# Movie Recommendation Using Collaborative filtering

In [120]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Data Preprocessing

In [228]:

```
id_df=pd.read_csv("C:\\Users\\Nelson\\Desktop\\machine-learning-ex\\u.data",names=['User_id','movie_id','Rating','Timestamp'],delimiter='\t')
```

In [85]:

```
id_df.drop(['Timestamp'],axis=1,inplace=True)
id_df
```

Out[85]:

| | User_id | movie_id | Rating |
|---|---|---|---|
| **0** | 196 | 242 | 3 |
| **1** | 186 | 302 | 3 |
| **2** | 22 | 377 | 1 |
| **3** | 244 | 51 | 2 |
| **4** | 166 | 346 | 1 |
| **...** | ... | ... | ... |
| **99995** | 880 | 476 | 3 |
| **99996** | 716 | 204 | 5 |
| **99997** | 276 | 1090 | 1 |
| **99998** | 13 | 225 | 2 |
| **99999** | 12 | 203 | 3 |

100000 rows × 3 columns

In [86]:

```
movie_rating=pd.read_csv("C:\\Users\\Nelson\\Desktop\\machine-learning-ex\\movierating.item",sep="|",usecols=range(2),encoding="ISO-8859-1",names=['movie_id','Tittle'])
```

In [230]:

```
recdf=pd.merge(movie_rating,id_df)
recdf.head()
```

Out[230]:

| | movie_id | Tittle | User_id | Rating | Timestamp |
|---|---|---|---|---|---|
| **0** | 1 | Toy Story (1995) | 308 | 4 | 887736532 |
| **1** | 1 | Toy Story (1995) | 287 | 5 | 875334088 |
| **2** | 1 | Toy Story (1995) | 148 | 4 | 877019411 |
| **3** | 1 | Toy Story (1995) | 280 | 4 | 891700426 |
| **4** | 1 | Toy Story (1995) | 66 | 3 | 883601324 |

In [234]:

```
rating=recdf.pivot_table(index=['User_id'],columns=['Tittle'],values='Rating')
rating.head()
```

Out[234]:

| Tittle | 'Til There Was You (1997) | 1-900 (1994) | 101 Dalmatians (1996) | 12 Angry Men (1957) | 187 (1997) | 2 Days in the Valley (1996) | 20,000 Leagues Under the Sea (1954) | 2001: A Space Odyssey (1968) | 3 Ninjas: High Noon At Mega Mountain (1998) | 39 Steps, The (1935) | ... | Yankee Zulu (1994) | Year of the Horse (1997) | You So Crazy (1994) | Young Frankens (1974) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| User_id | | | | | | | | | | | | | | | |
| 1 | NaN | NaN | 2.0 | 5.0 | NaN | NaN | 3.0 | 4.0 | NaN | NaN | ... | NaN | NaN | NaN | |
| 2 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | 1.0 | NaN | ... | NaN | NaN | NaN | N |
| 3 | NaN | NaN | NaN | NaN | 2.0 | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | N |
| 4 | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | ... | NaN | NaN | NaN | N |
| 5 | NaN | NaN | 2.0 | NaN | NaN | NaN | NaN | 4.0 | NaN | NaN | ... | NaN | NaN | NaN | |

5 rows × 1664 columns

The Pivot Table gives you a information about which user has rated which movie

In [132]:

```
StarWarsRating=rating['Star Wars (1977)']
```

Then we will be finding Correlation with respect to Star Wars(1997)

In [164]:

```
similarmovies=pd.DataFrame(rating.corrwith(StarWarsRating),columns=['Correlation'])
similarmovies.dropna(inplace=True)
```

In [170]:

```
sort=similarmovies.sort_values(by='Correlation',ascending=False)
sort
```

Out[170]:

| Tittle | Correlation |
|---|---|
| Hollow Reed (1996) | 1.0 |
| Commandments (1997) | 1.0 |
| Cosi (1996) | 1.0 |
| No Escape (1994) | 1.0 |
| Stripes (1981) | 1.0 |
| ... | ... |
| Roseanna's Grave (For Roseanna) (1997) | -1.0 |
| For Ever Mozart (1996) | -1.0 |
| American Dream (1990) | -1.0 |
| Frankie Starlight (1995) | -1.0 |
| Fille seule, La (A Single Girl) (1995) | -1.0 |

1410 rows × 1 columns

We will be selecting the movies which got rated by hundred users

In [197]:

```
status=recdf.groupby('Tittle').agg({'Rating':[np.size,np.mean]})
status.head()
```

Out[197]:

| | Rating | |
|---|---|---|
| | size | mean |
| Tittle | | |
| 'Til There Was You (1997) | 9 | 2.333333 |
| 1-900 (1994) | 5 | 2.600000 |
| 101 Dalmatians (1996) | 109 | 2.908257 |
| 12 Angry Men (1957) | 125 | 4.344000 |
| 187 (1997) | 41 | 3.024390 |

In [210]:

```
morethan100=status['Rating']['size']>=100
status[morethan100].sort_values([('Rating', 'mean')], ascending=False)[:15]
```

Out[210]:

| | Rating | |
|---|---|---|
| | size | mean |
| Tittle | | |
| Close Shave, A (1995) | 112 | 4.491071 |
| Schindler's List (1993) | 298 | 4.466443 |
| Wrong Trousers, The (1993) | 118 | 4.466102 |
| Casablanca (1942) | 243 | 4.456790 |
| Shawshank Redemption, The (1994) | 283 | 4.445230 |
| Rear Window (1954) | 209 | 4.387560 |
| Usual Suspects, The (1995) | 267 | 4.385768 |
| Star Wars (1977) | 583 | 4.358491 |
| 12 Angry Men (1957) | 125 | 4.344000 |
| Citizen Kane (1941) | 198 | 4.292929 |
| To Kill a Mockingbird (1962) | 219 | 4.292237 |
| One Flew Over the Cuckoo's Nest (1975) | 264 | 4.291667 |
| Silence of the Lambs, The (1991) | 390 | 4.289744 |
| North by Northwest (1959) | 179 | 4.284916 |
| Godfather, The (1972) | 413 | 4.283293 |

In [225]:

```
df=status[morethan100].join(sort)
df.head()
```

```
C:\Users\Nelson\anaconda3\lib\site-packages\pandas\core\reshape\merge.py:618: UserWarning: merging
between different levels can give an unintended result (2 levels on the left, 1 on the right)
  warnings.warn(msg, UserWarning)
```

Out[225]:

| (Rating |
|---|

| Tittle | (Rating, size) | (Rating, mean) | Correlation |
|---|---|---|---|
| 101 Dalmatians (1996) | 109 | 2.908257 | 0.211132 |
| 12 Angry Men (1957) | 125 | 4.344000 | 0.184289 |
| 2001: A Space Odyssey (1968) | 259 | 3.969112 | 0.230884 |
| Absolute Power (1997) | 127 | 3.370079 | 0.085440 |
| Abyss, The (1989) | 151 | 3.589404 | 0.203709 |

In [236]:

```
df=df.sort_values(by=['Correlation'],ascending=False)
df.head()
```

Out[236]:

| Tittle | (Rating, size) | (Rating, mean) | Correlation |
|---|---|---|---|
| Star Wars (1977) | 583 | 4.358491 | 1.000000 |
| Empire Strikes Back, The (1980) | 367 | 4.204360 | 0.747981 |
| Return of the Jedi (1983) | 507 | 4.007890 | 0.672556 |
| Raiders of the Lost Ark (1981) | 420 | 4.252381 | 0.536117 |
| Austin Powers: International Man of Mystery (1997) | 130 | 3.246154 | 0.377433 |

So we had built a recommender system to recommend movies to the user based on the past data