Project ini berisikan rekomendasi film berdasarkan formula weighted rating dari IMBD

In [1]:

```
import pandas as pd
import numpy as np

movie_df=pd.read_csv("E:/Boothcamp/dataset/title.basics.tsv",sep='\t')
rating_df=pd.read_csv("E:/Boothcamp/dataset/title.ratings.tsv",sep='\t')

# pd.set_option('display.max_columns', None)
```

langkah pengerjaan

1. **Melihat df.head untuk menentukan kolom apa yg akan berpengaruh dalam pembuatan simple recomender**

In [2]:

```
print("movie_df",movie_df.head())
print("rating_df",rating_df.head())
```

```
movie_df       tconst  titleType                                       primaryTitle  \
0  tt0221078      short                          Circle Dance, Ute Indians
1  tt8862466  tvEpisode  ¡El #TeamOsos va con todo al "Reality del amor"!
2  tt7157720  tvEpisode                                       Episode #3.41
3  tt2974998  tvEpisode                          Episode dated 16 May 1987
4  tt2903620  tvEpisode                Frances Bavier: Aunt Bee Retires

                                originalTitle  isAdult startYear  \
0                      Circle Dance, Ute Indians        0      1898
1  ¡El #TeamOsos va con todo al "Reality del amor"!        0      2018
2                                  Episode #3.41        0      2016
3                      Episode dated 16 May 1987        0      1987
4                Frances Bavier: Aunt Bee Retires        0      1973

  endYear runtimeMinutes             genres
0      \N             \N  Documentary,Short
1      \N             \N       Comedy,Drama
2      \N             29   Comedy,Game-Show
3      \N             \N               News
4      \N             \N        Documentary
rating_df       tconst  averageRating  numVotes
0  tt0000001            5.6      1608
1  tt0000002            6.0       197
2  tt0000003            6.5      1285
3  tt0000004            6.1       121
4  tt0000005            6.1      2050
```

In [3]:

```
print("movie_df info",movie_df.info())
print("movie_df info",rating_df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9025 entries, 0 to 9024
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   tconst          9025 non-null   object
 1   titleType       9025 non-null   object
 2   primaryTitle    9011 non-null   object
 3   originalTitle   9011 non-null   object
 4   isAdult         9025 non-null   int64
 5   startYear       9025 non-null   object
 6   endYear         9025 non-null   object
 7   runtimeMinutes  9025 non-null   object
 8   genres          9014 non-null   object
```

```
dtypes: int64(1), object(8)
memory usage: 634.7+ KB
movie_df info None
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1030009 entries, 0 to 1030008
Data columns (total 3 columns):
 #   Column         Non-Null Count    Dtype
---  ------         --------------    -----
 0   tconst         1030009 non-null  object
 1   averageRating  1030009 non-null  float64
 2   numVotes       1030009 non-null  int64
dtypes: float64(1), int64(1), object(1)
memory usage: 23.6+ MB
movie_df info None
```

Setelah melihat df.head , kita menentukan kolom yang berpengaruh dalam simple recomender, yaitu : tconst titleType primaryTitle originalTitle isAdult startYear endYear runtimeMinutes genres

kolom tsb harus di bersihkan null/ Nan dan type datanya harus sesuai

terlihat bahwa primaryTitle,originalTitle dan genres tedapat null maka kita akan mengambil data df yg tidak ada null dengan .loc

In [4]:

```
movie_df = movie_df.loc[(movie_df['primaryTitle'].notnull()) & (movie_df['originalTitle']
.notnull())]
movie_df = movie_df.loc[movie_df['genres'].notnull()]

#menampilkan jumlah data setelah data dengan nilai NULL dibuang

print("movie_df info",movie_df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 9000 entries, 0 to 8999
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   tconst          9000 non-null   object
 1   titleType       9000 non-null   object
 2   primaryTitle    9000 non-null   object
 3   originalTitle   9000 non-null   object
 4   isAdult         9000 non-null   int64
 5   startYear       9000 non-null   object
 6   endYear         9000 non-null   object
 7   runtimeMinutes  9000 non-null   object
 8   genres          9000 non-null   object
dtypes: int64(1), object(8)
memory usage: 703.1+ KB
movie_df info None
```

Merubah type kolom startyear & runtimeMinutes menjadi float

tapi tidak bisa karena ada \N sehingga tidak bisa dirubah menjadi float.

pertama2 kita harus merubah menjadi np.nan

lalu membuang nan di kolom startYear dan runTimesMinutas.

In [5]:

```
movie_df['startYear']=movie_df['startYear'].replace('\\N',np.nan)
movie_df['endYear']=movie_df['endYear'].replace('\\N',np.nan)
movie_df['runtimeMinutes']=movie_df['runtimeMinutes'].replace('\\N',np.nan)

movie_df['startYear']=movie_df['startYear'].astype('float')
movie_df['endYear']=movie_df['endYear'].astype('float')
movie_df['runtimeMinutes']=movie_df['runtimeMinutes'].astype('float')

movie_df=movie_df.dropna(subset=['startYear','runtimeMinutes'])
```

```
print("movie_df info",movie_df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2558 entries, 2 to 8990
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   tconst          2558 non-null   object 
 1   titleType       2558 non-null   object 
 2   primaryTitle    2558 non-null   object 
 3   originalTitle   2558 non-null   object 
 4   isAdult         2558 non-null   int64  
 5   startYear       2558 non-null   float64
 6   endYear         39 non-null     float64
 7   runtimeMinutes  2558 non-null   float64
 8   genres          2558 non-null   object 
dtypes: float64(3), int64(1), object(5)
memory usage: 199.8+ KB
movie_df info None
```

karena kita akan membuat simple recomender dengan kolom genre maka kita akan merubah kolom genre menjadi list

In [6]:

```
print('genre as is ',movie_df['genres'].head())

def split_genre(x):
    if ',' in x:
        return x.split(',')
    else :
        return []

movie_df['genres'] = movie_df['genres'].apply(lambda x :split_genre(x))



print(movie_df['genres'].head())

print('genre to be',movie_df.info())
```

```
2           Comedy,Game-Show
5    Animation,Comedy,Family
6     Animation,Comedy,Drama
7                     Comedy
8                      Adult
Name: genres, dtype: object
2            [Comedy, Game-Show]
5    [Animation, Comedy, Family]
6     [Animation, Comedy, Drama]
7                             []
8                             []
Name: genres, dtype: object
<class 'pandas.core.frame.DataFrame'>
Index: 2558 entries, 2 to 8990
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   tconst          2558 non-null   object 
 1   titleType       2558 non-null   object 
 2   primaryTitle    2558 non-null   object 
 3   originalTitle   2558 non-null   object 
 4   isAdult         2558 non-null   int64  
 5   startYear       2558 non-null   float64
 6   endYear         39 non-null     float64
 7   runtimeMinutes  2558 non-null   float64
 8   genres          2558 non-null   object 
dtypes: float64(3), int64(1), object(5)
memory usage: 199.8+ KB
None
```

sementara movie_df sudah cukup dibersihkan

saatnya mengecek data rating_df

In [7]:

```
print("rating_df",rating_df.head())
print(rating_df.info())

print(rating_df['averageRating'].isna().sum())

print(rating_df['numVotes'].isna().sum())
```

```
rating_df       tconst  averageRating  numVotes
0  tt0000001             5.6      1608
1  tt0000002             6.0       197
2  tt0000003             6.5      1285
3  tt0000004             6.1       121
4  tt0000005             6.1      2050
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1030009 entries, 0 to 1030008
Data columns (total 3 columns):
 #   Column         Non-Null Count    Dtype
---  ------         --------------    -----
 0   tconst         1030009 non-null  object
 1   averageRating  1030009 non-null  float64
 2   numVotes       1030009 non-null  int64
dtypes: float64(1), int64(1), object(1)
memory usage: 23.6+ MB
None
0
0
```

rating_df sudah siap digunakan (type data sudah sesuai)

selanjut nya menggabungkan movie_df dengan rating_df menjadi movie_rating_df

In [8]:

```
movie_rating_df=pd.merge(movie_df,rating_df,how='inner',on='tconst')

# print(movie_rating_df.head())

print(movie_rating_df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1004 entries, 0 to 1003
Data columns (total 11 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   tconst          1004 non-null   object
 1   titleType       1004 non-null   object
 2   primaryTitle    1004 non-null   object
 3   originalTitle   1004 non-null   object
 4   isAdult         1004 non-null   int64
 5   startYear       1004 non-null   float64
 6   endYear         17 non-null     float64
 7   runtimeMinutes  1004 non-null   float64
 8   genres          1004 non-null   object
 9   averageRating   1004 non-null   float64
 10  numVotes        1004 non-null   int64
dtypes: float64(4), int64(2), object(5)
memory usage: 86.4+ KB
None
```

setelah dicek dengan df.info selain kolom endYear sudah tidak ditemukan nan. kolom endyear ini tidak berpengaruh pada pembuatan system recomder. maka dapat diabaikan

selanjutnya kita membuat weight rating formula dari IMDB note : v= jumlah votes film tsb m = adalah jumlah minimum vote yang dibutuhkan agar masuk dalam chart (quantile 0.8 dari numVotes) R= rata2 rating film tsb C=

rata2 jumlah votes dari semesta film

In [9]:

```
def weight_rating(df):
    v = df['numVotes']
    m = df['numVotes'].quantile (0.8)
    R = df['averageRating']
    C = df['numVotes'].mean()
    df['score']=(v/(v+m)*R)+(m/(v+m)*C)
    return df['score']

weight_rating(movie_rating_df)

print(movie_rating_df.head())
```

```
      tconst  titleType          primaryTitle         originalTitle  isAdult  \
0  tt0043745      short             Lion Down             Lion Down        0
1  tt0167491      video     Wicked Covergirls     Wicked Covergirls        1
2  tt6574096  tvEpisode  Shadow Play - Part 2  Shadow Play - Part 2        0
3  tt2262289      movie               The Pin               The Pin        0
4  tt0874027  tvEpisode         Episode #32.9         Episode #32.9        0

   startYear  endYear  runtimeMinutes                          genres  \
0     1951.0      NaN             7.0     [Animation, Comedy, Family]
1     1998.0      NaN            85.0                              []
2     2017.0      NaN            22.0  [Adventure, Animation, Comedy]
3     2013.0      NaN            85.0                              []
4     2006.0      NaN            29.0       [Comedy, Game-Show, News]

   averageRating  numVotes       score
0            7.1       459  250.223731
1            5.7         7  715.825961
2            8.5       240  364.467008
3            7.7        27  660.558308
4            8.0         8  712.907286
```

Setelah berhasil memunculkan score , maka kita akan memfilter 100 film yang numVotes diatas dari nilai m

In [10]:

```
def hundred_film_recomender (df,top=10):
    m = df['numVotes'].quantile (0.8)
    df=df.loc[df['numVotes']>=m]
    df=df.sort_values(by='score',ascending=False)
    df=df[:top]
    return df

df_top=(hundred_film_recomender(movie_rating_df))

print(df_top)
```

```
        tconst  titleType              primaryTitle             originalTitle  \
657  tt3820128      short                  The Herd                  The Herd
358  tt0882806      movie                Sugar Boxx                Sugar Boxx
616  tt0026373      movie  Folies Bergère de Paris  Folies Bergère de Paris
933  tt0095194      movie           Galactic Gigolo           Galactic Gigolo
36   tt0446043      movie            Opie Gets Laid                 Sunnyvale
400  tt0241687      movie             Loving Memory             Loving Memory
605  tt0945153   tvSeries                Maui Fever                Maui Fever
961  tt1010399      movie           The Big Sellout     Der große Ausverkauf
2    tt6574096  tvEpisode      Shadow Play - Part 2     Shadow Play - Part 2
331  tt0017382      movie               The Show-Off              The Show-Off

     isAdult  startYear  endYear  runtimeMinutes  \
657        0     2014.0      NaN            21.0
358        0     2009.0      NaN            86.0
616        0     1935.0      NaN            82.0
933        0     1987.0      NaN            80.0
36         0     2005.0      NaN            75.0
400        0     1971.0      NaN            57.0
```

```
605         0     2007.0      NaN            30.0
961         0     2007.0      NaN            94.0
2           0     2017.0      NaN            22.0
331         0     1926.0      NaN            82.0

                            genres  averageRating  numVotes       score
657       [Horror, Short, Thriller]            6.5       230  371.220102
358                   [Crime, Drama]            3.5       229  370.516434
616                [Comedy, Musical]            6.6       231  370.477450
933                 [Comedy, Sci-Fi]            3.4       229  370.466434
36                 [Comedy, Romance]            4.7       230  370.318141
400                               []            6.1       233  368.650058
605                               []            2.7       233  366.935340
961                               []            7.6       238  365.532820
2    [Adventure, Animation, Comedy]            8.5       240  364.467008
331                  [Comedy, Drama]            6.8       242  362.082010
```

Membuat user_prefer_recomender dengan pilihan pada isAdult,startYear,genre

In [11]:

```
# print(movie_rating_df)

def user_prefer_recomender(df,ask_isAdult,ask_genres):
   if ask_isAdult.lower()=='yes':
      df=df.loc[df['isAdult']==1]
   if ask_isAdult.lower()=='no':
      df=df.loc[df['isAdult']==0]
   if ask_genres.lower()=='all':
      df=df
   else:
      def filter_genres (x):
         if ask_genres.lower() in str(x).lower():
            return True
         else:
            return False
      df = df.loc[df['genres'].apply (lambda x : filter_genres(x))]

   return df

print(user_prefer_recomender(df_top,ask_isAdult='no',ask_genres='Comedy'))
```

```
        tconst  titleType             primaryTitle            originalTitle  \
616  tt0026373      movie  Folies Bergère de Paris  Folies Bergère de Paris
933  tt0095194      movie          Galactic Gigolo          Galactic Gigolo
36   tt0446043      movie           Opie Gets Laid                Sunnyvale
2    tt6574096  tvEpisode     Shadow Play - Part 2     Shadow Play - Part 2
331  tt0017382      movie              The Show-Off              The Show-Off

     isAdult  startYear  endYear  runtimeMinutes  \
616        0     1935.0      NaN            82.0
933        0     1987.0      NaN            80.0
36         0     2005.0      NaN            75.0
2          0     2017.0      NaN            22.0
331        0     1926.0      NaN            82.0

                            genres  averageRating  numVotes       score
616                [Comedy, Musical]            6.6       231  370.477450
933                 [Comedy, Sci-Fi]            3.4       229  370.466434
36                 [Comedy, Romance]            4.7       230  370.318141
2    [Adventure, Animation, Comedy]            8.5       240  364.467008
331                  [Comedy, Drama]            6.8       242  362.082010
```