Python Implementation of Movie Recommendation System

November 26, 2022

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
import scipy
import scipy.stats as stats
import pylab
```

[2]:
```
df = pd.read_csv(r"content.csv")
```

[3]:
```
df
```

[3]:

	content_id	content_type	language	genre	duration	release_date	rating	episode_count	season_count
0	cont_475_19_32	series	english	drama	4980000	2018-07-01	10	32	19
1	cont_2185_15_21	series	english	drama	3000000	2016-03-29	4	21	15
2	cont_4857_13_28	series	tamil	comedy	3120000	2006-03-06	8	28	13
3	cont_3340_1_5	sports	hindi	cricket	9900000	2009-01-10	0	5	1
4	cont_1664_10_29	series	hindi	action	3660000	2020-05-25	2	29	10
...	...	...	...	...	...	...	...	...	...
48640	cont_4218_6_15	series	hindi	drama	3360000	2015-02-04	6	15	6
48641	cont_2533_1_14	series	marathi	sci-fi	3120000	2002-01-15	4	14	1
48642	cont_4606_33_5	series	hindi	drama	3180000	2006-02-18	6	5	33
48643	cont_3708_9_1	series	english	drama	4020000	2010-04-12	5	1	9
48644	cont_3470_2_4	series	english	horror	2760000	1997-03-26	8	4	2

```
[48645 rows x 9 columns]
```

[4]:
```
df.dtypes
```

[4]:
```
content_id       object
content_type     object
language         object
genre            object
duration          int64
release_date     object
rating            int64
episode_count     int64
season_count      int64
dtype: object
```

[5]:
```
df.describe
```

[5]:
```
<bound method NDFrame.describe of             content_id content_type language
genre  duration release_date  \
0       cont_475_19_32       series  english    drama   4980000   2018-07-01
1      cont_2185_15_21       series  english    drama   3000000   2016-03-29
2      cont_4857_13_28       series    tamil   comedy   3120000   2006-03-06
3        cont_3340_1_5       sports    hindi  cricket   9900000   2009-01-10
4      cont_1664_10_29       series    hindi   action   3660000   2020-05-25
...                ...          ...      ...      ...      ...          ...
48640   cont_4218_6_15       series    hindi    drama   3360000   2015-02-04
48641   cont_2533_1_14       series  marathi   sci-fi   3120000   2002-01-15
48642   cont_4606_33_5       series    hindi    drama   3180000   2006-02-18
48643    cont_3708_9_1       series  english    drama   4020000   2010-04-12
48644    cont_3470_2_4       series  english   horror   2760000   1997-03-26

       rating  episode_count  season_count
0          10             32            19
1           4             21            15
2           8             28            13
3           0              5             1
4           2             29            10
...       ...            ...           ...
48640       6             15             6
48641       4             14             1
48642       6              5            33
48643       5              1             9
48644       8              4             2

[48645 rows x 9 columns]>
```

[6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48645 entries, 0 to 48644
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   content_id     48645 non-null  object
 1   content_type   48645 non-null  object
 2   language       48645 non-null  object
 3   genre          48645 non-null  object
 4   duration       48645 non-null  int64 
 5   release_date   48645 non-null  object
 6   rating         48645 non-null  int64 
 7   episode_count  48645 non-null  int64 
 8   season_count   48645 non-null  int64 
dtypes: int64(4), object(5)
memory usage: 3.3+ MB
```

[7]:
```
df.drop(["content_id", "release_date"], axis=1, inplace=True)
```

[8]:
```
df
```

[8]:

	content_type	language	genre	duration	rating	episode_count	season_count
0	series	english	drama	4980000	10	32	19
1	series	english	drama	3000000	4	21	15
2	series	tamil	comedy	3120000	8	28	13
3	sports	hindi	cricket	9900000	0	5	1
4	series	hindi	action	3660000	2	29	10
...	...	...	...	...	...	...	...
48640	series	hindi	drama	3360000	6	15	6
48641	series	marathi	sci-fi	3120000	4	14	1
48642	series	hindi	drama	3180000	6	5	33
48643	series	english	drama	4020000	5	1	9
48644	series	english	horror	2760000	8	4	2

```
[48645 rows x 7 columns]
```

[9]:
```
df1 = pd.get_dummies(df.content_type)
```

[10]:
```
df1
```

[10]:
```
       movies  series  sports  teasers
0           0       1       0        0
1           0       1       0        0
2           0       1       0        0
3           0       0       1        0
4           0       1       0        0
...       ...     ...     ...      ...
48640       0       1       0        0
48641       0       1       0        0
48642       0       1       0        0
48643       0       1       0        0
48644       0       1       0        0

[48645 rows x 4 columns]
```

[11]:
```
df1.drop(["teasers"], axis=1, inplace=True)
```

[12]:
```
df1.tail()
```

[12]:
```
       movies  series  sports
48640       0       1       0
48641       0       1       0
48642       0       1       0
48643       0       1       0
48644       0       1       0
```

[13]:
```
df2 = pd.get_dummies(df.genre)
df3 = pd.get_dummies(df.language)
```

[14]:
```
df2.drop(["thriller"], axis=1, inplace=True)
```

[15]:
```
df3.drop(["telugu"], axis=1, inplace=True)
```

[16]:
```
df11 = pd.concat([df, df1], axis=1)
```

[17]:
```
df12 = pd.concat([df11, df2], axis=1)
```

[18]:
```
df13 = pd.concat([df12, df3], axis=1)
```

[19]:
```
df13
```

[19]:

	content_type	language	genre	duration	rating	episode_count	\
0	series	english	drama	4980000	10	32	
1	series	english	drama	3000000	4	21	
2	series	tamil	comedy	3120000	8	28	
3	sports	hindi	cricket	9900000	0	5	
4	series	hindi	action	3660000	2	29	
...	...	...	...	...	...	...	
48640	series	hindi	drama	3360000	6	15	
48641	series	marathi	sci-fi	3120000	4	14	
48642	series	hindi	drama	3180000	6	5	
48643	series	english	drama	4020000	5	1	
48644	series	english	horror	2760000	8	4	

	season_count	movies	series	sports	...	bengali	english	gujarati	\
0	19	0	1	0	...	0	1	0	
1	15	0	1	0	...	0	1	0	
2	13	0	1	0	...	0	0	0	
3	1	0	0	1	...	0	0	0	
4	10	0	1	0	...	0	0	0	
...	...	...	...	...	...	...	...	...	
48640	6	0	1	0	...	0	0	0	
48641	1	0	1	0	...	0	0	0	
48642	33	0	1	0	...	0	0	0	
48643	9	0	1	0	...	0	1	0	
48644	2	0	1	0	...	0	1	0	

	hindi	kannada	malayalam	marathi	oriya	punjabi	tamil
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
2	0	0	0	0	0	0	1
3	1	0	0	0	0	0	0
4	1	0	0	0	0	0	0
...	...	...	...	...	...	...	...
48640	1	0	0	0	0	0	0
48641	0	0	0	1	0	0	0
48642	1	0	0	0	0	0	0
48643	0	0	0	0	0	0	0
48644	0	0	0	0	0	0	0

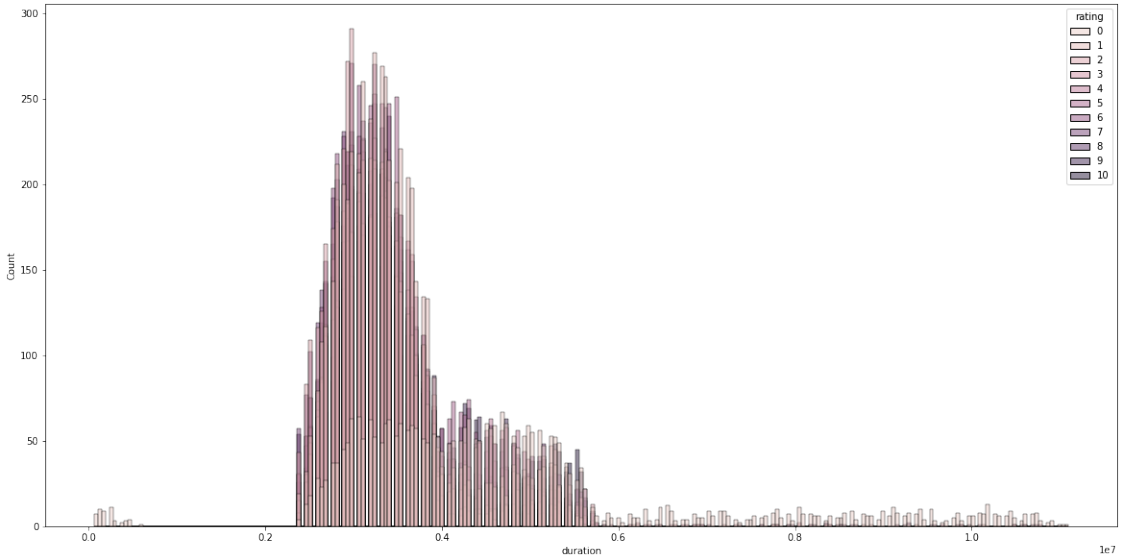
[48645 rows x 41 columns]

1 EDA

2 Histogram

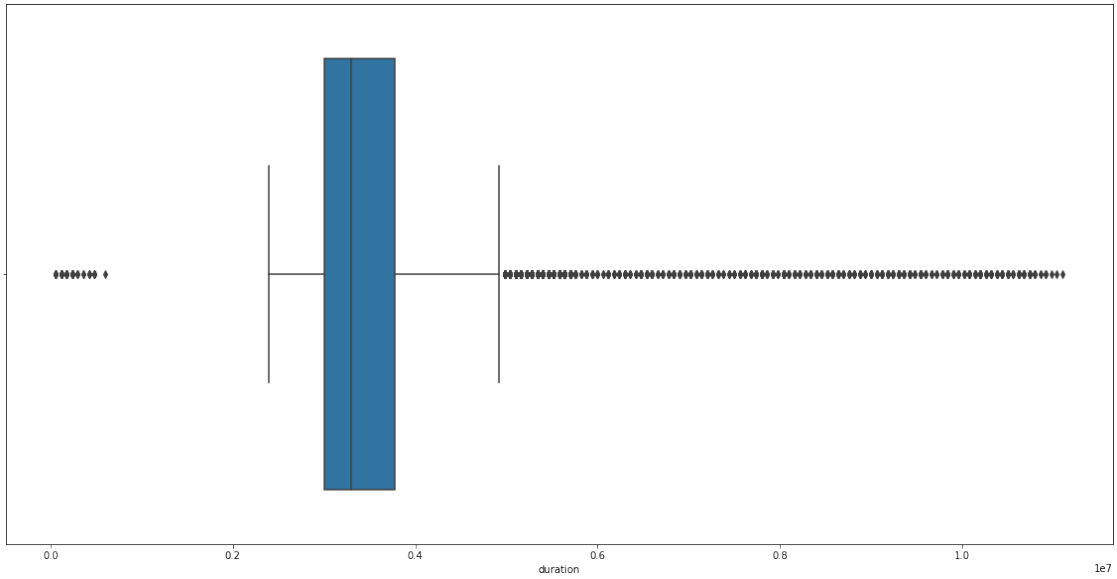
[20]:
```
plt.figure(figsize=(20, 10))
sns.histplot(x="duration", data=df13, hue="rating")
```

[20]: <AxesSubplot:xlabel='duration', ylabel='Count'>



[21]:
```
plt.figure(figsize=(15, 10))
sns.boxplot(x="duration", data=df13)
```

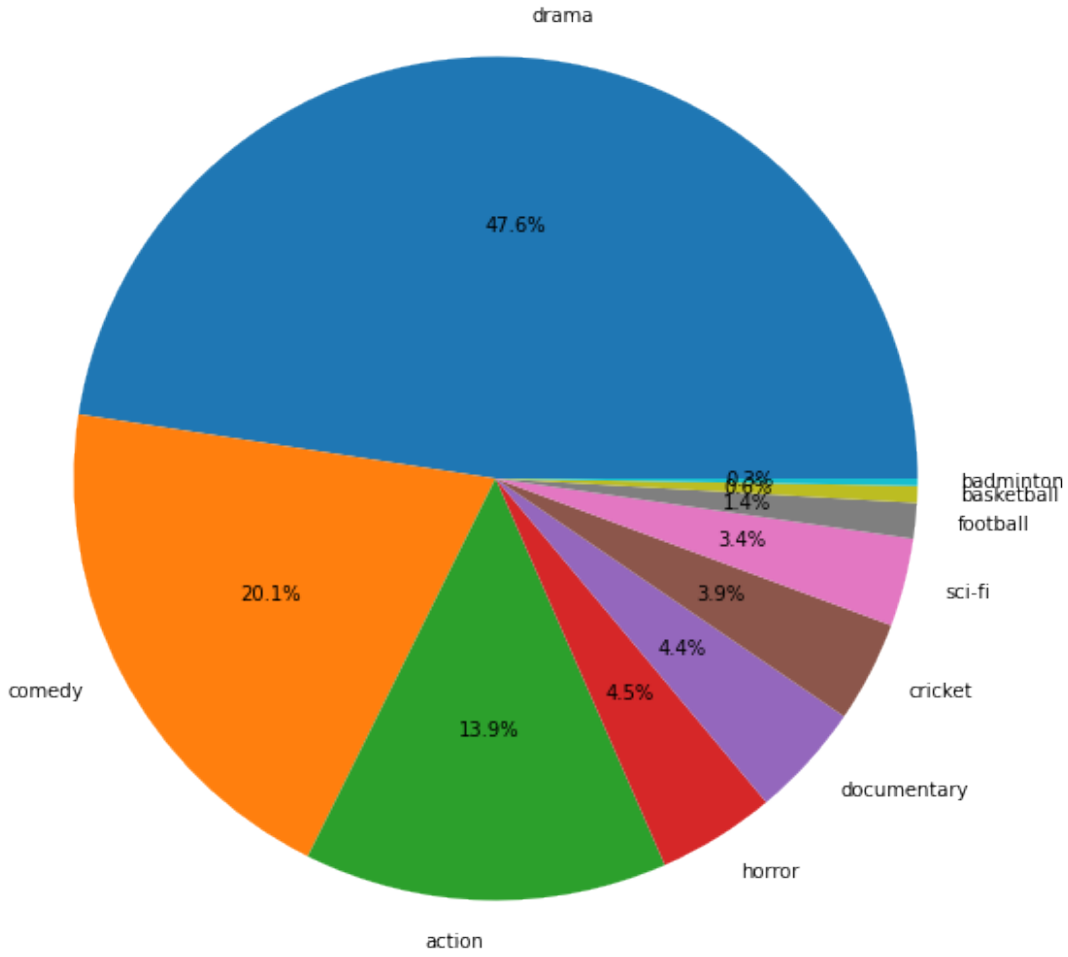
[21]: <AxesSubplot:xlabel='duration'>



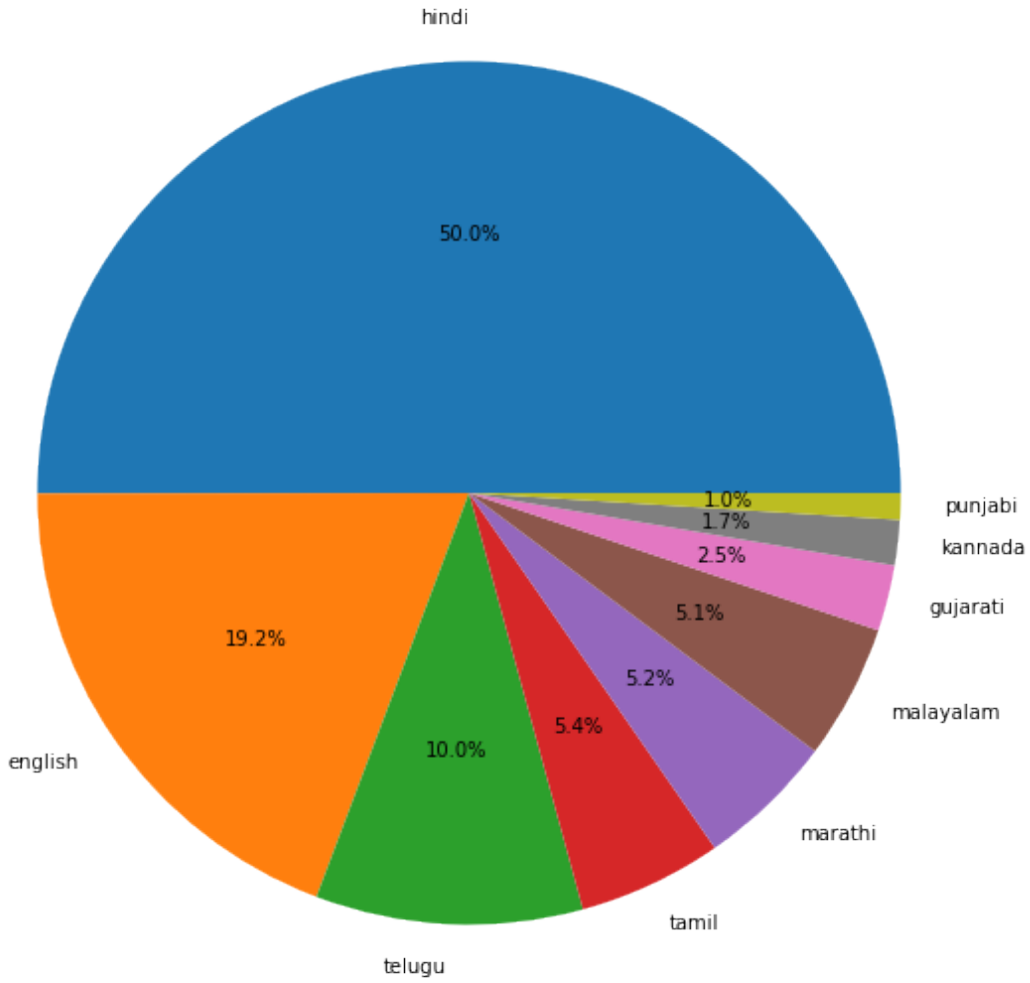
[22]:
```
df13.language.value_counts()
```

[22]:
```
hindi        23912
english       9194
telugu        4781
tamil         2577
marathi       2465
malayalam     2415
gujarati      1179
kannada        810
punjabi        474
bengali        454
oriya          384
Name: language, dtype: int64
```

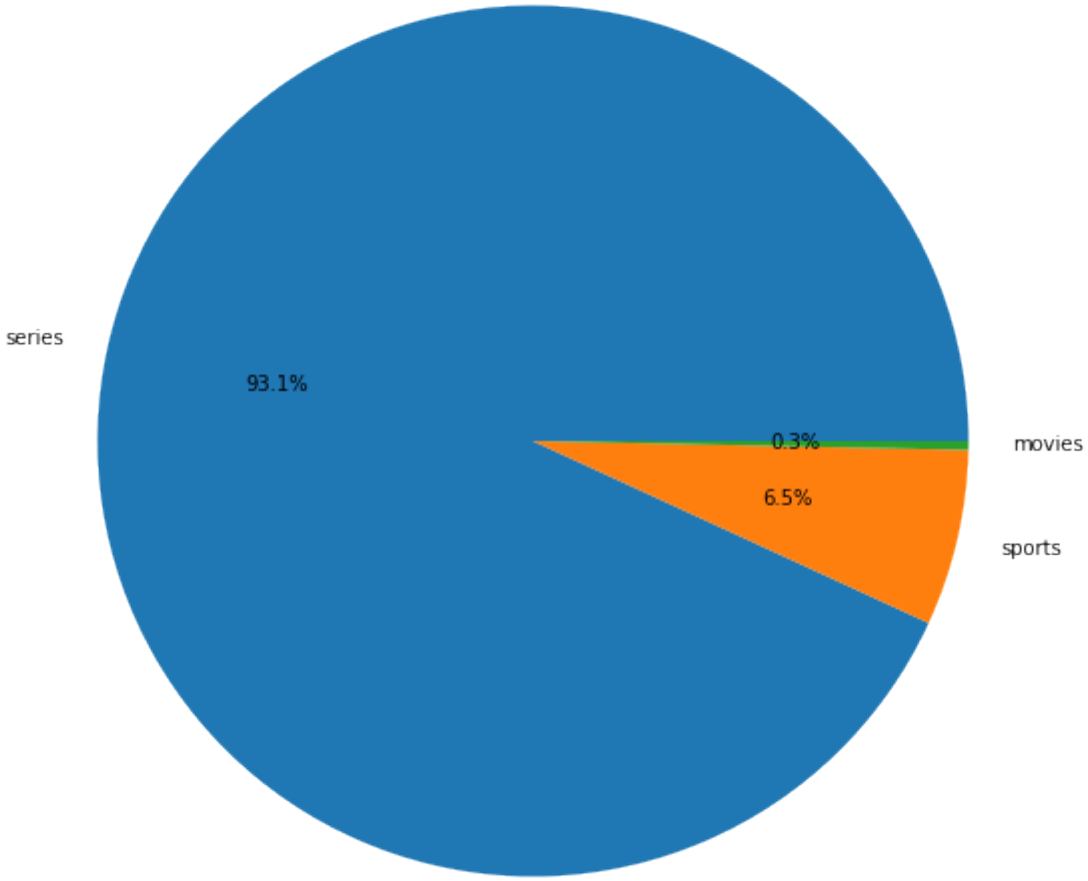
[23]:
```
plt.figure(figsize=(20, 10))
plt.pie(df13.genre.value_counts()[:10], labels=df13.genre.value_counts()[:10].
 ↪keys(), autopct="%.1f%%")
plt.show()
```



[24]:
```
plt.figure(figsize=(20, 10))
plt.pie(df13.language.value_counts()[:9], labels=df13.language.value_counts().
↪keys()[:9], autopct="%.1f%%")
plt.show()
```

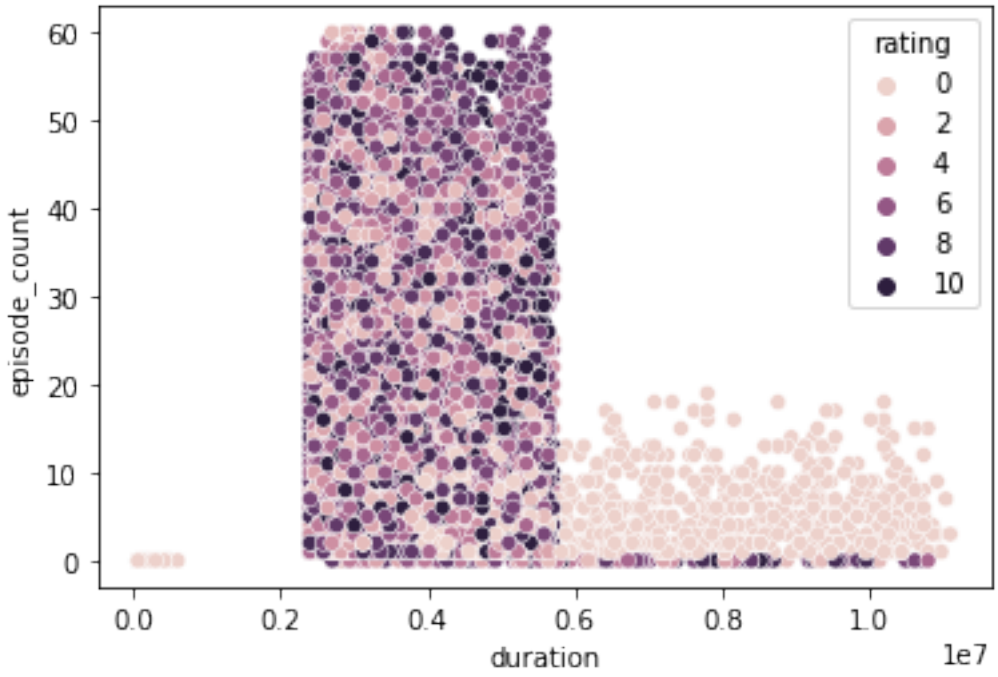


[25]:
```
plt.figure(figsize=(20, 10))
plt.pie(df13.content_type.value_counts()[:3], labels=df13.content_type.
↪value_counts()[:3].keys(), autopct="%.1f%%")
plt.show()
```



[26]:
```
sns.scatterplot(x="duration", y="episode_count", data=df13, hue="rating")
```

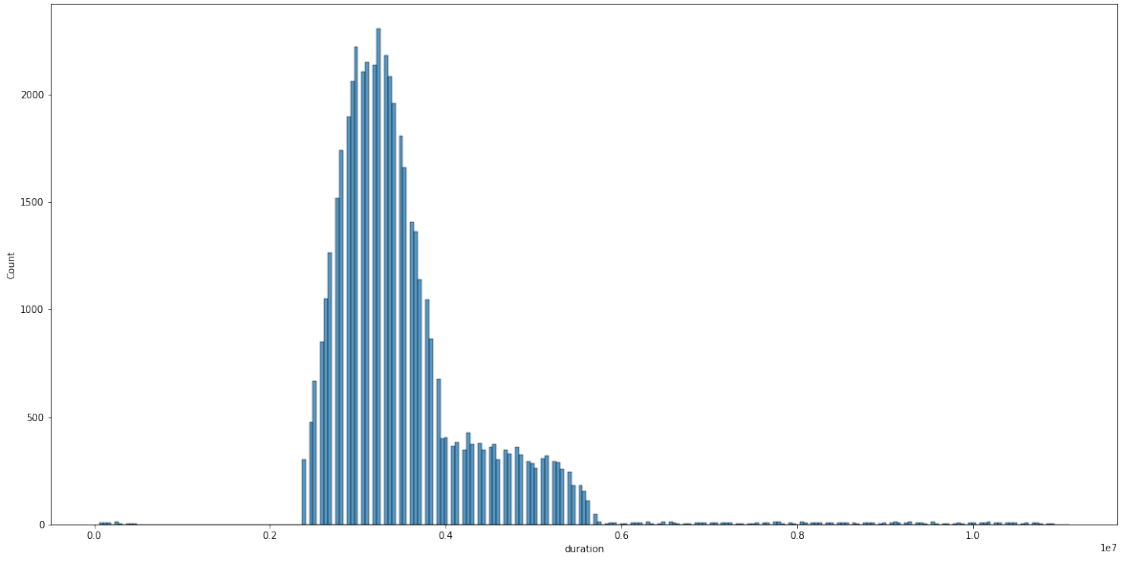
[26]:
```
<AxesSubplot:xlabel='duration', ylabel='episode_count'>
```



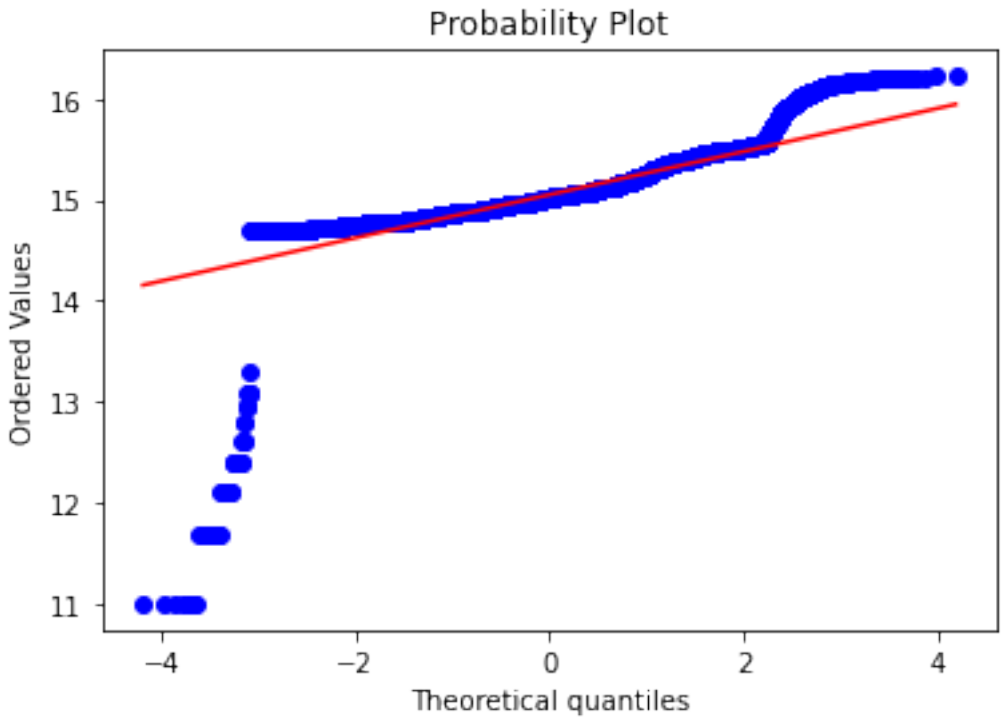
3 feature transformation

[27]:
```
plt.figure(figsize=(20, 10))
sns.histplot(x="duration", data=df13)
```

[27]: <AxesSubplot:xlabel='duration', ylabel='Count'>

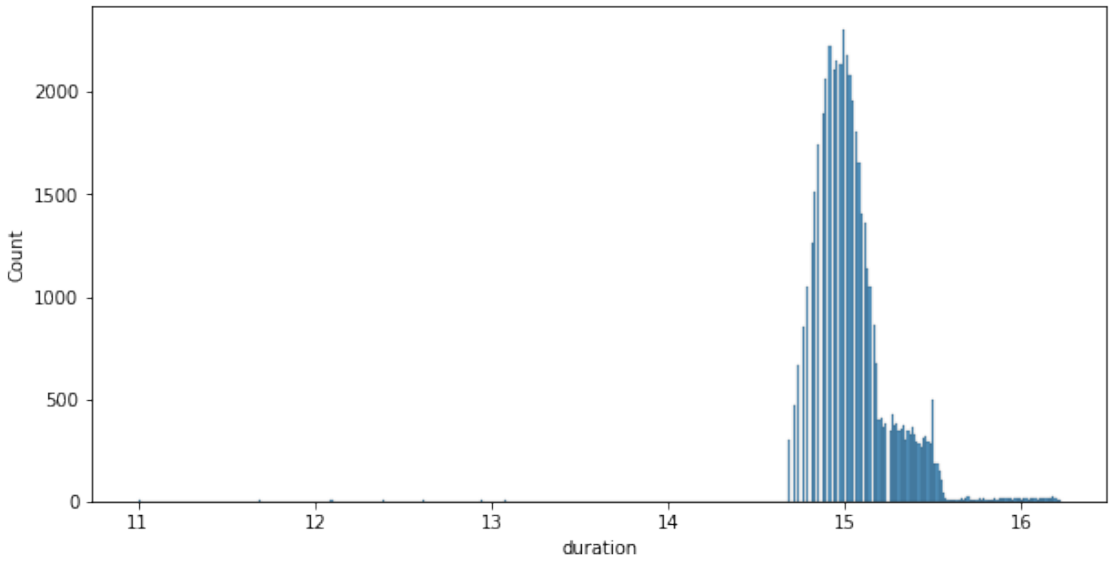


[28]:
```
stats.probplot(np.log(df13.duration), dist="norm", plot=pylab)
pylab.show()
```



[29]:
```
plt.figure(figsize=(10, 5))
sns.histplot(np.log(df13.duration))
```

[29]:
```
<AxesSubplot:xlabel='duration', ylabel='Count'>
```



[30]:
```
df13["normal_duration"] = np.log(df13.duration)
# df13["normal_duration"] = 1/(df13.duration)
```

[31]:
```
df13
```

[31]:

	content_type	language	genre	duration	rating	episode_count	\
0	series	english	drama	4980000	10	32	
1	series	english	drama	3000000	4	21	
2	series	tamil	comedy	3120000	8	28	
3	sports	hindi	cricket	9900000	0	5	
4	series	hindi	action	3660000	2	29	
...	...	...	...	...	...	...	
48640	series	hindi	drama	3360000	6	15	
48641	series	marathi	sci-fi	3120000	4	14	
48642	series	hindi	drama	3180000	6	5	
48643	series	english	drama	4020000	5	1	
48644	series	english	horror	2760000	8	4	

	season_count	movies	series	sports	...	english	gujarati	hindi	\
0	19	0	1	0	...	1	0	0	
1	15	0	1	0	...	1	0	0	
2	13	0	1	0	...	0	0	0	
3	1	0	0	1	...	0	0	1	
4	10	0	1	0	...	0	0	1	
...	...	...	...	...	...	...	...	...	
48640	6	0	1	0	...	0	0	1	
48641	1	0	1	0	...	0	0	0	
48642	33	0	1	0	...	0	0	1	

```
48643            9       0       1       0  ...        1         0      0
48644            2       0       1       0  ...        1         0      0

       kannada  malayalam  marathi  oriya  punjabi  tamil  normal_duration
0            0          0        0      0        0      0        15.420940
1            0          0        0      0        0      0        14.914123
2            0          0        0      0        0      1        14.953344
3            0          0        0      0        0      0        16.108045
4            0          0        0      0        0      0        15.112974
...        ...        ...      ...    ...      ...    ...              ...
48640        0          0        0      0        0      0        15.027452
48641        0          0        1      0        0      0        14.953344
48642        0          0        0      0        0      0        14.972392
48643        0          0        0      0        0      0        15.206792
48644        0          0        0      0        0      0        14.830741

[48645 rows x 42 columns]
```

[32]:
```
droplist = ["content_type", "language", "genre", "duration"]
```

[33]:
```
df13.drop(droplist, axis=1, inplace=True)
```

[34]:
```
df13
```

[34]:
```
       rating  episode_count  season_count  movies  series  sports  action  \
0          10             32            19       0       1       0       0
1           4             21            15       0       1       0       0
2           8             28            13       0       1       0       0
3           0              5             1       0       0       1       0
4           2             29            10       0       1       0       1
...       ...            ...           ...     ...     ...     ...     ...
48640       6             15             6       0       1       0       0
48641       4             14             1       0       1       0       0
48642       6              5            33       0       1       0       0
48643       5              1             9       0       1       0       0
48644       8              4             2       0       1       0       0

       adventure  animation  badminton  ...  english  gujarati  hindi  \
0              0          0          0  ...        1         0      0
1              0          0          0  ...        1         0      0
2              0          0          0  ...        0         0      0
3              0          0          0  ...        0         0      1
4              0          0          0  ...        0         0      1
...          ...        ...        ...  ...      ...       ...    ...
48640          0          0          0  ...        0         0      1
48641          0          0          0  ...        0         0      0
48642          0          0          0  ...        0         0      1
```

```
48643          0          0          0  ...        1         0      0
48644          0          0          0  ...        1         0      0

       kannada  malayalam  marathi  oriya  punjabi  tamil  normal_duration
0            0          0        0      0        0      0        15.420940
1            0          0        0      0        0      0        14.914123
2            0          0        0      0        0      1        14.953344
3            0          0        0      0        0      0        16.108045
4            0          0        0      0        0      0        15.112974
...        ...        ...      ...    ...      ...    ...              ...
48640        0          0        0      0        0      0        15.027452
48641        0          0        1      0        0      0        14.953344
48642        0          0        0      0        0      0        14.972392
48643        0          0        0      0        0      0        15.206792
48644        0          0        0      0        0      0        14.830741

[48645 rows x 38 columns]
```

4 Outlier dect

using z score

[35]:
```
z = np.abs(df13.normal_duration-df13.normal_duration.mean()) / \
    df13.normal_duration.std()
```

[36]:
```
df13["ob"] = z[z > 3]
#ob is set of outlier
```

[37]:
```
df13
```

[37]:
```
       rating  episode_count  season_count  movies  series  sports  action  \
0          10             32            19       0       1       0       0
1           4             21            15       0       1       0       0
2           8             28            13       0       1       0       0
3           0              5             1       0       0       1       0
4           2             29            10       0       1       0       1
...       ...            ...           ...     ...     ...     ...     ...
48640       6             15             6       0       1       0       0
48641       4             14             1       0       1       0       0
48642       6              5            33       0       1       0       0
48643       5              1             9       0       1       0       0
48644       8              4             2       0       1       0       0

       adventure  animation  badminton  ...  gujarati  hindi  kannada  \
0              0          0          0  ...         0      0        0
1              0          0          0  ...         0      0        0
2              0          0          0  ...         0      0        0
```

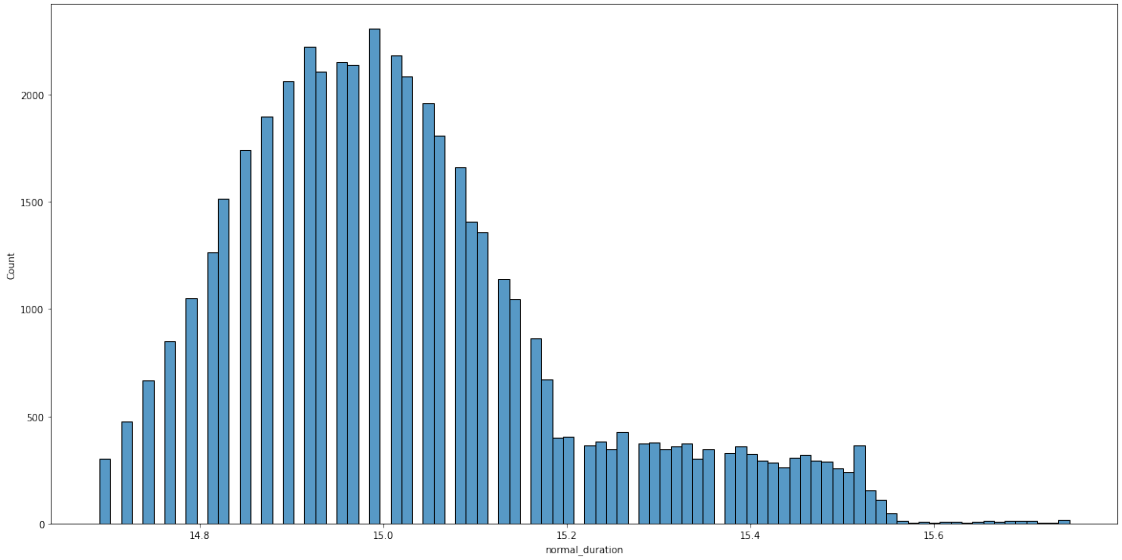
```
3              0          0          0  ...          0      1        0
4              0          0          0  ...          0      1        0
...          ...        ...        ...  ...        ...    ...      ...
48640          0          0          0  ...          0      1        0
48641          0          0          0  ...          0      0        0
48642          0          0          0  ...          0      1        0
48643          0          0          0  ...          0      0        0
48644          0          0          0  ...          0      0        0

       malayalam  marathi  oriya  punjabi  tamil  normal_duration        ob
0              0        0      0        0      0        15.420940       NaN
1              0        0      0        0      0        14.914123       NaN
2              0        0      0        0      1        14.953344       NaN
3              0        0      0        0      0        16.108045  4.526122
4              0        0      0        0      0        15.112974       NaN
...          ...      ...    ...      ...    ...              ...       ...
48640          0        0      0        0      0        15.027452       NaN
48641          0        1      0        0      0        14.953344       NaN
48642          0        0      0        0      0        14.972392       NaN
48643          0        0      0        0      0        15.206792       NaN
48644          0        0      0        0      0        14.830741       NaN

[48645 rows x 39 columns]
```

[38]:
```
a = []
for i in range(0, len(df13["ob"])):
    if df13["ob"][i] > 3:
        a.append(i)
```

[39]:
```
df13.drop(a, inplace=True)
#outlier removed
```

[40]:
```
plt.figure(figsize=(20, 10))
sns.histplot(df13.normal_duration)
```

[40]:
```
<AxesSubplot:xlabel='normal_duration', ylabel='Count'>
```



[41]:
```
df13.drop(["ob"], axis="columns", inplace=True)
```

[42]:
```
df13
```

[42]:

	rating	episode_count	season_count	movies	series	sports	action	\
0	10	32	19	0	1	0	0	
1	4	21	15	0	1	0	0	
2	8	28	13	0	1	0	0	
4	2	29	10	0	1	0	1	
5	10	37	1	0	1	0	0	
...	...	...	...	...	...	...	...	
48640	6	15	6	0	1	0	0	
48641	4	14	1	0	1	0	0	
48642	6	5	33	0	1	0	0	
48643	5	1	9	0	1	0	0	
48644	8	4	2	0	1	0	0	

	adventure	animation	badminton	...	english	gujarati	hindi	\
0	0	0	0	...	1	0	0	
1	0	0	0	...	1	0	0	
2	0	0	0	...	0	0	0	
4	0	0	0	...	0	0	1	
5	0	0	0	...	0	0	1	
...	...	...	...	...	...	...	...	
48640	0	0	0	...	0	0	1	
48641	0	0	0	...	0	0	0	
48642	0	0	0	...	0	0	1	
48643	0	0	0	...	1	0	0	

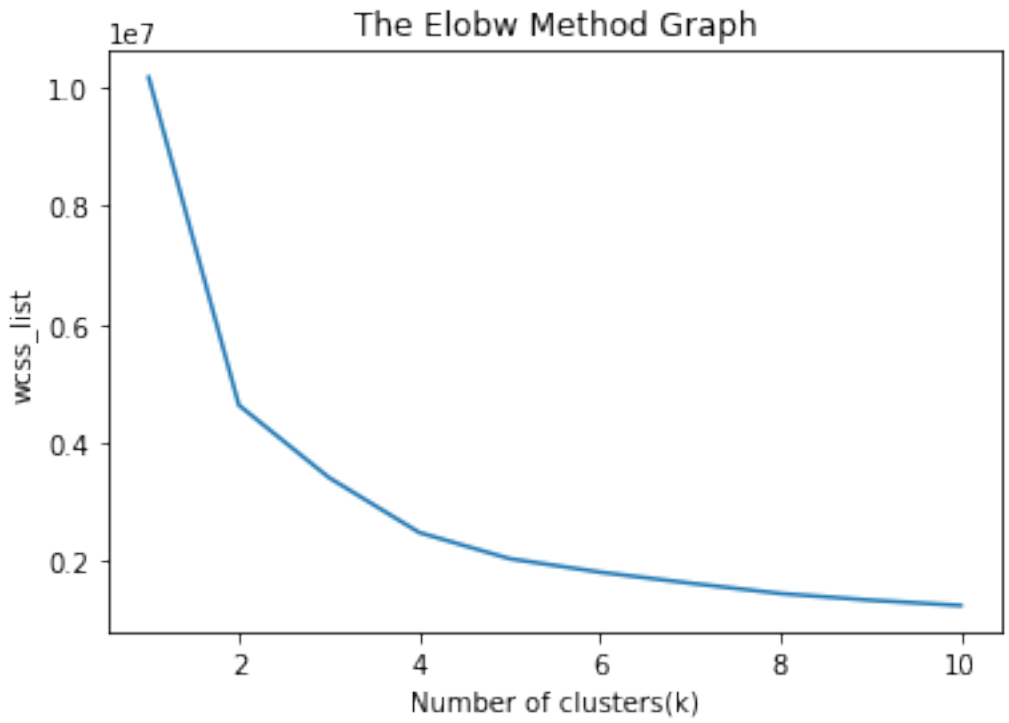
```
48644          0          0          0  ...         1         0      0

       kannada  malayalam  marathi  oriya  punjabi  tamil  normal_duration
0            0          0        0      0        0      0        15.420940
1            0          0        0      0        0      0        14.914123
2            0          0        0      0        0      1        14.953344
4            0          0        0      0        0      0        15.112974
5            0          0        0      0        0      0        14.933925
...        ...        ...      ...    ...      ...    ...              ...
48640        0          0        0      0        0      0        15.027452
48641        0          0        1      0        0      0        14.953344
48642        0          0        0      0        0      0        14.972392
48643        0          0        0      0        0      0        15.206792
48644        0          0        0      0        0      0        14.830741

[48113 rows x 38 columns]
```

5 Kmeans clustering

[43]:
```
from sklearn.cluster import KMeans
```

[44]:
```
x = df13
```

[45]:
```
from sklearn.cluster import KMeans
from sklearn import metrics
from scipy.spatial.distance import cdist
import numpy as np
import matplotlib.pyplot as plt
wcss_list= []  #Initializing the list for the values of WCSS

#Using for loop for iterations from 1 to 10.
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i, init='k-means++', random_state= 42)
    kmeans.fit(x)
    wcss_list.append(kmeans.inertia_)
plt.plot(range(1, 11), wcss_list)
plt.title('The Elobw Method Graph')
plt.xlabel('Number of clusters(k)')
plt.ylabel('wcss_list')
plt.show()
```



[46]:
```
kmeans = KMeans(n_clusters=4, random_state=0).fit(x)
```

[47]:
```
kmeans.cluster_centers_
```

[47]:
```
array([[ 5.22073237e+00,  2.15353934e+01,  5.79703240e+00,
        -2.68882139e-17,  9.91015519e-01,  8.98448135e-03,
         1.51102641e-01, -2.34458720e-18,  2.81892565e-18,
         3.40321263e-04,  3.40321263e-04, -2.41234983e-18,
         2.09501770e-01,  5.37707596e-03, -4.82469967e-18,
         4.71004628e-02,  5.05513204e-01, -4.68917440e-18,
        -4.82469967e-18,  2.17805608e-03,  2.04192758e-04,
         4.67601416e-02, -1.17229360e-18, -1.17229360e-18,
         3.10372992e-02, -1.17229360e-18,  5.44514021e-04,
         7.75932480e-03,  1.76150286e-01,  2.75660223e-02,
         4.94690988e-01,  1.64034849e-02,  4.87340049e-02,
         4.36291860e-02,  7.14674653e-03,  8.23577457e-03,
         6.03729921e-02,  1.50341519e+01],
       [ 4.70429477e+00,  6.58023938e+00,  4.60023469e+00,
         4.55292185e-03,  8.72330439e-01,  1.23116639e-01,
         1.14527106e-01,  1.87749355e-04,  3.28561371e-04,
         4.55292185e-03,  1.19220840e-02,  1.40812016e-04,
         1.89486036e-01,  7.31283736e-02,  2.81624032e-04,
         4.27129782e-02,  4.48439333e-01,  3.75498709e-04,
```

```
  2.81624032e-04,  2.57685989e-02,  4.03661112e-03,
  4.21497301e-02,  9.38746773e-05,  9.38746773e-05,
  3.73151842e-02,  9.38746773e-05,  3.70804975e-03,
  1.01384651e-02,  2.01361183e-01,  2.70359071e-02,
  4.88336071e-01,  1.59586951e-02,  4.97535790e-02,
  4.57169678e-02,  7.88547289e-03,  9.38746773e-03,
  5.05045764e-02,  1.50508636e+01],
[ 5.50291181e+00,  3.97537438e+01,  5.37118691e+00,
  4.42354486e-17,  1.00000000e+00, -1.06858966e-15,
  1.86633389e-01,  1.49077799e-19, -1.70761842e-18,
  3.81639165e-17,  5.98479599e-17, -1.42301535e-18,
  1.99805879e-01,  2.91433544e-16, -2.84603070e-18,
  4.61730449e-02,  5.00831947e-01,  2.98155597e-19,
 -2.84603070e-18,  1.07552856e-16, -2.16840434e-17,
  4.56184138e-02,  7.45388994e-20,  7.45388994e-20,
  2.09373267e-02,  7.45388994e-20, -1.25767452e-17,
  6.37825846e-03,  1.45729340e-01,  1.15085968e-02,
  5.19412091e-01,  1.78868552e-02,  3.36938436e-02,
  8.20854132e-02,  1.09539656e-02,  1.41430948e-02,
  6.07321131e-02,  1.50345333e+01],
[ 5.76407015e+00,  8.79547308e+00,  2.02601958e+01,
  1.51788304e-17,  1.00000000e+00, -6.93889390e-17,
  1.43352365e-01,  1.76182853e-19, -1.97866896e-18,
  3.12250226e-17, -1.82145965e-17, -5.14996032e-19,
  2.32463295e-01,  2.70616862e-16, -1.02999206e-18,
  3.67047308e-02,  4.87969005e-01,  3.52365706e-19,
 -1.02999206e-18, -6.93889390e-17, -1.56125113e-17,
  5.34257749e-02,  8.80914265e-20,  8.80914265e-20,
  4.60848287e-02,  8.80914265e-20, -1.19262239e-17,
  1.52936378e-02,  2.09624796e-01,  2.26345840e-02,
  4.53915171e-01,  2.01876020e-02,  8.07504078e-02,
  5.15905383e-02,  6.32137031e-03,  1.03996737e-02,
  3.48694943e-02,  1.50386752e+01]])
```

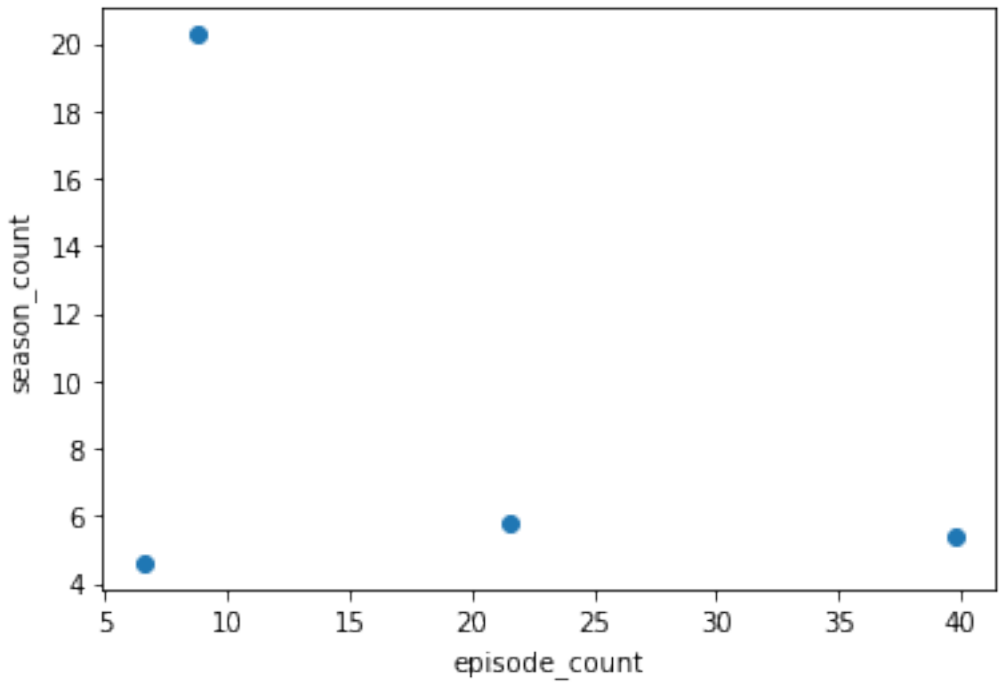
[48]:
```
df13["cluster"] = kmeans.predict(x)
```

[49]:
```
x = kmeans.cluster_centers_[:, 1]
```

[50]:
```
y = kmeans.cluster_centers_[:, 2]
```

[51]:
```
plt.scatter(x, y)
plt.xlabel("episode_count")
plt.ylabel("season_count")
```

[51]:
```
Text(0, 0.5, 'season_count')
```



[52]:
```
df1 = pd.read_csv(r"content.csv")
```

[53]:
```
df1
```

[53]:

	content_id	content_type	language	genre	duration	release_date	rating	episode_count	season_count
0	cont_475_19_32	series	english	drama	4980000	2018-07-01	10	32	19
1	cont_2185_15_21	series	english	drama	3000000	2016-03-29	4	21	15
2	cont_4857_13_28	series	tamil	comedy	3120000	2006-03-06	8	28	13
3	cont_3340_1_5	sports	hindi	cricket	9900000	2009-01-10	0	5	1
4	cont_1664_10_29	series	hindi	action	3660000	2020-05-25	2	29	10
...	...	...	...	...	...	...	...	...	...
48640	cont_4218_6_15	series	hindi	drama	3360000	2015-02-04			
48641	cont_2533_1_14	series	marathi	sci-fi	3120000	2002-01-15			
48642	cont_4606_33_5	series	hindi	drama	3180000	2006-02-18			
48643	cont_3708_9_1	series	english	drama	4020000	2010-04-12			
48644	cont_3470_2_4	series	english	horror	2760000	1997-03-26			

```
48640       6              15              6
48641       4              14              1
48642       6               5             33
48643       5               1              9
48644       8               4              2

[48645 rows x 9 columns]
```

6 Checking Correlation

[54]:
```
df13.corr()
```

[54]:
```
                  rating  episode_count  season_count    movies    series  \
rating          1.000000       0.080658      0.113670  0.005182  0.401427
episode_count   0.080658       1.000000     -0.098487 -0.057595  0.212568
season_count    0.113670      -0.098487      1.000000 -0.048768  0.232964
movies          0.005182      -0.057595     -0.048768  1.000000 -0.179052
series          0.401427       0.212568      0.232964 -0.179052  1.000000
sports         -0.408996      -0.204927     -0.227361 -0.011077 -0.981796
action          0.030891       0.076420      0.024063  0.000634  0.097489
adventure       0.006423      -0.011684     -0.009894  0.202873 -0.036325
animation       0.007513      -0.015458     -0.013088  0.268384 -0.048055
badminton      -0.076492      -0.039331     -0.042522 -0.002072 -0.183619
basketball     -0.122089      -0.064945     -0.067869 -0.003307 -0.293075
biography       0.007493      -0.010119     -0.008568  0.175691 -0.031458
comedy          0.039144       0.011787      0.031856 -0.002943  0.122379
cricket        -0.311455      -0.155980     -0.173138 -0.008435 -0.747648
crime          -0.000327      -0.014311     -0.012117  0.248473 -0.044490
documentary     0.033428       0.011608     -0.005449 -0.000597  0.052109
drama           0.097922       0.043073      0.054142 -0.037423  0.239033
family         -0.000377      -0.016525     -0.013992  0.286917 -0.051373
fantasy         0.002101      -0.014311     -0.012117  0.248473 -0.044490
football       -0.183476      -0.089668     -0.101994 -0.004969 -0.440436
hockey         -0.071442      -0.036659     -0.039714 -0.001935 -0.171496
horror          0.008406      -0.000334      0.020526  0.003597  0.052092
musical         0.000862      -0.008262     -0.006996  0.143450 -0.025685
mystery        -0.002291      -0.008262     -0.006996  0.143450 -0.025685
sci-fi          0.060079      -0.033933      0.014385 -0.003286  0.046003
sport           0.004016      -0.008262     -0.006996  0.143450 -0.025685
tennis         -0.070633      -0.033062     -0.039265 -0.001913 -0.169554
bengali         0.042514      -0.021826      0.015473  0.000436  0.023505
english        -0.080065      -0.060837     -0.011581 -0.001260 -0.187695
gujarati        0.043447      -0.024130     -0.000656  0.007898  0.036865
hindi           0.010255       0.027815     -0.025685  0.001234 -0.010291
kannada        -0.020218       0.006065      0.008483 -0.005878  0.032828
malayalam       0.003703      -0.032441      0.058530 -0.010332  0.057706
```

```
marathi          -0.001591       0.052725     -0.002844 -0.002019  0.056672
oriya             0.036806       0.015305     -0.014117 -0.004026  0.022486
punjabi           0.010736       0.010415      0.003006 -0.004483  0.025039
tamil            -0.007585       0.019454      0.001797  0.001678  0.057307
normal_duration  -0.069210      -0.037832     -0.038589  0.068853 -0.238329
cluster           0.056126       0.032503      0.497782 -0.002352  0.025658

                    sports    action  adventure  animation  badminton  ...  \
rating           -0.408996  0.030891   0.006423   0.007513  -0.076492  ...
episode_count    -0.204927  0.076420  -0.011684  -0.015458  -0.039331  ...
season_count     -0.227361  0.024063  -0.009894  -0.013088  -0.042522  ...
movies           -0.011077  0.000634   0.202873   0.268384  -0.002072  ...
series           -0.981796  0.097489  -0.036325  -0.048055  -0.183619  ...
sports            1.000000 -0.099207  -0.002247  -0.002973   0.187023  ...
action           -0.099207  1.000000  -0.003670  -0.004856  -0.018554  ...
adventure        -0.002247 -0.003670   1.000000  -0.000110  -0.000420  ...
animation        -0.002973 -0.004856  -0.000110   1.000000  -0.000556  ...
badminton         0.187023 -0.018554  -0.000420  -0.000556   1.000000  ...
basketball        0.298509 -0.029614  -0.000671  -0.000887  -0.003391  ...
biography        -0.001946 -0.003179  -0.000072  -0.000095  -0.000364  ...
comedy           -0.123814 -0.202228  -0.004581  -0.006060  -0.023156  ...
cricket           0.761511 -0.075547  -0.001711  -0.002264  -0.008650  ...
crime            -0.002752 -0.004496  -0.000102  -0.000135  -0.000515  ...
documentary      -0.052847 -0.086316  -0.001955  -0.002587  -0.009884  ...
drama            -0.235719 -0.385007  -0.008721  -0.011537  -0.044085  ...
family           -0.003178 -0.005191  -0.000118  -0.000156  -0.000594  ...
fantasy          -0.002752 -0.004496  -0.000102  -0.000135  -0.000515  ...
football          0.448602 -0.044504  -0.001008  -0.001334  -0.005096  ...
hockey            0.174675 -0.017329  -0.000393  -0.000519  -0.001984  ...
horror           -0.053639 -0.087610  -0.001985  -0.002625  -0.010032  ...
musical          -0.001589 -0.002595  -0.000059  -0.000078  -0.000297  ...
mystery          -0.001589 -0.002595  -0.000059  -0.000078  -0.000297  ...
sci-fi           -0.046122 -0.075332  -0.001706  -0.002257  -0.008626  ...
sport            -0.001589 -0.002595  -0.000059  -0.000078  -0.000297  ...
tennis            0.172698 -0.017133  -0.000388  -0.000513  -0.001962  ...
bengali          -0.023974  0.000692  -0.000887  -0.001173  -0.004484  ...
english           0.191009  0.015108   0.001495  -0.001342   0.022071  ...
gujarati         -0.038993 -0.055526  -0.001443  -0.001909  -0.007293  ...
hindi             0.010221  0.007083  -0.004403  -0.001517   0.012541  ...
kannada          -0.032230  0.083152  -0.001192  -0.001578  -0.006028  ...
malayalam        -0.056656 -0.077699  -0.002096  -0.002773  -0.010596  ...
marathi          -0.057209  0.046845  -0.002117  -0.002800  -0.010699  ...
oriya            -0.022077  0.051714  -0.000817  -0.001081  -0.004129  ...
punjabi          -0.024583 -0.016461  -0.000910  -0.001203  -0.004598  ...
tamil            -0.058569  0.049707  -0.002167   0.004794  -0.010954  ...
normal_duration   0.228936 -0.012892   0.018724   0.017119   0.047451  ...
cluster          -0.025624  0.013480  -0.000477  -0.000631  -0.004847  ...
```

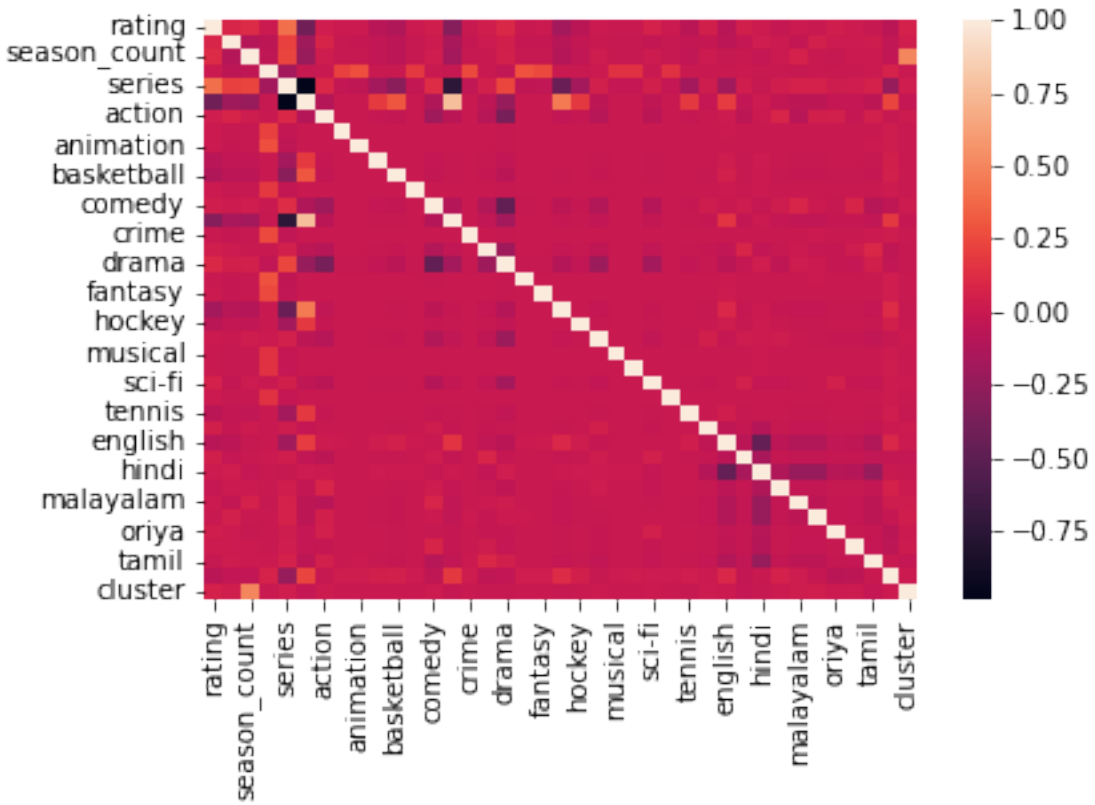
	gujarati	hindi	kannada	malayalam	marathi	oriya	\
rating	0.043447	0.010255	-0.020218	0.003703	-0.001591	0.036806	
episode_count	-0.024130	0.027815	0.006065	-0.032441	0.052725	0.015305	
season_count	-0.000656	-0.025685	0.008483	0.058530	-0.002844	-0.014117	
movies	0.007898	0.001234	-0.005878	-0.010332	-0.002019	-0.004026	
series	0.036865	-0.010291	0.032828	0.057706	0.056672	0.022486	
sports	-0.038993	0.010221	-0.032230	-0.056656	-0.057209	-0.022077	
action	-0.055526	0.007083	0.083152	-0.077699	0.046845	0.051714	
adventure	-0.001443	-0.004403	-0.001192	-0.002096	-0.002117	-0.000817	
animation	-0.001909	-0.001517	-0.001578	-0.002773	-0.002800	-0.001081	
badminton	-0.007293	0.012541	-0.006028	-0.010596	-0.010699	-0.004129	
basketball	-0.011640	0.006091	-0.009621	-0.016912	-0.017077	-0.006590	
biography	-0.001249	0.002768	-0.001033	-0.001815	-0.001833	-0.000707	
comedy	0.019875	-0.020093	0.018523	0.083130	0.001940	-0.011769	
cricket	-0.029694	0.004021	-0.024543	-0.043144	-0.043566	-0.016812	
crime	-0.001767	-0.003532	-0.001460	-0.002567	0.005858	-0.001000	
documentary	0.079049	-0.041649	-0.013060	0.002247	-0.017555	-0.019208	
drama	-0.017349	0.029137	-0.051618	0.017958	0.002972	-0.018719	
family	-0.002040	-0.003003	-0.001686	-0.002965	0.004325	-0.001155	
fantasy	-0.001767	-0.003532	-0.001460	-0.002567	-0.002592	-0.001000	
football	-0.017492	0.002086	-0.014458	-0.025416	-0.025664	-0.009904	
hockey	-0.006811	0.009930	-0.005630	-0.009896	-0.009993	-0.003856	
horror	0.017404	0.011135	0.020546	-0.003767	-0.006475	-0.019496	
musical	-0.001020	0.006559	-0.000843	-0.001482	-0.001497	-0.000578	
mystery	-0.001020	0.006559	-0.000843	-0.001482	-0.001497	-0.000578	
sci-fi	0.050821	-0.028062	-0.024474	-0.009327	-0.001169	0.051794	
sport	-0.001020	0.000111	-0.000843	-0.001482	-0.001497	-0.000578	
tennis	-0.006734	-0.000738	-0.005566	-0.009784	-0.009880	-0.003813	
bengali	-0.015391	-0.095621	-0.012721	-0.022362	-0.022581	-0.008714	
english	-0.075673	-0.470148	-0.062547	-0.109949	-0.111024	-0.042844	
gujarati	1.000000	-0.155528	-0.020691	-0.036372	-0.036727	-0.014173	
hindi	-0.155528	1.000000	-0.128552	-0.225976	-0.228184	-0.088055	
kannada	-0.020691	-0.128552	1.000000	-0.030063	-0.030357	-0.011715	
malayalam	-0.036372	-0.225976	-0.030063	1.000000	-0.053363	-0.020593	
marathi	-0.036727	-0.228184	-0.030357	-0.053363	1.000000	-0.020794	
oriya	-0.014173	-0.088055	-0.011715	-0.020593	-0.020794	1.000000	
punjabi	-0.015782	-0.098053	-0.013045	-0.022931	-0.023155	-0.008935	
tamil	-0.037600	-0.233607	-0.031078	-0.054632	-0.055165	-0.021288	
normal_duration	-0.031217	-0.000920	0.059154	0.023192	-0.015152	-0.071638	
cluster	-0.022794	-0.009523	0.008145	0.020708	0.034446	0.004390	

	punjabi	tamil	normal_duration	cluster
rating	0.010736	-0.007585	-0.069210	0.056126
episode_count	0.010415	0.019454	-0.037832	0.032503
season_count	0.003006	0.001797	-0.038589	0.497782
movies	-0.004483	0.001678	0.068853	-0.002352

```
series           0.025039  0.057307        -0.238329  0.025658
sports          -0.024583 -0.058569         0.228936 -0.025624
action          -0.016461  0.049707        -0.012892  0.013480
adventure       -0.000910 -0.002167         0.018724 -0.000477
animation       -0.001203  0.004794         0.017119 -0.000631
badminton       -0.004598 -0.010954         0.047451 -0.004847
basketball      -0.007338 -0.017483         0.053310 -0.005380
biography       -0.000788 -0.001877         0.016253 -0.000413
comedy           0.071072 -0.087373        -0.050685  0.009744
cricket         -0.018721 -0.044601         0.175467 -0.019583
crime           -0.001114  0.005621         0.018248 -0.000584
documentary      0.023778  0.100932        -0.050296 -0.011000
drama           -0.027997  0.046102        -0.038612 -0.006394
family          -0.001286 -0.003065         0.027347 -0.000675
fantasy         -0.001114 -0.002654         0.026120 -0.000584
football        -0.011028 -0.026274         0.111001 -0.012349
hockey          -0.004294 -0.010231         0.043073 -0.003817
horror          -0.021710 -0.037939        -0.013887  0.006512
musical         -0.000643 -0.001532         0.013533 -0.000337
mystery         -0.000643 -0.001532         0.021630 -0.000337
sci-fi          -0.018667 -0.024033         0.012909  0.008426
sport           -0.000643 -0.001532         0.009436 -0.000337
tennis          -0.004246 -0.010115         0.031127 -0.006446
bengali         -0.009703 -0.023117         0.005494  0.013977
english         -0.047708 -0.113662         0.086802  0.004853
gujarati        -0.015782 -0.037600        -0.031217 -0.022794
hindi           -0.098053 -0.233607        -0.000920 -0.009523
kannada         -0.013045 -0.031078         0.059154  0.008145
malayalam       -0.022931 -0.054632         0.023192  0.020708
marathi         -0.023155 -0.055165        -0.015152  0.034446
oriya           -0.008935 -0.021288        -0.071638  0.004390
punjabi          1.000000 -0.023705        -0.059538  0.013611
tamil           -0.023705  1.000000        -0.004603 -0.023092
normal_duration -0.059538 -0.004603         1.000000  0.003295
cluster          0.013611 -0.023092         0.003295  1.000000

[39 rows x 39 columns]
```

[55]:
```
sns.heatmap(df13.corr())
```

[55]:
```
<AxesSubplot:>
```



[56]:
```
corrMatrix=df13.corr().abs()
upperMatrix = corrMatrix.where(np.triu(np.ones(corrMatrix.shape), k=1).astype(np.
 ↪bool))

# Find index of feature columns with correlation greater than 0.95
corrFutures = [column for column in upperMatrix.columns if␣
 ↪any(upperMatrix[column] > 0.95)]

df13.drop(columns=corrFutures)
```

```
C:\Users\dell\AppData\Local\Temp\ipykernel_9756\3918593334.py:2:
DeprecationWarning: `np.bool` is a deprecated alias for the builtin `bool`. To
silence this warning, use `bool` by itself. Doing this will not modify any
behavior and is safe. If you specifically wanted the numpy scalar type, use
`np.bool_` here.
Deprecated in NumPy 1.20; for more details and guidance:
https://numpy.org/devdocs/release/1.20.0-notes.html#deprecations
  upperMatrix = corrMatrix.where(np.triu(np.ones(corrMatrix.shape),
k=1).astype(np.bool))
```

[56]:

	rating	episode_count	season_count	movies	series	action	adventure	\
0	10	32	19	0	1	0	0	
1	4	21	15	0	1	0	0	
2	8	28	13	0	1	0	0	
4	2	29	10	0	1	1	0	
5	10	37	1	0	1	0	0	
...	...	...	...	...	...	...	...	
48640	6	15	6	0	1	0	0	
48641	4	14	1	0	1	0	0	
48642	6	5	33	0	1	0	0	
48643	5	1	9	0	1	0	0	
48644	8	4	2	0	1	0	0	

	animation	badminton	basketball	...	gujarati	hindi	kannada	\
0	0	0	0	...	0	0	0	
1	0	0	0	...	0	0	0	
2	0	0	0	...	0	0	0	
4	0	0	0	...	0	1	0	
5	0	0	0	...	0	1	0	
...	...	...	...	...	...	...	...	
48640	0	0	0	...	0	1	0	
48641	0	0	0	...	0	0	0	
48642	0	0	0	...	0	1	0	
48643	0	0	0	...	0	0	0	
48644	0	0	0	...	0	0	0	

	malayalam	marathi	oriya	punjabi	tamil	normal_duration	cluster
0	0	0	0	0	0	15.420940	2
1	0	0	0	0	0	14.914123	0
2	0	0	0	0	1	14.953344	0
4	0	0	0	0	0	15.112974	0
5	0	0	0	0	0	14.933925	2
...	...	...	...	...	...	...	...
48640	0	0	0	0	0	15.027452	0
48641	0	1	0	0	0	14.953344	1
48642	0	0	0	0	0	14.972392	3
48643	0	0	0	0	0	15.206792	1
48644	0	0	0	0	0	14.830741	1

[48113 rows x 38 columns]

7 Dropping highly correlated columns

[57]:
```
input = df13.drop("cluster", axis=1)
output = df13["cluster"]
```

8 Splitting column

[58]:
```
from sklearn.model_selection import train_test_split
```

[59]:
```
X_train, X_test, y_train, y_test = train_test_split(
    input, output, test_size=0.20, random_state=42)
```

9 Model building

[60]:
```
from sklearn.naive_bayes import CategoricalNB
from sklearn.model_selection import cross_val_score
```

[61]:
```
model = CategoricalNB()
```

[62]:
```
model.fit(X_train, y_train)
```

[62]:
```
CategoricalNB()
```

[63]:
```
model.predict(X_test)
```

[63]:
```
array([1, 1, 1, ..., 3, 1, 1])
```

[64]:
```
print(f"train Score:{model.score(X_train,y_train)}")
print(f"test Score:{model.score(X_test,y_test)}")
```

```
train Score:0.9797090153286568
test Score:0.9799438844435208
```