Problem Statement:-

Build a recommender system by using cosine simillarties score

1.Import Necessary Libraries

In [3]:
```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
from sklearn.metrics import pairwise_distances
from scipy.spatial.distance import cosine, correlation
```

2. Import Data

In [4]:
```
Recc_Book = pd.read_csv('book (1).csv')
Recc_Book
```

Out[4]:

	Unnamed: 0	User.ID	Book.Title	Book.Rating
0	1	276726	Classical Mythology	5
1	2	276729	Clara Callan	3
2	3	276729	Decision in Normandy	6
3	4	276736	Flu: The Story of the Great Influenza Pandemic...	8
4	5	276737	The Mummies of Urumchi	6
...	...	...	...	...
9995	9996	162121	American Fried: Adventures of a Happy Eater.	7
9996	9997	162121	Cannibal In Manhattan	9
9997	9998	162121	How to Flirt: A Practical Guide	7
9998	9999	162121	Twilight	8
9999	10000	162129	Kids Say the Darndest Things	6

10000 rows × 4 columns

3. Data Understanding

In [5]:
```
Recc_Book.shape
```

Out[5]:
```
(10000, 4)
```

In [6]:
```
Recc_Book.dtypes
```

Out[6]:
```
Unnamed: 0      int64
User.ID         int64
Book.Title     object
Book.Rating     int64
dtype: object
```

In [7]:
```
Recc_Book.isna().sum()
```

Out[7]:
```
Unnamed: 0     0
User.ID        0
Book.Title     0
Book.Rating    0
dtype: int64
```

In [8]:
```
Recc_Book['User.ID'].nunique()
```

Out[8]:
```
2182
```

In [9]:
```
Recc_Book['User.ID']
```

Out[9]:
```
0       276726
1       276729
2       276729
3       276736
4       276737
         ...  
9995    162121
9996    162121
9997    162121
9998    162121
9999    162129
Name: User.ID, Length: 10000, dtype: int64
```

In [10]:
```
Recc_Book['Book.Title'].nunique()
```

Out[10]:
```
9659
```

In [11]:

```
Recc_Book['Book.Title']
```

Out[11]:

```
0                                      Classical Mythology
1                                             Clara Callan
2                                     Decision in Normandy
3        Flu: The Story of the Great Influenza Pandemic...
4                                   The Mummies of Urumchi
                              ...
9995         American Fried: Adventures of a Happy Eater.
9996                                 Cannibal In Manhattan
9997                       How to Flirt: A Practical Guide
9998                                              Twilight
9999                          Kids Say the Darndest Things
Name: Book.Title, Length: 10000, dtype: object
```

In [12]:

```
Recc_Book['Book.Rating']
```

Out[12]:

```
0       5
1       3
2       6
3       8
4       6
       ..
9995    7
9996    9
9997    7
9998    8
9999    6
Name: Book.Rating, Length: 10000, dtype: int64
```

In [13]:

```
Recc_Book['Book.Rating'].nunique()
```

Out[13]:

```
10
```

In [14]:

```
del Recc_Book['Unnamed: 0']
```

In [15]:
```
Recc_Book
```

Out[15]:

	User.ID	Book.Title	Book.Rating
0	276726	Classical Mythology	5
1	276729	Clara Callan	3
2	276729	Decision in Normandy	6
3	276736	Flu: The Story of the Great Influenza Pandemic...	8
4	276737	The Mummies of Urumchi	6
...	...	...	...
9995	162121	American Fried: Adventures of a Happy Eater.	7
9996	162121	Cannibal In Manhattan	9
9997	162121	How to Flirt: A Practical Guide	7
9998	162121	Twilight	8
9999	162129	Kids Say the Darndest Things	6

10000 rows × 3 columns

4. Data Preparation

In [16]:
```
Recc_Book.sort_values(['User.ID'])
```

Out[16]:

	User.ID	Book.Title	Book.Rating
2401	8	Wings	5
2400	8	The Western way: A practical guide to the West...	5
2399	8	Ancient Celtic Romances	5
2402	8	Truckers	5
2405	8	The Art Of Celtia	7
...	...	...	...
2395	278854	La crónica del Perú (Crónicas de América)	7
2398	278854	Celtic Mythology (Library of the World's Myths...	8
2393	278854	A corrente de Trewis Scott	7
2394	278854	As valkírias	7
2397	278854	A Treasury of Irish Myth, Legend, and Folklore	6

10000 rows × 3 columns

Converting long data into wide data using pivot table

In [17]:

```
Recc_Book_1 = Recc_Book.pivot_table(values='Book.Rating', index='User.ID', columr
Recc_Book_1
```

Out[17]:

Book.Title	Jason, Madison &amp	Other Stories;Merril;1985;McClelland &amp	Repairing PC Drives &amp	'48	'O Au No Keia: Voices from Hawai'I's Mahu and Transgender Communities	...AND THE HORSE HE RODE IN ON : THE PEOPLE V. KENNETH STARR	01-0 A Nov Millenr
User.ID							
8	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	
10	0.0	0.0	0.0	0.0	0.0	0.0	
12	0.0	0.0	0.0	0.0	0.0	0.0	
14	0.0	0.0	0.0	0.0	0.0	0.0	
...	...	...	...	...	...	...	
278846	0.0	0.0	0.0	0.0	0.0	0.0	
278849	0.0	0.0	0.0	0.0	0.0	0.0	
278851	0.0	0.0	0.0	0.0	0.0	0.0	
278852	0.0	0.0	0.0	0.0	0.0	0.0	
278854	0.0	0.0	0.0	0.0	0.0	0.0	

2182 rows × 9659 columns

Calculating cosine similarity between cosines of array data

In [18]:

```
User_recc = 1-pairwise_distances(Recc_Book_1.values, metric='cosine')
User_recc
```

Out[18]:

```
array([[1., 0., 0., ..., 0., 0., 0.],
       [0., 1., 0., ..., 0., 0., 0.],
       [0., 0., 1., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 1., 0., 0.],
       [0., 0., 0., ..., 0., 1., 0.],
       [0., 0., 0., ..., 0., 0., 1.]])
```

In [19]:
```
User_recc_1 = pd.DataFrame(User_recc)
User_recc_1
```

Out[19]:
```
        0    1    2    3    4    5    6    7    8    9  ...  2172  2173  2174  2175  2176  2177  21
   0  1.0  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
   1  0.0  1.0  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
   2  0.0  0.0  1.0  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
   3  0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  0.0  0.0  ...   0.0   0.0   0.0   0.0   0.0   0.0
   4  0.0  0.0  0.0  0.0  1.0  0.0  0.0  0.0  0.0  0.0  ...   0.0   0.0   0.0   0.0   0.0   0.0
 ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...   ...   ...   ...   ...   ...   ...
2177  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   1.0
2178  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
2179  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
2180  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
2181  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

2182 rows × 2182 columns
```

Set index & columns name to user id

In [20]:
```
User_recc_1.index = Recc_Book['User.ID'].unique()
User_recc_1.columns = Recc_Book['User.ID'].unique()
User_recc_1
```

Out[20]:
```
        276726  276729  276736  276737  276744  276745  276747  276748  276751  276754  ...  1
276726     1.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0  ...
276729     0.0     1.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0  ...
276736     0.0     0.0     1.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0  ...
276737     0.0     0.0     0.0     1.0     0.0     0.0     0.0     0.0     0.0     0.0  ...
276744     0.0     0.0     0.0     0.0     1.0     0.0     0.0     0.0     0.0     0.0  ...
   ...     ...     ...     ...     ...     ...     ...     ...     ...     ...     ...  ...
162107     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0  ...
162109     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0  ...
162113     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0  ...
162121     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0  ...
162129     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0  ...

2182 rows × 2182 columns
```

Nullifying Diagonal Values

In [21]:
```
np.fill_diagonal(User_recc, 0)
User_recc_1
```

Out[21]:

	276726	276729	276736	276737	276744	276745	276747	276748	276751	276754	...	1
276726	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
276729	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
276736	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
276737	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
276744	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
...	...	...	...	...	...	...	...	...	...	...	...	
162107	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
162109	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
162113	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
162121	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	
162129	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	

2182 rows × 2182 columns

Most Similar Users

In [23]:
```
User_recc_1.idxmax(axis=1)
```

Out[23]:
```
276726    276726
276729    276726
276736    276726
276737    276726
276744    276726
           ...
162107    276726
162109    276726
162113    161453
162121    276726
162129    276726
Length: 2182, dtype: int64
```

1. Extract the books which user ID 162113 & 161453 have watched

```
In [25]: Recc_Book[(Recc_Book['User.ID']==162113) | (Recc_Book['User.ID']==161453)]
```

Out[25]:

	User.ID	Book.Title	Book.Rating
8959	161453	Bread, Tomato, Garlic: Quick Cooking With 3 Ma...	9
8960	161453	The Ubiquitous Shrimp: From Simple to Exotic, ...	8
9989	162113	The Cape Ann (Contemporary American Fiction)	8

Inference:

The book read by the user 162113 is recommended to user 161453

2. Extract the books which user ID 276729 & 276726 have watched

```
In [29]: Recc_Book[(Recc_Book['User.ID']==276729) | (Recc_Book['User.ID']==276726)]
```

Out[29]:

	User.ID	Book.Title	Book.Rating
0	276726	Classical Mythology	5
1	276729	Clara Callan	3
2	276729	Decision in Normandy	6

```
In [35]: User_1 = Recc_Book[(Recc_Book['User.ID']==276729)]
         User_2 = Recc_Book[(Recc_Book['User.ID']==276726)]
```

```
In [36]: User_1['Book.Title']
```

```
Out[36]: 1            Clara Callan
         2    Decision in Normandy
         Name: Book.Title, dtype: object
```

```
In [37]: User_2['Book.Title']
```

```
Out[37]: 0    Classical Mythology
         Name: Book.Title, dtype: object
```

```
In [39]: pd.merge(User_1, User_2, how = 'outer',on='Book.Title')
```

Out[39]:

	User.ID_x	Book.Title	Book.Rating_x	User.ID_y	Book.Rating_y
0	276729.0	Clara Callan	3.0	NaN	NaN
1	276729.0	Decision in Normandy	6.0	NaN	NaN
2	NaN	Classical Mythology	NaN	276726.0	5.0

Inference:

The book read by the user 276726 is recommended to user 276729