of the data object, in a dataset. If this distance is less, there will be a high degree of similarity, but when the distance is large, there will be a low degree of similarity.

Some of the popular similarity measures are –

1. Euclidean Distance.
2. Manhattan Distance.
3. Jaccard Similarity.
4. Minkowski Distance.
5. Cosine Similarity.

Cosine similarity is a metric, helpful in determining, how similar the data objects are irrespective of their size. We can measure the similarity between two sentences in Python using Cosine Similarity. In cosine similarity, data objects in a dataset are treated as a vector. The formula to find the cosine similarity between two vectors is –

$\text{Cos}(x, y) = x \cdot y / ||x|| * ||y||$

## Output:

In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
df_anime = pd.read_csv('anime.csv')
df_anime.head()
```

Out[2]:

| | anime_id | name | genre | type | episodes | rating | members |
|---|---|---|---|---|---|---|---|
| 0 | 32281 | Kimi no Na wa. | Drama, Romance, School, Supernatural | Movie | 1 | 9.37 | 200630 |
| 1 | 5114 | Fullmetal Alchemist: Brotherhood | Action, Adventure, Drama, Fantasy, Magic, Mili... | TV | 64 | 9.26 | 793665 |
| 2 | 28977 | Gintama° | Action, Comedy, Historical, Parody, Samurai, S... | TV | 51 | 9.25 | 114262 |
| 3 | 9253 | Steins;Gate | Sci-Fi, Thriller | TV | 24 | 9.17 | 673572 |
| 4 | 9969 | Gintama' | Action, Comedy, Historical, Parody, Samurai, S... | TV | 51 | 9.16 | 151266 |

In [3]:
```
df_anime.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12294 entries, 0 to 12293
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   anime_id  12294 non-null  int64
 1   name      12294 non-null  object
 2   genre     12232 non-null  object
 3   type      12269 non-null  object
 4   episodes  12294 non-null  object
 5   rating    12064 non-null  float64
 6   members   12294 non-null  int64
dtypes: float64(1), int64(2), object(4)
memory usage: 672.5+ KB
```

In [4]:
```
df_anime['episodes'].unique()
```

Out[4]:
```
array(['1', '64', '51', '24', '10', '148', '110', '13', '201', '25', '22',
       '75', '4', '26', '12', '27', '43', '74', '37', '2', '11', '99',
       'Unknown', '39', '101', '47', '50', '62', '33', '112', '23', '3',
       '94', '6', '8', '14', '7', '40', '15', '203', '77', '291', '120',
       '102', '96', '38', '79', '175', '103', '70', '153', '45', '5',
       '21', '63', '52', '28', '145', '36', '69', '60', '178', '114',
       '35', '61', '34', '109', '20', '9', '49', '366', '97', '48', '78',
       '358', '155', '104', '113', '54', '167', '161', '42', '142', '31',
       '373', '220', '46', '195', '17', '1787', '73', '147', '127', '16',
       '19', '98', '150', '76', '53', '124', '29', '115', '224', '44',
       '58', '93', '154', '92', '67', '172', '86', '30', '276', '59',
       '72', '330', '41', '105', '128', '137', '56', '55', '65', '243',
       '193', '18', '191', '180', '91', '192', '66', '182', '32', '164',
       '100', '296', '694', '95', '68', '117', '151', '130', '87', '170',
       '119', '84', '108', '156', '140', '331', '305', '300', '510',
       '200', '88', '1471', '526', '143', '726', '136', '1818', '237',
       '1428', '365', '163', '283', '71', '260', '199', '225', '312',
       '240', '1306', '1565', '773', '1274', '90', '475', '263', '83',
       '85', '1006', '80', '162', '132', '141', '125'], dtype=object)
```

In [5]:
```
df_anime['episodes'] = df_anime['episodes'].replace('Unknown',None)
```

In [6]:
```
df_anime['episodes'].dropna(inplace=True)
```

In [7]:
```
df_anime['episodes'] = pd.to_numeric(df_anime['episodes'])
```

In [8]:
```
df_anime.head()
```

Out[8]:

| | anime_id | name | genre | type | episodes | rating | members |
|---|---|---|---|---|---|---|---|
| 0 | 32281 | Kimi no Na wa. | Drama, Romance, School, Supernatural | Movie | 1.0 | 9.37 | 200630 |
| 1 | 5114 | Fullmetal Alchemist: Brotherhood | Action, Adventure, Drama, Fantasy, Magic, Mili... | TV | 64.0 | 9.26 | 793665 |
| 2 | 28977 | Gintama° | Action, Comedy, Historical, Parody, Samurai, S... | TV | 51.0 | 9.25 | 114262 |
| 3 | 9253 | Steins;Gate | Sci-Fi, Thriller | TV | 24.0 | 9.17 | 673572 |
| 4 | 9969 | Gintama' | Action, Comedy, Historical, Parody, Samurai, S... | TV | 51.0 | 9.16 | 151266 |

In [9]:
```
df_anime.dropna(inplace=True)
```

In [10]:
```
df_anime['type'].unique()
```

Out[10]:
```
array(['Movie', 'TV', 'OVA', 'Special', 'Music', 'ONA'], dtype=object)
```

In [11]:
```
df_anime.members.max()
```

Out[11]:
```
1013917
```

In [12]:
```
df_anime.sort_values(by=['members'],ascending=False, inplace=True)
```

In [13]:
```
df_anime.reset_index(inplace=True, drop=True)
```

In [14]:
```
df_anime = df_anime[:100]
```

In [15]:
```
df_anime
```

Out[15]:

| | anime_id | name | genre | type | episodes | rating | members |
|---|---|---|---|---|---|---|---|
| **0** | 1535 | Death Note | Mystery, Police, Psychological, Supernatural, ... | TV | 37.0 | 8.71 | 1013917 |
| **1** | 16498 | Shingeki no Kyojin | Action, Drama, Fantasy, Shounen, Super Power | TV | 25.0 | 8.54 | 896229 |
| **2** | 11757 | Sword Art Online | Action, Adventure, Fantasy, Game, Romance | TV | 25.0 | 7.83 | 893100 |
| **3** | 5114 | Fullmetal Alchemist: Brotherhood | Action, Adventure, Drama, Fantasy, Magic, Mili... | TV | 64.0 | 9.26 | 793665 |
| **4** | 6547 | Angel Beats! | Action, Comedy, Drama, School, Supernatural | TV | 13.0 | 8.39 | 717796 |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **95** | 270 | Hellsing | Action, Horror, Seinen, Supernatural, Vampire | TV | 13.0 | 7.64 | 308995 |
| **96** | 1887 | Lucky☆Star | Comedy, Parody, School, Slice of Life | TV | 24.0 | 7.87 | 305837 |
| **97** | 227 | FLCL | Action, Comedy, Dementia, Mecha, Parody, Sci-Fi | OVA | 6.0 | 8.06 | 305165 |
| **98** | 15583 | Date A Live | Comedy, Harem, Mecha, Romance, School, Sci-Fi | TV | 12.0 | 7.54 | 301358 |
| **99** | 6347 | Baka to Test to Shoukanjuu | Comedy, Romance, School, Super Power | TV | 13.0 | 7.83 | 301282 |

100 rows × 7 columns

In [16]:
```
index = df_anime.anime_id
```

In [17]:
```
ratings = pd.read_csv("rating.csv")
ratings
```

Out[17]:

| | user_id | anime_id | rating |
|---|---|---|---|
| **0** | 1 | 20 | -1 |
| **1** | 1 | 24 | -1 |
| **2** | 1 | 79 | -1 |
| **3** | 1 | 226 | -1 |
| **4** | 1 | 241 | -1 |
| **...** | ... | ... | ... |
| **7813732** | 73515 | 16512 | 7 |
| **7813733** | 73515 | 17187 | 9 |
| **7813734** | 73515 | 22145 | 10 |
| **7813735** | 73516 | 790 | 9 |
| **7813736** | 73516 | 8074 | 9 |

7813737 rows × 3 columns

In [18]:
```
ratings.drop(ratings[ratings['rating']<1].index, inplace=True)
```

In [19]:
```
df_ratings = pd.DataFrame(columns=ratings.columns)
df_ratings = df_ratings.append({'user_id':-1, 'anime_id':-1, 'rating':-1}, ignore_index=True)
```

In [20]:
```
df_ratings
```

Out[20]:

| | user_id | anime_id | rating |
|---|---|---|---|
| **0** | -1 | -1 | -1 |

In [21]:
```
for i in index:
    df_ratings = pd.concat([df_ratings, ratings[ratings['anime_id']==i]], axis=0)
```

In [33]:
```
df_ratings.head()
```

Out[33]:

| | user_id | anime_id | rating |
|---|---|---|---|
| **7813736** | 73516 | 8074 | 9 |
| **7813608** | 73515 | 1482 | 8 |
| **7813571** | 73515 | 356 | 8 |
| **7813567** | 73515 | 270 | 9 |
| **7813647** | 73515 | 3588 | 9 |

In [23]:
```
df_ratings.drop([0], inplace=True)
```

In [34]:
```
df_ratings.head()
```

Out[34]:

| | user_id | anime_id | rating |
|---|---|---|---|
| **7813736** | 73516 | 8074 | 9 |
| **7813608** | 73515 | 1482 | 8 |
| **7813571** | 73515 | 356 | 8 |
| **7813567** | 73515 | 270 | 9 |
| **7813647** | 73515 | 3588 | 9 |

In [25]:
```
df_ratings.sort_values(by=['user_id'],ascending=False, inplace=True)
```

In [26]:
```
df_ratings_new = df_ratings[:100000]
df_ratings_new.anime_id.nunique()
```

Out[26]:
```
100
```

In [27]:
```
pivot = df_ratings_new.pivot_table(index='anime_id', columns='user_id', values='rating')
```

In [27]:
```
pivot = df_ratings_new.pivot_table(index='anime_id', columns='user_id', values='rating')
```

In [28]:
```
pivot = pivot.fillna(0)
pivot
```

Out[28]:

| user_id | 68060 | 68061 | 68062 | 68063 | 68064 | 68065 | 68067 | 68068 | 68069 | 68070 | ... | 73506 | 73507 | 73508 | 73509 | 73510 | 73511 | 73512 | 73513 | 73515 | 73516 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| anime_id | | | | | | | | | | | | | | | | | | | | | |
| 1 | 0.0 | 0.0 | 0.0 | 4.0 | 0.0 | 0.0 | 0.0 | 9.0 | 0.0 | 0.0 | ... | 0.0 | 9.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 9.0 | 10.0 | 0.0 |
| 20 | 8.0 | 0.0 | 0.0 | 10.0 | 7.0 | 0.0 | 0.0 | 6.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 30 | 0.0 | 0.0 | 9.0 | 0.0 | 0.0 | 0.0 | 0.0 | 10.0 | 0.0 | 0.0 | ... | 0.0 | 9.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 8.0 | 0.0 |
| 121 | 0.0 | 0.0 | 0.0 | 0.0 | 8.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 9.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 8.0 | 0.0 |
| 164 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 8.0 | 0.0 | 0.0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 28223 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 28999 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 30276 | 0.0 | 0.0 | 0.0 | 0.0 | 9.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 31043 | 0.0 | 0.0 | 0.0 | 0.0 | 9.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 31240 | 0.0 | 0.0 | 0.0 | 0.0 | 8.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |

100 rows × 4910 columns

In [29]:
```
from sklearn.metrics.pairwise import cosine_similarity
```

In [30]:
```
similarity_scores = cosine_similarity(pivot)
similarity_scores.shape
```

Out[30]:
```
(100, 100)
```

In [31]:
```
###

def recommend(anime_name):
    # index fetch
    anime = df_anime.anime_id[df_anime['name'] == anime_name].values[0]
    index = np.where(pivot.index==anime)[0][0]
    similar_items = sorted(list(enumerate(similarity_scores[index])),key=lambda x:x[1],reverse=True)[1:6]

    data = []
    for i in similar_items:
        item = df_anime.name[df_anime['anime_id'] == pivot.index[i[0]]].values[0]
        data.append(item)

    return data
```

In [32]:
```
recommend('Fullmetal Alchemist: Brotherhood')
```

Out[32]:
```
['Shingeki no Kyojin',
 'Death Note',
 'Fullmetal Alchemist',
 'Code Geass: Hangyaku no Lelouch',
 'Steins;Gate']
```

# Conclusion:

Thus we studied, understood and built our recommendation system which will recommend us anime based on input provided by us.