kaggle task for data science with pandas ai

https://www.kaggle.com/code/harunshimanto/pandas-with-data-science-ai

In [12]:
```
import pandas as pd # import libraries
```

In [13]:
```
ratings=pd.read_csv(r'C:\Users\archa\Downloads\archive\rating.csv')
```

In [14]:
```
ratings.shape
```

Out[14]: (20000263, 4)

In [15]:
```
tags=pd.read_csv(r'C:\Users\archa\Downloads\archive\tag.csv')
```

In [16]:
```
tags.shape
```

Out[16]: (465564, 4)

In [17]:
```
ratings.head()
```

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

In [18]:
```
tags.head()
```

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 01:41:18
2	65	353	dark hero	2013-05-10 01:41:19
3	65	521	noir thriller	2013-05-10 01:39:43
4	65	592	dark hero	2013-05-10 01:41:18

In [19]:
```
movies=pd.read_csv(r'C:\Users\archa\Downloads\archive\movie.csv')
```

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

Out[20]: (27278, 3)

In [21]:
```
movies.head()
```

Out[21]:

	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

In [22]:
```
print(tags.columns)
print(ratings.columns)
print(movies.columns)
```

```
Index(['userId', 'movieId', 'tag', 'timestamp'], dtype='object')
Index(['userId', 'movieId', 'rating', 'timestamp'], dtype='object')
Index(['movieId', 'title', 'genres'], dtype='object')
```

In [23]:
```
del ratings['timestamp']
del tags['timestamp']
```

In [24]:
```
print(tags.columns)
print(ratings.columns)
print(movies.columns)
```

```
Index(['userId', 'movieId', 'tag'], dtype='object')
Index(['userId', 'movieId', 'rating'], dtype='object')
Index(['movieId', 'title', 'genres'], dtype='object')
```

In [25]:
```
tags.head(3)
```

Out[25]:

	userId	movieId	tag
0	18	4141	Mark Waters
1	65	208	dark hero
2	65	353	dark hero

In [27]:
```
tags.iloc[2]
```

Out[27]:
```
userId            65
movieId          353
tag        dark hero
Name: 2, dtype: object
```

In [28]:
```
row_0=tags.iloc[0]
row_0
```

Out[28]:
```
userId              18
movieId           4141
tag        Mark Waters
Name: 0, dtype: object
```

In [29]:
```
row_1=tags.iloc[1]
row_1
```

Out[29]:
```
userId            65
movieId          208
tag        dark hero
Name: 1, dtype: object
```

In [30]:
```
row_0.index
```

Out[30]:
```
Index(['userId', 'movieId', 'tag'], dtype='object')
```

In [34]:
```
ratings['rating'].describe()
```

Out[34]:
```
count    2.000026e+07
mean     3.525529e+00
std      1.051989e+00
min      5.000000e-01
25%      3.000000e+00
50%      3.500000e+00
75%      4.000000e+00
max      5.000000e+00
Name: rating, dtype: float64
```

In [35]:
```
ratings['rating'].mean()
```

Out[35]:
```
np.float64(3.5255285642993797)
```

In [37]:
```
ratings.mean()
```

Out[37]:
```
userId     69045.872583
movieId     9041.567330
rating         3.525529
dtype: float64
```

In [38]:
```
ratings['rating'].max()
```

Out[38]:
```
5.0
```

In [40]:
```
ratings['rating'].std()
```

Out[40]:
```
1.051988919275684
```

In [41]:
```
ratings['rating'].mode()
```

Out[41]:
```
0    4.0
Name: rating, dtype: float64
```

In [43]:
```
ratings.corr()
```

Out[43]:

	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 [45]:
```
filter1=ratings['rating']>10
print(filter1)
filter1.any()
```

```
0           False
1           False
2           False
3           False
4           False
            ...
20000258    False
20000259    False
20000260    False
20000261    False
20000262    False
Name: rating, Length: 20000263, dtype: bool
```

Out[45]: np.False_

In [47]:
```
filter2=ratings['rating']>0
filter2.all()
```

Out[47]: np.True_

In [48]:
```
movies.shape
```

Out[48]: (27278, 3)

In [49]:
```
movies.isnull().any().any()
```

Out[49]: np.False_

In [51]:
```
ratings.shape
```

Out[51]: (20000263, 3)

In [52]:
```
ratings.isnull().any().any()
```

Out[52]: np.False_

In [53]:
```
tags.shape
```

Out[53]: (465564, 3)

In [54]:
```
tags.isnull().any().any()
```

Out[54]:
```
np.True_
```

In [56]:
```
%matplotlib inline
ratings.hist(column='rating',figsize=(10,5))
```

Out[56]:
```
array([[<Axes: title={'center': 'rating'}>]], dtype=object)
```

In [62]:
```
ratings.boxplot(column ='rating',figsize=(10,5))
```

Out[62]:
```
<Axes: title={'center': 'rating'}>
```

In [63]:
```
tags['tag'].head()
```

Out[63]:
```
0      Mark Waters
1        dark hero
2        dark hero
3    noir thriller
4        dark hero
Name: tag, dtype: object
```

In [64]:
```
movies[['title','genres']].head()
```

Out[64]:

	title	genres
0	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	Jumanji (1995)	Adventure|Children|Fantasy
2	Grumpier Old Men (1995)	Comedy|Romance
3	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	Father of the Bride Part II (1995)	Comedy

In [ ]:
```

```

In [66]:
```
ratings[-10:]
```

Out[66]:

	userId	movieId	rating
20000253	138493	60816	4.5
20000254	138493	61160	4.0
20000255	138493	65682	4.5
20000256	138493	66762	4.5
20000257	138493	68319	4.5
20000258	138493	68954	4.5
20000259	138493	69526	4.5
20000260	138493	69644	3.0
20000261	138493	70286	5.0
20000262	138493	71619	2.5

In [67]:
```
tags_counts=tags['tag'].value_counts()
tags_counts[-10:]
```

Out[67]:
```
tag
chiptunes         1
ewan macgregor    1
Disguises         1
retarted          1
operatic          1
heartrending      1
film crew         1
es                1
girltalk          1
Spanish films     1
Name: count, dtype: int64
```

In [ ]:
```
tags_counts[:10].plot(kind='bar',figsize=(10,5))
tags_counts
```

Out[ ]:
```
tag
sci-fi             3384
based on a book    3281
atmospheric        2917
comedy             2779
action             2657
                   ...
heartrending          1
film crew             1
es                    1
girltalk              1
Spanish films         1
Name: count, Length: 38643, dtype: int64
```

In [ ]:
```

```

In [ ]:

In [ ]: