Problem Statement: -

The Entertainment Company, which is a startup online movie watching platform, wants to improvise its collection of movies and showcase those that are highly rated, and recommend those movies to its customer by their movie watching footprints. For this the company has collected its data and shared it with you to provide some analytical insights and also to come up with a Recommendation Algorithm so that it can automate its process for effective recommendations based on Users Interest and behavior patterns.

Objective :-

Recommend those movies to its customer by their movie watching footprints.

In [2]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt


# import Dataset
Entertainments = pd.read_csv("D:/360Digi/Entertainment.csv")
Entertainments.shape # shape
Entertainments.columns
Entertainments
```

Out[2]:

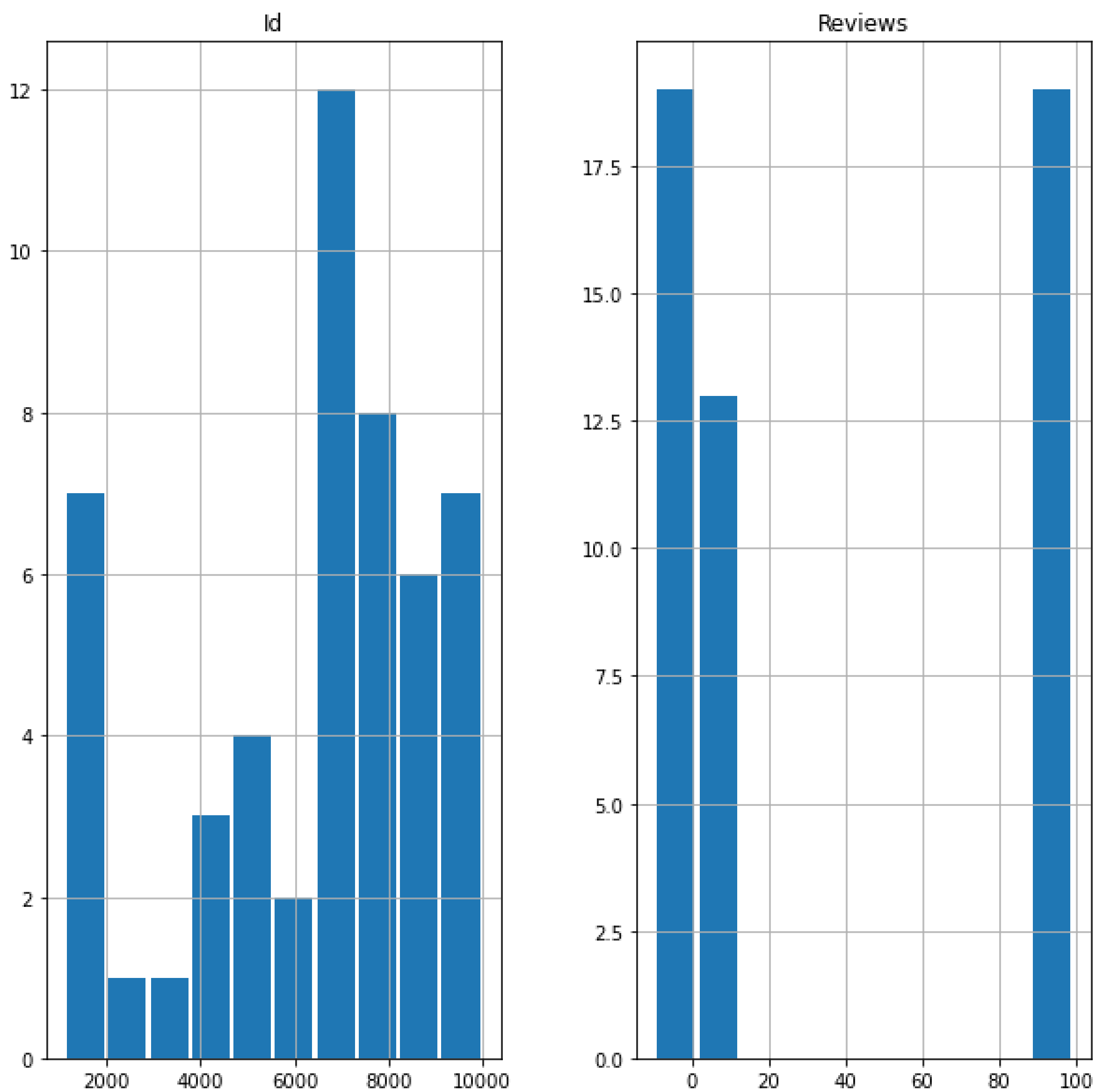
	Id	Titles	Category	Reviews
0	6973	Toy Story (1995)	Drama, Romance, School, Supernatural	-8.98
1	6778	Jumanji (1995)	Action, Adventure, Drama, Fantasy, Magic, Mili...	8.88
2	9702	Grumpier Old Men (1995)	Action, Comedy, Historical, Parody, Samurai, S...	99.00
3	6769	Waiting to Exhale (1995)	Sci-Fi, Thriller	99.00
4	1123	Father of the Bride Part II (1995)	Action, Comedy, Historical, Parody, Samurai, S...	-0.44
5	9860	Heat (1995)	Comedy, Drama, School, Shounen, Sports	-6.65
6	1803	Sabrina (1995)	Action, Adventure, Shounen, Super Power	99.00
7	9721	Tom and Huck (1995)	Drama, Military, Sci-Fi, Space	-5.19
8	6563	Sudden Death (1995)	Action, Comedy, Historical, Parody, Samurai, S...	-7.86
9	1323	GoldenEye (1995)	Action, Comedy, Historical, Parody, Samurai, S...	3.01
10	7889	American President, The (1995)	Drama, Fantasy, Romance, Slice of Life, Supern...	99.00
11	7650	Dracula: Dead and Loving It (1995)	Drama, School, Shounen	-7.23
12	5467	Balto (1995)	Action, Comedy, Historical, Parody, Samurai, S...	99.00
13	7451	Nixon (1995)	Action, Drama, Mecha, Military, Sci-Fi, Super ...	5.97
14	5898	Cutthroat Island (1995)	Comedy, Drama, School, Shounen, Sports	-5.92
15	7355	Casino (1995)	Adventure, Drama, Supernatural	2.91
16	6791	Sense and Sensibility (1995)	Drama, Music, Romance, School, Shounen	99.00
17	8574	Four Rooms (1995)	Adventure, Fantasy, Historical, Mystery, Seine...	0.53
18	9679	Ace Ventura: When Nature Calls (1995)	Fantasy, Slice of Life	99.00
19	5465	Money Train (1995)	Action, Mecha, Military, School, Sci-Fi, Super...	99.00

	Id	Titles	Category	Reviews
20	6478	Get Shorty (1995)	Comedy, Drama, Shounen, Sports	99.00
21	8670	Copycat (1995)	Action, Drama, Historical, Martial Arts, Roman...	99.00
22	6865	Assassins (1995)	Action, Adventure, Comedy, Drama, Sci-Fi, Space	6.17
23	1324	Powder (1995)	Action, Comedy, Parody, Sci-Fi, Seinen, Super ...	-8.20
24	6677	Leaving Las Vegas (1995)	Action, Adventure, Fantasy	-0.34
25	9657	Othello (1995)	Comedy, Mystery, Romance, School, Sci-Fi, Supe...	8.01
26	4654	Now and Then (1995)	Comedy, Mystery, Romance, Supernatural, Vampire	-3.98
27	1233	Persuasion (1995)	Adventure, Fantasy, Historical, Mystery, Seine...	-4.61
28	4647	City of Lost Children (1995)	Adventure, Fantasy, Historical, Mystery, Seine...	99.00
29	5871	Shanghai Triad (Yao a yao yao dao waipo qiao) ...	Action, Adventure, Comedy, Mecha, Sci-Fi	99.00
30	8754	Dangerous Minds (1995)	Comedy, Drama, School, Shounen, Slice of Life	-0.19
31	2797	Twelve Monkeys (a.k.a. 12 Monkeys) (1995)	Drama, Fantasy, Shoujo, Slice of Life, Superna...	99.00
32	6590	Babe (1995)	Comedy, Drama, Shounen, Sports	-1.41
33	5236	Dead Man Walking (1995)	Adventure, Fantasy, Historical, Mystery, Seine...	99.00
34	7547	It Takes Two (1995)	Drama, Fantasy, Shoujo, Slice of Life, Superna...	5.92
35	6698	Clueless (1995)	Adventure, Drama, Fantasy, Romance	99.00
36	5355	Cry, the Beloved Country (1995)	Action, Fantasy, Supernatural, Thriller	5.15
37	7589	Richard III (1995)	Action, Mystery, Supernatural, Vampire	99.00
38	7643	Dead Presidents (1995)	Drama, Horror, Mystery, Police, Psychological,...	2.96
39	6868	Restoration (1995)	Comedy, Drama, Romance, Shounen	99.00
40	4172	Mortal Kombat (1995)	Mystery, Police, Psychological, Supernatural, ...	-8.74
41	8558	To Die For (1995)	Comedy, Parody	99.00
42	9979	How to Make an American Quilt (1995)	Action, Comedy, School, Shounen	99.00
43	1110	Seven (a.k.a. Se7en) (1995)	Comedy, Drama, School, Shounen, Sports	-5.05
44	1264	Pocahontas (1995)	Comedy, Drama, Shounen, Sports	7.86
45	3469	When Night Is Falling (1995)	Action, Drama, Mystery, Romance, Supernatural,...	6.31
46	6439	Usual Suspects, The (1995)	Drama, Fantasy, Shoujo, Slice of Life, Superna...	-8.45

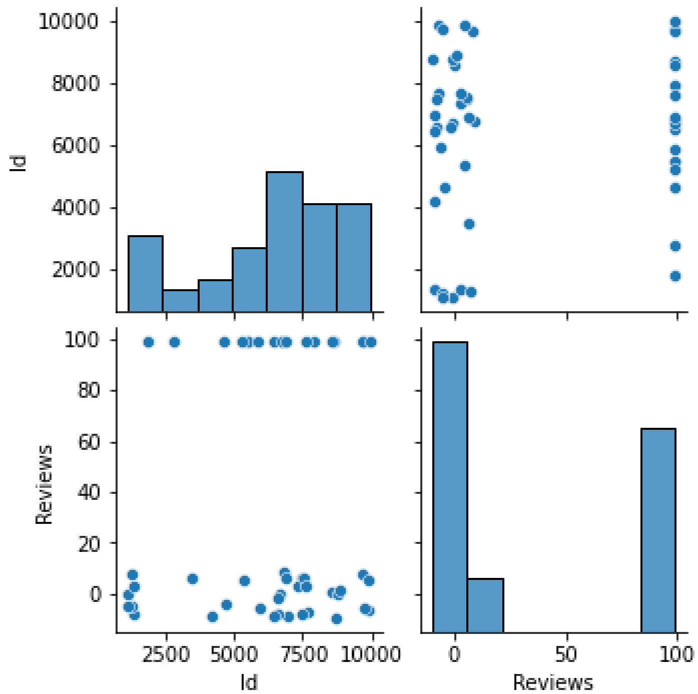
	Id	Titles	Category	Reviews
47	7464	Mighty Aphrodite (1995)	Psychological, Seinen, Sports	-7.86
48	8879	Lamerica (1994)	Adventure, Fantasy, Historical, Mystery, Seine...	1.46
49	8746	Big Green, The (1995)	Mystery, Psychological, Seinen, Supernatural	-9.42
50	9844	Georgia (1995)	Mystery, Psychological, Romance	5.15

In [3]:
```
Entertainments.hist(grid=True, rwidth=0.9, figsize=(10,10))
```

Out[3]:
```
array([[<AxesSubplot:title={'center':'Id'}>,
        <AxesSubplot:title={'center':'Reviews'}>]], dtype=object)
```



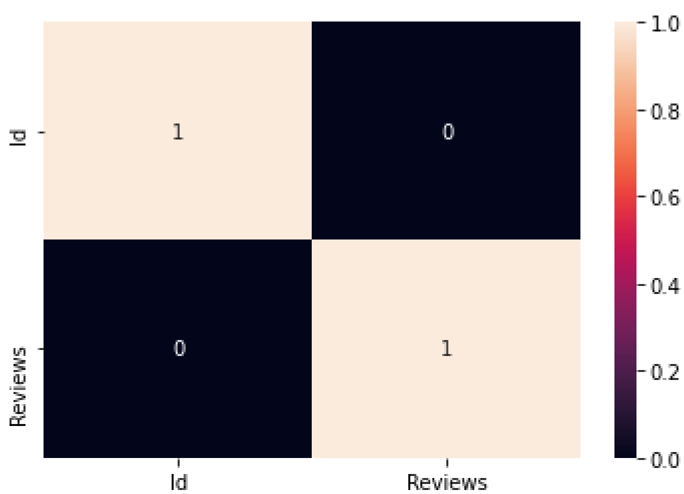
In [4]:
```
sns.pairplot(Entertainments)
plt.figure(figsize=(8,8))
plt.show()
```




```
<Figure size 576x576 with 0 Axes>
```

In [5]:
```
a = Entertainments.corr(method ='pearson')
sns.heatmap(a>0.85,annot=True)
```

Out[5]: `<AxesSubplot:>`



In [ ]:

In [16]:

```
from sklearn.feature_extraction.text import TfidfVectorizer #term frequencey- inv

# Creating a Tfidf Vectorizer to remove all stop words
tfidf = TfidfVectorizer(stop_words = "english")    # taking stop words from tfid

# replacing the NaN values in overview column with empty string
Entertainments["Titles"].isnull().sum()
Entertainments["Titles"] = Entertainments["Titles"].fillna(" ")
```

In [21]:

```
# Preparing the Tfidf matrix by fitting and transforming
tfidf_matrix = tfidf.fit_transform(Entertainments["Titles"])   #Transform a count
tfidf_matrix.shape
```

Out[21]: (51, 90)

In [25]:

```
from sklearn.metrics.pairwise import linear_kernel

# Computing the cosine similarity on Tfidf matrix
cosine_sim_matrix = linear_kernel(tfidf_matrix, tfidf_matrix)

# creating a mapping of movie name to index number
Entertainment_index = pd.Series(Entertainments.index, index = Entertainments["Tit
Entertainment_index = Entertainment_index[~Entertainment_index.index.duplicated(k

Entertainment_id = Entertainment_index["Othello (1995)"]
Entertainment_id
```

Out[25]: 25

In [28]:

```
def get_recommendations(Name, topN):
    # topN = 10
    # Getting the movie index using its title
    Entertainment_id = Entertainment_index[Name]

    # Getting the pair wise similarity score for all the anime's with that
    # anime
    cosine_scores = list(enumerate(cosine_sim_matrix[Entertainment_id]))

    # Sorting the cosine_similarity scores based on scores
    cosine_scores = sorted(cosine_scores, key=lambda x:x[1], reverse = True)

    # Get the scores of top N most similar movies
    cosine_scores_N = cosine_scores[0: topN+1]


# Getting the movie index
    Entertainment_idx  =  [i[0] for i in cosine_scores_N]
    Entertainment_scores =  [i[1] for i in cosine_scores_N]

    # Similar movies and scores
    Entertainment_similar_show = pd.DataFrame(columns=["Titles", "Score"])
    Entertainment_similar_show["name"] = Entertainments.loc[Entertainment_idx, "T
    Entertainment_similar_show["Score"] = Entertainment_scores
    Entertainment_similar_show.reset_index(inplace = True)
    # anime_similar_show.drop(["index"], axis=1, inplace=True)
    print (Entertainment_similar_show)
    # return (anime_similar_show)
```

In [29]:

```
# Enter your anime and number of anime's to be recommended
get_recommendations("Othello (1995)", topN = 10)
Entertainment_index["Othello (1995)"]
```

```
    index Titles     Score                name
0      25    NaN  1.000000      Othello (1995)
1      26    NaN  0.232828  Now and Then (1995)
2       1    NaN  0.054209      Jumanji (1995)
3       5    NaN  0.054209         Heat (1995)
4       6    NaN  0.054209      Sabrina (1995)
5       9    NaN  0.054209    GoldenEye (1995)
6      12    NaN  0.054209        Balto (1995)
7      13    NaN  0.054209        Nixon (1995)
8      15    NaN  0.054209       Casino (1995)
9      17    NaN  0.054209   Four Rooms (1995)
10     20    NaN  0.054209   Get Shorty (1995)
```

Out[29]: 25

In [ ]:

Summary

1- User based recommentation systems

2- Top 10 recommendation for movies are showed above for Othello (1995).

3- Item based recommentation systems

In [ ]:

```
```