In [13]:
```
# 3rd Project---> IMDB MOVIES RATING DATA ANALYSIS USING PANDAS

import pandas as pd
```

In [14]:
```
movies= pd.read_excel(r'C:\Users\sirius\Desktop\DATA SCIENCE CLASS PRACTICE\movie.xlsx')
movies
```

Out[14]:

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy
...	...	...	...
27273	131254	Kein Bund für's Leben (2007)	Comedy
27274	131256	Feuer, Eis & Dosenbier (2002)	Comedy
27275	131258	The Pirates (2014)	Adventure
27276	131260	Rentun Ruusu (2001)	(no genres listed)
27277	131262	Innocence (2014)	Adventure|Fantasy|Horror

27278 rows × 3 columns

In [15]:
```
movies.head(10)
```

Out[15]:

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy
5	6	Heat (1995)	Action|Crime|Thriller
6	7	Sabrina (1995)	Comedy|Romance
7	8	Tom and Huck (1995)	Adventure|Children
8	9	Sudden Death (1995)	Action
9	10	GoldenEye (1995)	Action|Adventure|Thriller

In [16]:
```
rating= pd.read_csv(r'C:\Users\sirius\Desktop\DATA SCIENCE CLASS PRACTICE\rating.csv')
```

In [17]:
```
rating.head(10)
```

Out[17]:

	userId	movieId	rating	timestamp
0	1	2	3.5	2005-04-02 23:53:47
1	1	29	3.5	2005-04-02 23:31:16
2	1	32	3.5	2005-04-02 23:33:39
3	1	47	3.5	2005-04-02 23:32:07
4	1	50	3.5	2005-04-02 23:29:40
5	1	112	3.5	2004-09-10 03:09:00
6	1	151	4.0	2004-09-10 03:08:54
7	1	223	4.0	2005-04-02 23:46:13
8	1	253	4.0	2005-04-02 23:35:40
9	1	260	4.0	2005-04-02 23:33:46

In [18]:
```
tag= pd.read_csv(r'C:\Users\sirius\Desktop\DATA SCIENCE CLASS PRACTICE\tag.csv')
tag.head(10)
```

Out[18]:

	userId	movieId	tag	timestamp
0	18	4141	Mark Waters	2009-04-24 18:19:40
1	65	208	dark hero	2013-05-10 1:41:18
2	65	353	dark hero	2013-05-10 1:41:19
3	65	521	noir thriller	2013-05-10 1:39:43
4	65	592	dark hero	2013-05-10 1:41:18
5	65	668	bollywood	2013-05-10 1:37:56
6	65	898	screwball comedy	2013-05-10 1:42:40
7	65	1248	noir thriller	2013-05-10 1:39:43
8	65	1391	mars	2013-05-10 1:40:55
9	65	1617	neo-noir	2013-05-10 1:43:37

In [19]:
```
rating.shape
```

Out[19]:
```
(20000263, 4)
```

In [20]:
```
tag.shape
```

Out[20]:
```
(45379, 4)
```

In [21]:
```
print(len(movies))
print(len(tag))
print(len(rating))
```

```
27278
45379
20000263
```

In [22]:
```
movies.head(2)
```

Out[22]:

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy

In [23]:
```
tag.head(3)
```

Out[23]:

	userId	movieId	tag	timestamp
0	18	4141	Mark Waters	2009-04-24 18:19:40
1	65	208	dark hero	2013-05-10 1:41:18
2	65	353	dark hero	2013-05-10 1:41:19

In [24]:
```
rating.tail(3)
```

Out[24]:

	userId	movieId	rating	timestamp
20000260	138493	69644	3.0	2009-12-07 18:10:57
20000261	138493	70286	5.0	2009-11-13 15:42:24
20000262	138493	71619	2.5	2009-10-17 20:25:36

In [25]:
```
#FOR CURRENT ANALAYSIS , WE WILL REMOVE TIMESTAMP

del rating['timestamp']
```

In [26]:
```
rating.head(2)
```

Out[26]:

	userId	movieId	rating
0	1	2	3.5
1	1	29	3.5

In [27]:
```
del tag['timestamp']
```

In [28]:
```
tag.head(1)
```

Out[28]:

	userId	movieId	tag
0	18	4141	Mark Waters

In [29]:
```
len(rating.columns)
```

Out[29]:
```
3
```

In [30]:
```
len(tag.columns)
```

Out[30]:
```
3
```

In [31]:
```
# DATA STRUCTURES

row_0= tag.iloc[0]
type(row_0)
```

Out[31]:
```
pandas.core.series.Series
```

In [32]:
```
print(row_0)
```

```
userId              18
movieId           4141
tag         Mark Waters
Name: 0, dtype: object
```

In [33]:
```
row_1=tag.iloc[0:4]
print(row_1)
```

```
   userId  movieId           tag
0      18     4141   Mark Waters
1      65      208     dark hero
2      65      353     dark hero
3      65      521  noir thriller
```

In [34]:
```
row_0.index
```

Out[34]:
```
Index(['userId', 'movieId', 'tag'], dtype='object')
```

In [35]:
```
row_2=tag.iloc[[0,12,100,1000]]
print(row_2)
```

```
      userId  movieId                tag
0         18     4141        Mark Waters
12        65     2022              jesus
100      121    52973              drugs
1000     359    69526  needed more autobots
```

In [36]:
```
rating.corr()
```

Out[36]:

	userId	movieId	rating
userId	1.000000	-0.000850	0.001175
movieId	-0.000850	1.000000	0.002606
rating	0.001175	0.002606	1.000000

In [37]:
```
filter2= rating['rating']>0
filter2.all()
```

Out[37]:
```
True
```

In [38]:
```
#DATA CLEANING : HANDLING MISSING DATA

movies.shape
```

Out[38]:
```
(27278, 3)
```

In [39]:
```
movies.isnull().any().any()
```

Out[39]:
```
False
```

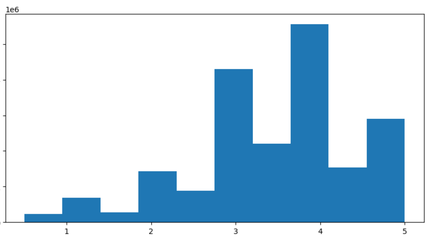
In [40]:
```
rating.isnull().any().any()
```

Out[40]:
```
False
```

In [44]:
```
# DATA VISUALIZATION

import numpy as np
import matplotlib.pyplot as plt

%matplotlib inline
plt.rcParams['figure.figsize']=10,5
plt.hist(rating['rating'],histtype='bar')
plt.show()
```



In [ ]: