MOVIE RECOMMENDATION SYSTEM

Import libraries

```
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.preprocessing import MinMaxScaler
```

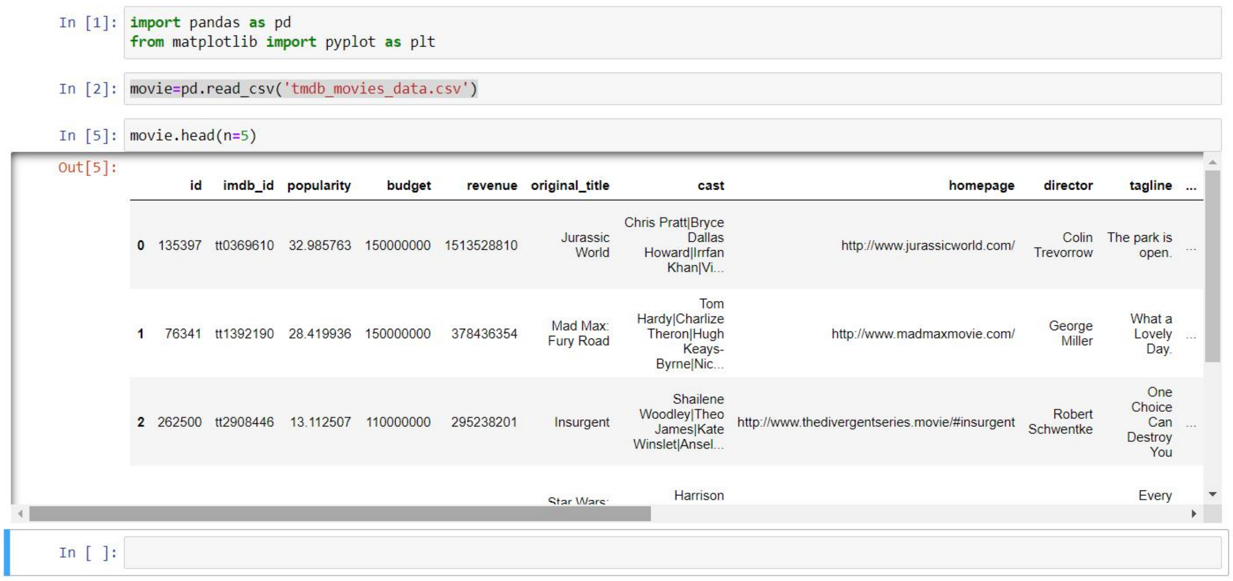
Read CSV File

```
movie=pd.read_csv('tmdb_movies_data.csv')
```

Link to Dataset

GLANCE AT THE DATA SET

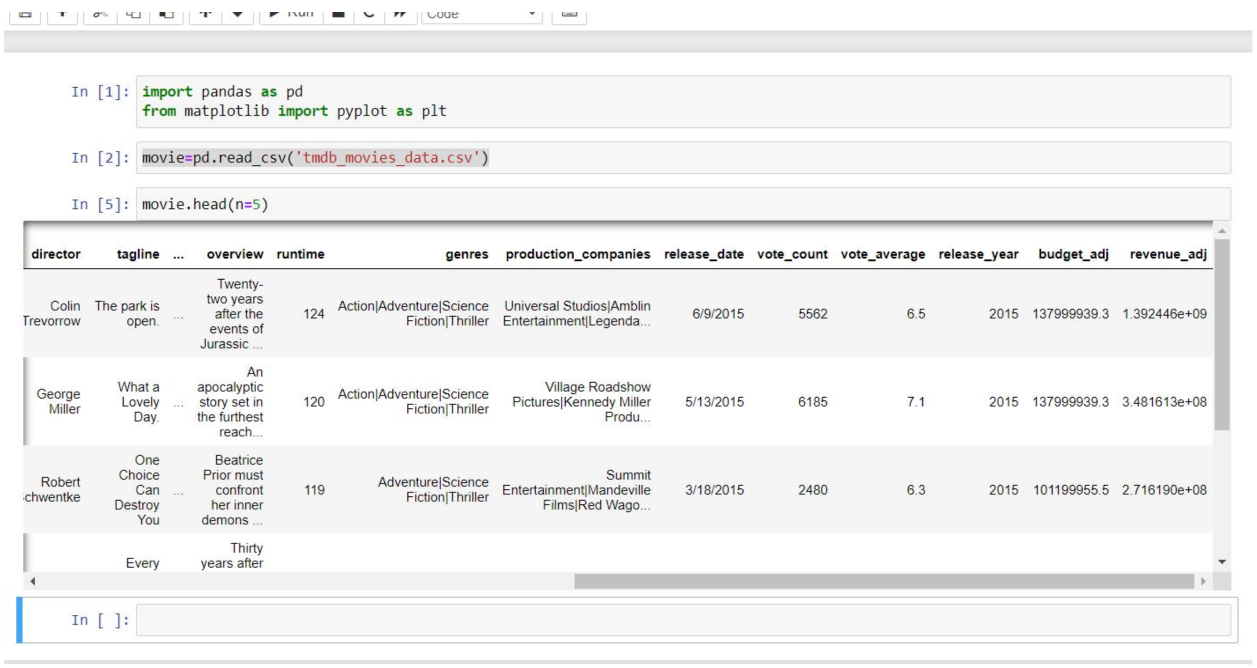
```
movie.head(n=5)     //a look at first 5 rows of data set
```



```
In [1]: import pandas as pd
        from matplotlib import pyplot as plt

In [2]: movie=pd.read_csv('tmdb_movies_data.csv')

In [5]: movie.head(n=5)

Out[5]:
```

	id	imdb_id	popularity	budget	revenue	original_title	cast	homepage	director	tagline	...
0	135397	tt0369610	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	http://www.jurassicworld.com/	Colin Trevorrow	The park is open.	...
1	76341	tt1392190	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	http://www.madmaxmovie.com/	George Miller	What a Lovely Day.	...
2	262500	tt2908446	13.112507	110000000	295238201	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	http://www.thedivergentseries.movie/#insurgent	Robert Schwentke	One Choice Can Destroy You	...
						Star Wars:	Harrison			Every	

```
In [ ]:
```



```
In [1]: import pandas as pd
        from matplotlib import pyplot as plt

In [2]: movie=pd.read_csv('tmdb_movies_data.csv')

In [5]: movie.head(n=5)
```

director	tagline	...	overview	runtime	genres	production_companies	release_date	vote_count	vote_average	release_year	budget_adj	revenue_adj
Colin Trevorrow	The park is open.	...	Twenty-two years after the events of Jurassic ...	124	Action|Adventure|Science Fiction|Thriller	Universal Studios|Amblin Entertainment|Legenda...	6/9/2015	5562	6.5	2015	137999939.3	1.392446e+09
George Miller	What a Lovely Day.	...	An apocalyptic story set in the furthest reach...	120	Action|Adventure|Science Fiction|Thriller	Village Roadshow Pictures|Kennedy Miller Produ...	5/13/2015	6185	7.1	2015	137999939.3	3.481613e+08
Robert chwentke	One Choice Can Destroy You	...	Beatrice Prior must confront her inner demons ...	119	Adventure|Science Fiction|Thriller	Summit Entertainment|Mandeville Films|Red Wago...	3/18/2015	2480	6.3	2015	101199955.5	2.716190e+08
	Every		Thirty years after									

```
In [ ]:
```

movie.shape //a look at rows and column in data set

Output:

```
(10866, 21)
```

Preprocessing

```
movie['populartity']
```

Output:

```
0        32.985763
1        28.419936
2        13.112507
3        11.173104
4         9.335014
           ...
10861     0.080598
10862     0.065543
10863     0.065141
```

```
10864      0.064317
10865      0.035919
```

```
movie['vote_count']
```

Output:

```
0          5562
1          6185
2          2480
3          5292
4          2947
           ...
10861        11
10862        20
10863        11
10864        22
10865        15
```

```
movie['vote_average']
```

Output:

```
0          6.5
1          7.1
2          6.3
3          7.5
4          7.3
          ...
10861      7.4
10862      5.7
10863      6.5
10864      5.4
10865      1.5
```

We are going to use popularity ,vote_count,vote_average column to recommend movie to a new user:

Calculating various statistical parameter on popularity :

```
movie.describe()['popularity']
```

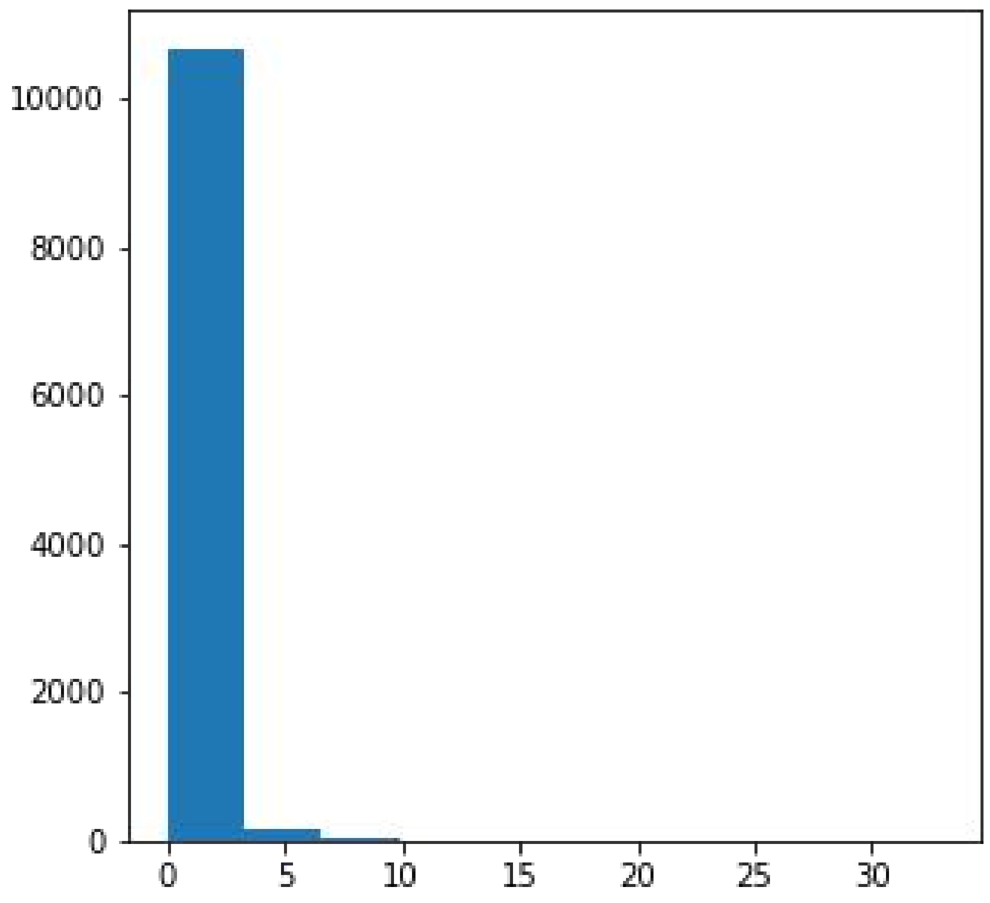
Output:

```
count    10866.000000
mean         0.646441
std          1.000185
min          0.000065
25%          0.207583
50%          0.383856
75%          0.713817
max         32.985763
```

By seeing above data it is clear that highest popularity has been rated as 32.985763 whereas mean is 0.6466441 data is highly skewed 75th percentile shows 0.713817

Seeing the histogram Distribution:

```
plt.figure(figsize=(5,5))
plt.hist(movie['popularity'])
plt.show()
```



Calculating various statistical parameter on vote_count :

```
movie.describe()['vote_count']
```

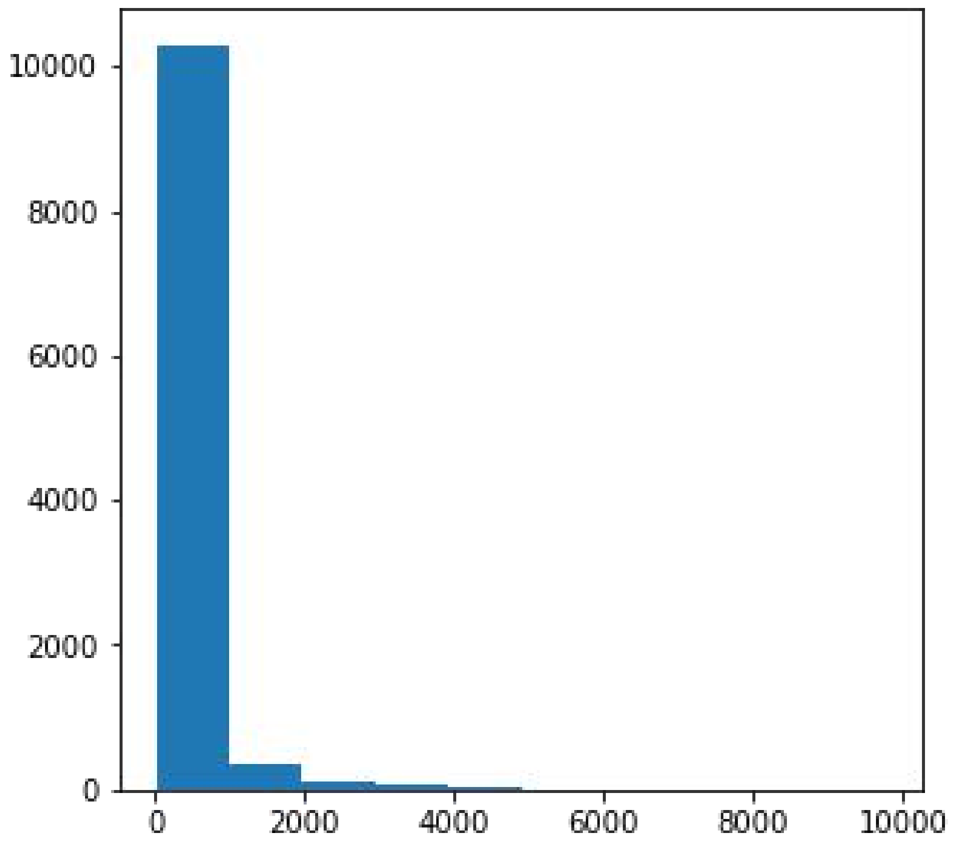
Output:

```
count    10866.000000
mean       217.389748
std        575.619058
min         10.000000
25%         17.000000
50%         38.000000
75%        145.750000
max       9767.000000
```

By seeing above data it is clear that highest votecount is b
een rated as 9767 whereas mean is 217.389748 75th percentile
 shows 145.75

Seeing the histogram Distribution:

```
plt.figure(figsize=(5,5))
plt.hist(movie['vote_count'])
plt.show()
```



Calculating various statistical parameter on vote_average :

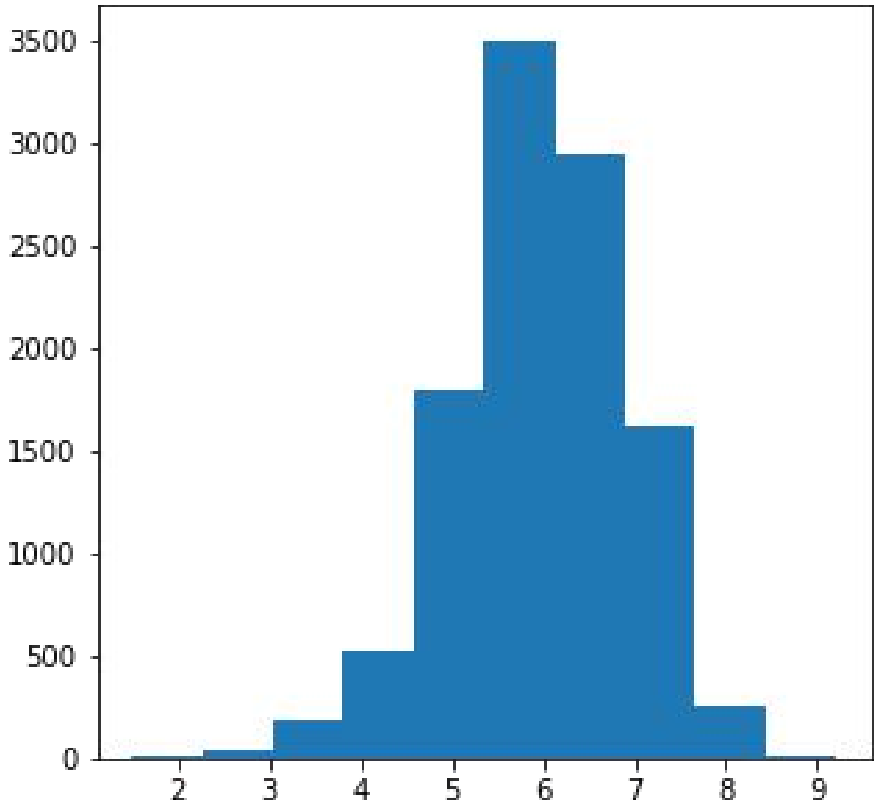
```
movie.describe()['vote_average']
```

```
count    10866.000000
mean         5.974922
std          0.935142
min          1.500000
25%          5.400000
50%          6.000000
75%          6.600000
max          9.200000
```

By seeing above data it is clear that highest votecount is been rated as 9.200000 whereas mean is 5.97492275th percentile shows 6.600000

Seeing the histogram Distribution:

```
plt.figure(figsize=(5,5))
plt.hist(movie['vote_average'])
plt.show()
```



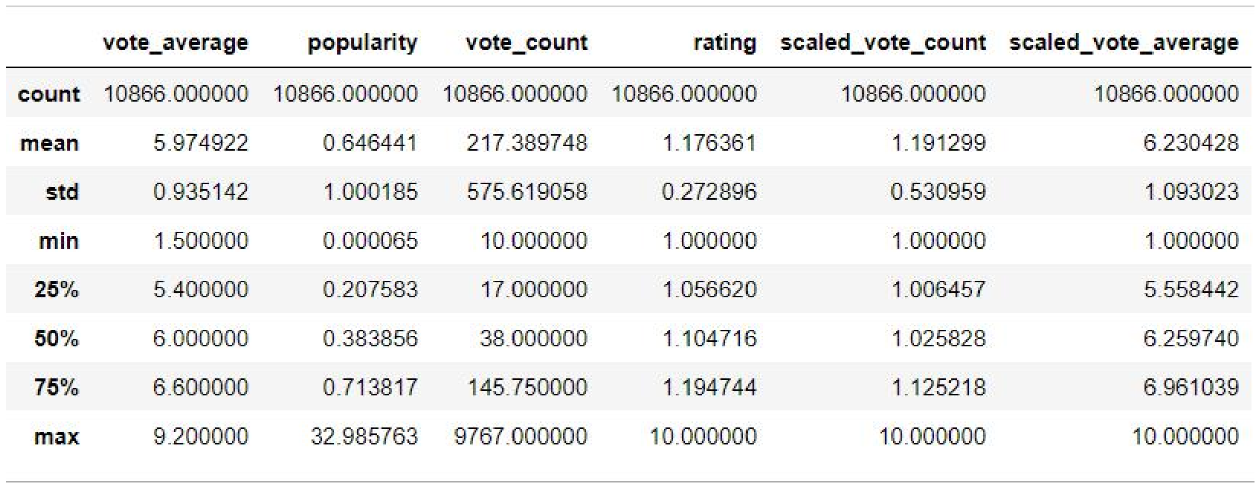
SCALING DATA

We are going to use sklearn preprocessing to scale populari
ty (1-10).

```
pd_data=pd.DataFrame(movie[{'original_title','popularity','v
ote_count','vote_average'}])
scaler = MinMaxScaler(feature_range=(1, 10))
pd_data[['rating','scaled_vote_count','scaled_vote_average']]
= scaler.fit_transform(
pd_data[['popularity','vote_count','vote_average']])
print(pd_data)
```

```
         rating        scaled_vote_count   scaled_vote_average

0       10.000000              6.121246                6.844156
1        8.754235              6.695911                7.545455
2        4.577671              3.278364                6.610390
3        4.048514              5.872194                8.012987
4        3.546999              3.709132                7.779221
...           ...                   ...                     ...
10861    1.021973              1.000922                7.896104
10862    1.017865              1.009224                5.909091
10863    1.017756              1.000922                6.844156
10864    1.017531              1.011069                5.558442
10865    1.009783              1.004612                1.000000
```

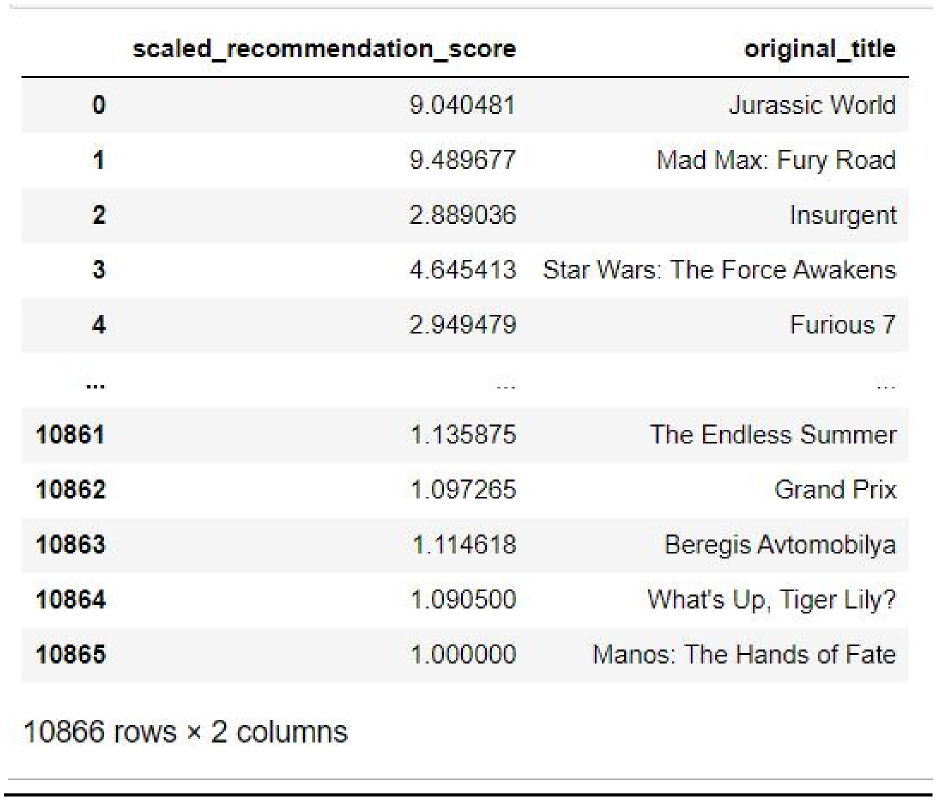
```
pd_data.describe()
```



	vote_average	popularity	vote_count	rating	scaled_vote_count	scaled_vote_average
count	10866.000000	10866.000000	10866.000000	10866.000000	10866.000000	10866.000000
mean	5.974922	0.646441	217.389748	1.176361	1.191299	6.230428
std	0.935142	1.000185	575.619058	0.272896	0.530959	1.093023
min	1.500000	0.000065	10.000000	1.000000	1.000000	1.000000
25%	5.400000	0.207583	17.000000	1.056620	1.006457	5.558442
50%	6.000000	0.383856	38.000000	1.104716	1.025828	6.259740
75%	6.600000	0.713817	145.750000	1.194744	1.125218	6.961039
max	9.200000	32.985763	9767.000000	10.000000	10.000000	10.000000

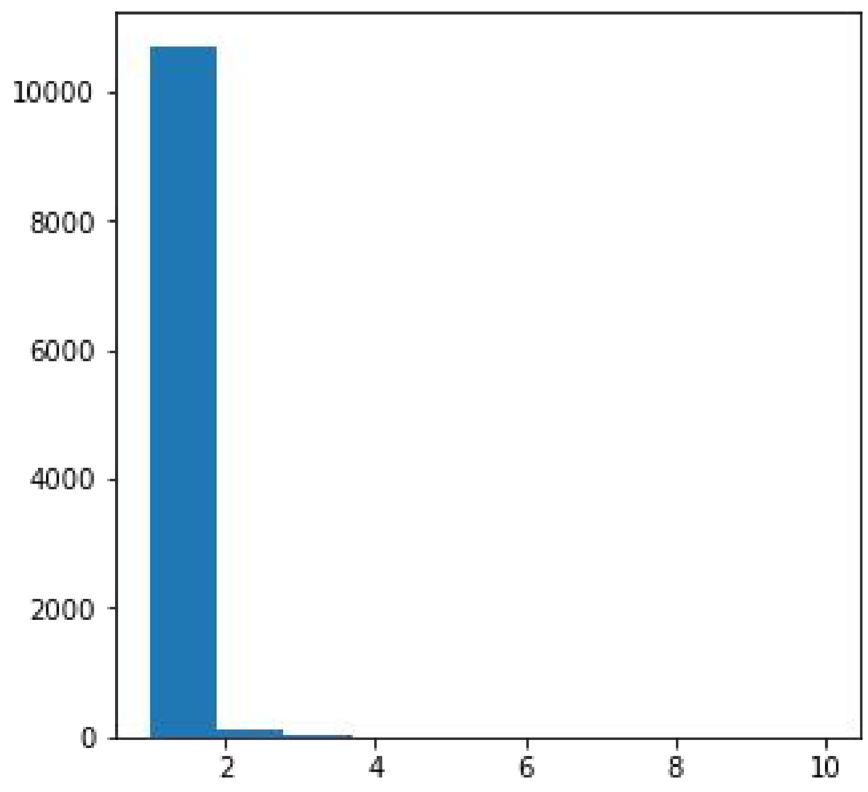
SINCE ALL THE THREE DATA REFLECTS MOVIE CHOICE WE ARE GOING TO TAKE PRODUCT OF THESE AND THEN SCALE IT 1-10 AND WE CALL THIS AS SCALED_RECOMMENDATION_SCORE.

```
pd_data['recommendation_score']=pd_data["scaled_vote_count"]
*pd_data["scaled_vote_average"]*pd_data["rating"]
pd_data[['scaled_recommendation_score']]= scaler.fit_transfo
rm(
pd_data[['recommendation_score']])
pd_data[['scaled_recommendation_score','original_title']]
```



	scaled_recommendation_score	original_title
0	9.040481	Jurassic World
1	9.489677	Mad Max: Fury Road
2	2.889036	Insurgent
3	4.645413	Star Wars: The Force Awakens
4	2.949479	Furious 7
...	...	...
10861	1.135875	The Endless Summer
10862	1.097265	Grand Prix
10863	1.114618	Beregis Avtomobilya
10864	1.090500	What's Up, Tiger Lily?
10865	1.000000	Manos: The Hands of Fate

10866 rows × 2 columns



```
pd_data.scaled_recommendation_score.describe()

count    10866.000000
mean         1.169833
std          0.268769
min          1.000000
25%          1.101473
50%          1.122421
75%          1.152103
max         10.000000
```

Sorting Movies

```
sorted=pd_data.sort_values(['scaled_recommendation_score'],ascending=False)[['scaled_recommendation_score','original_title']]
```

Recommending Movies

```
print(sorted[['original_title']])
```

Output:

```
id                      original_title
629                       Interstellar
1                   Mad Max: Fury Road
0                       Jurassic World
1919                         Inception
630            Guardians of the Galaxy
...                                ...
3822                       Sand Sharks
7220     Superbabies: Baby Geniuses 2
4882                    Jurassic Shark
7772                     Transmorphers
10865          Manos: The Hands of Fate
```

Our recommendation system recommends Interstellar as first choice.