Movie Prediction

August 19, 2020

0.0.1 Project

Your boss has just acquired data about how much audiences and critics like movies as well as numerous other variables about the movies. This dataset is provided below, and it includes information from Rotten Tomatoes and IMDB for a random sample of movies.

She is interested in learning what attributes make a movie popular. She is also interested in learning something new about movies. She wants you team to figure it all out.

As part of this project you will complete exploratory data analysis (EDA), modeling, and prediction.

The data set is comprised of 651 randomly sampled movies produced and released before 2016.

Some of these variables are only there for informational purposes and do not make any sense to include in a statistical analysis. It is up to you to decide which variables are meaningful and which should be omitted. For example information in the the `actor1` through `actor5` variables was used to determine whether the movie casts an actor or actress who won a best actor or actress Oscar.

You might also choose to omit certain observations or restructure some of the variables to make them suitable for answering your research questions.

When you are fitting a model you should also be careful about collinearity, as some of these variables may be dependent on each other.

0.0.2 Import Libraries

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

import xgboost as xgb
from xgboost import XGBClassifier, XGBRegressor
from xgboost import to_graphviz, plot_importance

#from sklearn.experimental import enable_hist_gradient_boosting
#from sklearn.ensemble import _hist_gradient_boosting
#from sklearn.ensemble import HistGradientBoostingRegressor,␣
↪HistGradientBoostingRegressor
```

```
%matplotlib inline
sns.set_style('dark')
sns.set(font_scale=1.2)

from sklearn.model_selection import cross_val_score, train_test_split,␣
 ↪GridSearchCV, RandomizedSearchCV
from sklearn.preprocessing import LabelEncoder, StandardScaler, MinMaxScaler,␣
 ↪OneHotEncoder
from sklearn.metrics import confusion_matrix, classification_report,␣
 ↪mean_absolute_error, mean_squared_error,r2_score
from sklearn.metrics import plot_confusion_matrix, plot_precision_recall_curve,␣
 ↪plot_roc_curve, accuracy_score
from sklearn.metrics import auc, f1_score, precision_score, recall_score,␣
 ↪roc_auc_score


import feature_engine.missing_data_imputers as mdi
from feature_engine.outlier_removers import Winsorizer

#from tpot import TPOTClassifier, TPOTRegressor
#from imblearn.under_sampling import RandomUnderSampler
#from imblearn.over_sampling import RandomOverSampler

import warnings
warnings.filterwarnings('ignore')

import pickle
from pickle import dump, load

np.random.seed(0)

#from pycaret.classification import *
#from pycaret.clustering import *
from pycaret.regression import *

pd.set_option('display.max_columns',100)
#pd.set_option('display.max_rows',100)
pd.set_option('display.width', 1000)
```

0.0.3 Data Exploration and Analysis

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

[3]:
```
df
```

[3]:
```
                          title    title_type               genre  runtime
mpaa_rating                    studio  thtr_rel_year  thtr_rel_month
thtr_rel_day  dvd_rel_year  dvd_rel_month  dvd_rel_day  imdb_rating
imdb_num_votes   critics_rating  critics_score audience_rating  audience_score
best_pic_nom best_pic_win best_actor_win best_actress_win best_dir_win
top200_box           director             actor1               actor2
actor3              actor4               actor5
imdb_url                                          rt_url
0                    Filly Brown  Feature Film               Drama     80.0
R       Indomina Media Inc.           2013               4            19
2013.0            7.0         30.0          5.5             899           Rotten
45         Upright              73           no           no             no
no           no         no   Michael D. Olmos     Gina Rodriguez          Jenni
Rivera  Lou Diamond Phillips       Emilio Rivera  Joseph Julian Soria
http://www.imdb.com/title/tt1869425/
//www.rottentomatoes.com/m/filly_brown_2012/
1                       The Dish  Feature Film               Drama    101.0
PG-13     Warner Bros. Pictures           2001               3            14
2001.0            8.0         28.0          7.3           12285  Certified Fresh
96         Upright              81           no           no             no
no           no         no          Rob Sitch          Sam Neill      Kevin
Harrington     Patrick Warburton            Tom Long       Genevieve Mooy
http://www.imdb.com/title/tt0205873/
//www.rottentomatoes.com/m/dish/
2            Waiting for Guffman  Feature Film              Comedy     84.0
R    Sony Pictures Classics           1996               8            21
2001.0            8.0         21.0          7.6           22381  Certified Fresh
91         Upright              91           no           no             no
no           no         no  Christopher Guest  Christopher Guest      Catherine
O'Hara          Parker Posey         Eugene Levy          Bob Balaban
http://www.imdb.com/title/tt0118111/
//www.rottentomatoes.com/m/waiting_for_guffman/
3           The Age of Innocence  Feature Film               Drama    139.0
PG         Columbia Pictures           1993              10             1
2001.0           11.0          6.0          7.2           35096  Certified Fresh
80         Upright              76           no           no            yes
no          yes         no    Martin Scorsese   Daniel Day-Lewis     Michelle
Pfeiffer          Winona Ryder    Richard E. Grant         Alec McCowen
http://www.imdb.com/title/tt0106226/
//www.rottentomatoes.com/m/age_of_innocence/
4                    Malevolence  Feature Film              Horror     90.0
R  Anchor Bay Entertainment           2004               9            10
2005.0            4.0         19.0          5.1             2386           Rotten
33         Spilled              27           no           no             no
no           no         no        Stevan Mena      Samantha Dark    R. Brandon
Johnson       Brandon Johnson       Heather Magee       Richard Glover
http://www.imdb.com/title/tt0388230/
```

```
//www.rottentomatoes.com/m/10004684-malevolence/
..                          …           …                 …      …
…                       …            …             …           …
…            …          …          …             …              …
…             …             …          …          …            …
…          …        …                …                 …
…                   …                 …                  …
…                                                 …
646           Death Defying Acts  Feature Film               Drama     97.0
PG         Genius Productions          2008                7            11
2008.0           10.0         28.0          5.9            8345           Rotten
44         Spilled              26           no            no              no
no           no          no  Gillian Armstrong        Guy Pearce  Catherine
Zeta-Jones          Timothy Spall       Saoirse Ronan          Jack Bailey
http://www.imdb.com/title/tt0472071/
//www.rottentomatoes.com/m/death_defying_acts/
647                   Half Baked  Feature Film              Comedy     82.0
R        Universal Pictures           1998                1            16
1998.0            6.0         30.0          6.7           46794           Rotten
29         Upright              81           no            no              no
no           no          no        Tamra Davis     Dave Chappelle
Guillermo Diaz             Jim Breuer    Harland Williams          Rachel True
http://www.imdb.com/title/tt0120693/
//www.rottentomatoes.com/m/half_baked/
648            Dance of the Dead  Feature Film  Action & Adventure     87.0
R  Grindhouse Entertainment           2008                3             9
2008.0           10.0         14.0          5.9           10087            Fresh
80         Spilled              52           no            no              no
no           no          no       Gregg Bishop      Jared Kusnitz      Greyson
Chadwick        Chandler Darby  Carissa Capobianco       Randy McDowell
http://www.imdb.com/title/tt0926063/
//www.rottentomatoes.com/m/1203339-dance_of_th…
649  Around the World in 80 Days  Feature Film  Action & Adventure    120.0
PG      Buena Vista Pictures          2004                6            16
2004.0           11.0          2.0          5.8           66054           Rotten
31         Spilled              34           no            no              no
no           no         yes       Frank Coraci        Jackie Chan          Steve
Coogan           Ewen Bremner         Robert Fyfe          Ian McNeice
http://www.imdb.com/title/tt0327437/
//www.rottentomatoes.com/m/around_the_world_in…
650                          LOL  Feature Film              Comedy     97.0
PG-13           Lionsgate Films          2012                 5             4
2012.0            7.0         31.0          4.2           43574           Rotten
17         Spilled              51           no            no              no
no           no          no       Liza Azuelos        Miley Cyrus           Demi
Moore          Ashley Greene       Douglas Booth       Adam G. Sevani
http://www.imdb.com/title/tt1592873/
```

```
//www.rottentomatoes.com/m/lol_2011/

[651 rows x 32 columns]
```

[4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 651 entries, 0 to 650
Data columns (total 32 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   title             651 non-null    object
 1   title_type        651 non-null    object
 2   genre             651 non-null    object
 3   runtime           650 non-null    float64
 4   mpaa_rating       651 non-null    object
 5   studio            643 non-null    object
 6   thtr_rel_year     651 non-null    int64
 7   thtr_rel_month    651 non-null    int64
 8   thtr_rel_day      651 non-null    int64
 9   dvd_rel_year      643 non-null    float64
 10  dvd_rel_month     643 non-null    float64
 11  dvd_rel_day       643 non-null    float64
 12  imdb_rating       651 non-null    float64
 13  imdb_num_votes    651 non-null    int64
 14  critics_rating    651 non-null    object
 15  critics_score     651 non-null    int64
 16  audience_rating   651 non-null    object
 17  audience_score    651 non-null    int64
 18  best_pic_nom      651 non-null    object
 19  best_pic_win      651 non-null    object
 20  best_actor_win    651 non-null    object
 21  best_actress_win  651 non-null    object
 22  best_dir_win      651 non-null    object
 23  top200_box        651 non-null    object
 24  director          649 non-null    object
 25  actor1            649 non-null    object
 26  actor2            644 non-null    object
 27  actor3            642 non-null    object
 28  actor4            638 non-null    object
 29  actor5            636 non-null    object
 30  imdb_url          651 non-null    object
 31  rt_url            651 non-null    object
dtypes: float64(5), int64(6), object(21)
memory usage: 162.9+ KB
```

[5]:
```
df.describe(include='all')
```

[5]:
```
                    title    title_type  genre     runtime mpaa_rating
studio  thtr_rel_year  thtr_rel_month  thtr_rel_day  dvd_rel_year  dvd_rel_month
dvd_rel_day  imdb_rating  imdb_num_votes critics_rating  critics_score
audience_rating  audience_score best_pic_nom best_pic_win best_actor_win
best_actress_win best_dir_win top200_box         director         actor1
actor2      actor3        actor4       actor5
imdb_url                                    rt_url
count                  651           651    651  650.000000         651
643     651.000000      651.000000    651.000000    643.000000     643.000000
643.000000   651.000000      651.000000            651     651.000000
651      651.000000          651          651            651              651
651        651              649            649           644         642
638          636                                  651
651
unique                 647             3     11         NaN           6
211            NaN             NaN           NaN           NaN            NaN
NaN          NaN             NaN              3            NaN              2
NaN            2            2              2               2            2
2              532            485           572         601           607
615                                  650
650
top     Where the Heart Is  Feature Film  Drama         NaN           R
Paramount Pictures            NaN             NaN           NaN           NaN
NaN          NaN          NaN             NaN         Rotten            NaN
Upright             NaN           no           no             no
no           no         no  Martin Scorsese  John Travolta  Diane Keaton  Gary
Busey  John P. Ryan  Joan Cusack  http://www.imdb.com/title/tt1155592/
//www.rottentomatoes.com/m/man_on_wire/
freq                     2          591    305         NaN         329
37            NaN             NaN           NaN           NaN            NaN
NaN          NaN             NaN           307            NaN             376
NaN          629          644            558             579          608
636                4             7             5           3             3
2                                    2                                      2
mean                   NaN           NaN    NaN  105.821538         NaN
NaN    1997.941628        6.740399     14.416283   2004.427683       6.332815
15.007776     6.493088    57532.983103           NaN      57.688172
NaN       62.362519          NaN          NaN           NaN              NaN
NaN        NaN             NaN            NaN           NaN         NaN
NaN          NaN                                  NaN
NaN
std                    NaN           NaN    NaN   19.445047         NaN
NaN      10.974501        3.554223      8.861167      4.643588       3.378228
8.867400     1.084747   112124.386910           NaN      28.402971
NaN       20.222624          NaN          NaN           NaN              NaN
NaN        NaN             NaN            NaN           NaN         NaN
NaN          NaN                                  NaN
```

```
NaN
min                    NaN           NaN    NaN   39.000000          NaN
NaN    1970.000000        1.000000      1.000000   1991.000000        1.000000
1.000000     1.900000      180.000000            NaN       1.000000
NaN       11.000000          NaN           NaN            NaN              NaN
NaN        NaN              NaN            NaN           NaN          NaN
NaN          NaN                                  NaN
NaN
25%                    NaN           NaN    NaN   92.000000          NaN
NaN    1990.000000        4.000000      7.000000   2001.000000        3.000000
7.000000     5.900000     4545.500000            NaN      33.000000
NaN       46.000000          NaN           NaN            NaN              NaN
NaN        NaN              NaN            NaN           NaN          NaN
NaN          NaN                                  NaN
NaN
50%                    NaN           NaN    NaN  103.000000          NaN
NaN    2000.000000        7.000000     15.000000   2004.000000        6.000000
15.000000     6.600000    15116.000000            NaN      61.000000
NaN       65.000000          NaN           NaN            NaN              NaN
NaN        NaN              NaN            NaN           NaN          NaN
NaN          NaN                                  NaN
NaN
75%                    NaN           NaN    NaN  115.750000          NaN
NaN    2007.000000       10.000000     21.000000   2008.000000        9.000000
23.000000     7.300000    58300.500000            NaN      83.000000
NaN       80.000000          NaN           NaN            NaN              NaN
NaN        NaN              NaN            NaN           NaN          NaN
NaN          NaN                                  NaN
NaN
max                    NaN           NaN    NaN  267.000000          NaN
NaN    2014.000000       12.000000     31.000000   2015.000000       12.000000
31.000000     9.000000   893008.000000            NaN     100.000000
NaN       97.000000          NaN           NaN            NaN              NaN
NaN        NaN              NaN            NaN           NaN          NaN
NaN          NaN                                  NaN
NaN
```

[6]:
```
df.shape
```

[6]:
```
(651, 32)
```

[7]:
```
df.columns
```

[7]:
```
Index(['title', 'title_type', 'genre', 'runtime', 'mpaa_rating', 'studio',
'thtr_rel_year', 'thtr_rel_month', 'thtr_rel_day', 'dvd_rel_year',
'dvd_rel_month', 'dvd_rel_day', 'imdb_rating', 'imdb_num_votes',
'critics_rating', 'critics_score', 'audience_rating', 'audience_score',
```

```
'best_pic_nom', 'best_pic_win', 'best_actor_win', 'best_actress_win',
'best_dir_win', 'top200_box', 'director', 'actor1', 'actor2', 'actor3',
'actor4', 'actor5', 'imdb_url', 'rt_url'], dtype='object')
```

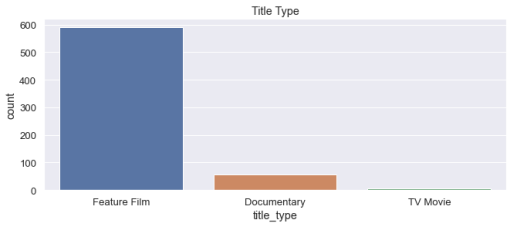
0.0.4 Data Visualization

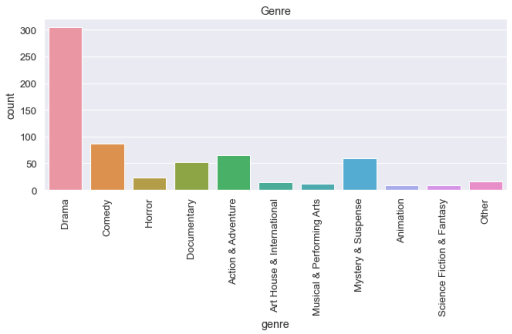
0.0.5 Univariate Data Exploration

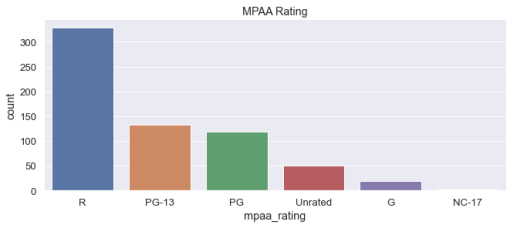
[8]:
```
df.hist(bins=50, figsize=(20,15))
plt.tight_layout()
plt.show()
```

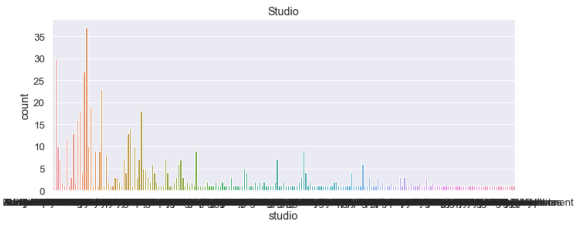
[9]:
```
fig = plt.figure(figsize=(20,40))

plt.subplot(7,2,1)
plt.title("Title Type")
sns.countplot(df.title_type)

plt.subplot(7,2,2)
plt.title("Genre")
plt.xticks(rotation=90)
```

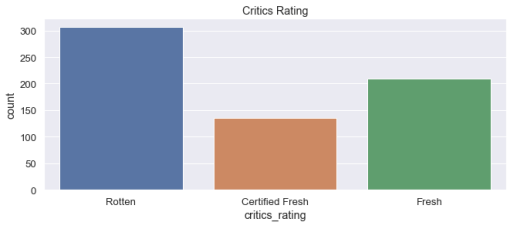
```
sns.countplot(df.genre)

plt.subplot(7,2,3)
plt.title("MPAA Rating")
sns.countplot(df.mpaa_rating)

plt.subplot(7,2,4)
plt.title("Studio")
sns.countplot(df.studio)

plt.subplot(7,2,5)
plt.title("Critics Rating")
sns.countplot(df.critics_rating)

plt.subplot(7,2,6)
plt.title("Audience Rating")
sns.countplot(df.audience_rating)

plt.subplot(7,2,7)
plt.title("Best Pic Nominee")
sns.countplot(df.best_pic_nom)

plt.subplot(7,2,8)
plt.title("Best Picture Winner")
sns.countplot(df.best_pic_win)

plt.subplot(7,2,9)
plt.title("Best Actor Winner")
sns.countplot(df.best_actor_win)

plt.subplot(7,2,10)
plt.title("Best Actress Winner")
sns.countplot(df.best_actress_win)

plt.subplot(7,2,11)
plt.title("Best Director Winner")
sns.countplot(df.best_dir_win)

plt.subplot(7,2,12)
plt.title("Box Office Top 200")
sns.countplot(df.top200_box)

plt.tight_layout()
plt.show()
```

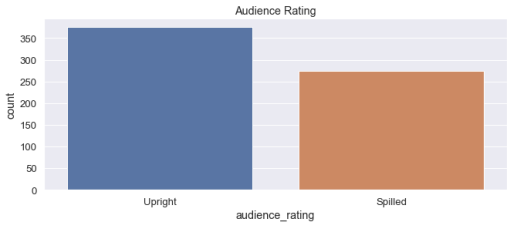

Title Type
count
title_type
Feature Film
Documentary
TV Movie

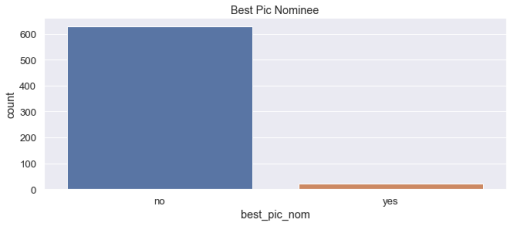

Genre
count
genre
Drama
Comedy
Horror
Documentary
Action & Adventure
Art House & International
Musical & Performing Arts
Mystery & Suspense
Animation
Science Fiction & Fantasy
Other

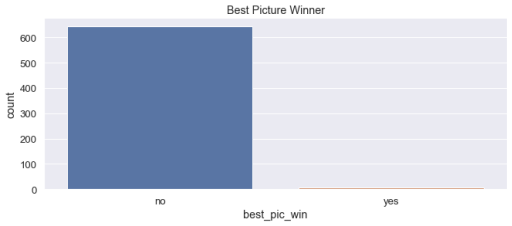

MPAA Rating
count
mpaa_rating
R
PG-13
PG
Unrated
G
NC-17

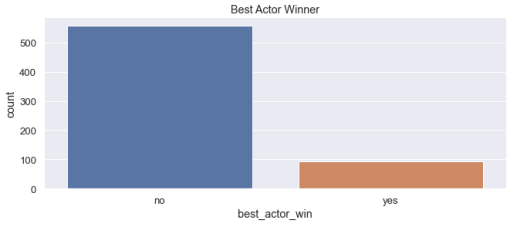

Studio
count
studio

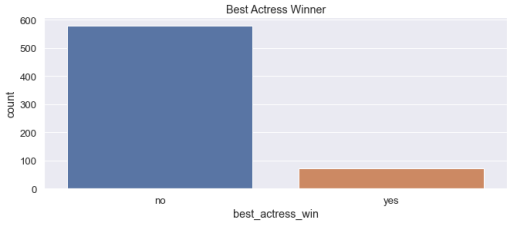

Critics Rating
count
critics_rating
Rotten
Certified Fresh
Fresh

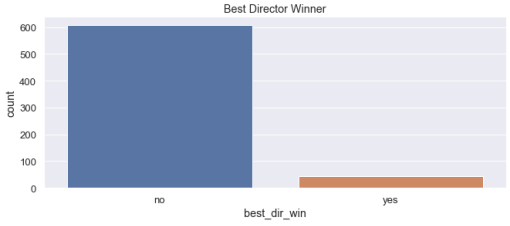

Audience Rating
count
audience_rating
Upright
Spilled

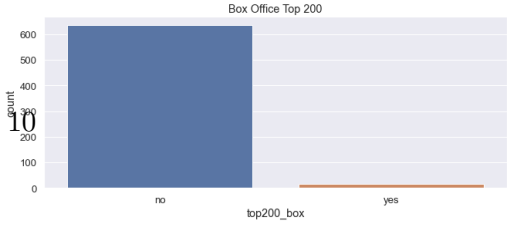

Best Pic Nominee
count
best_pic_nom
no
yes


Best Picture Winner
count
best_pic_win
no
yes


Best Actor Winner
count
best_actor_win
no
yes


Best Actress Winner
count
best_actress_win
no
yes


Best Director Winner
count
best_dir_win
no
yes


Box Office Top 200
count
top200_box
no
yes

0.0.6 Drop unwanted features

[10]:
```
df.columns
```

[10]:
```
Index(['title', 'title_type', 'genre', 'runtime', 'mpaa_rating', 'studio',
'thtr_rel_year', 'thtr_rel_month', 'thtr_rel_day', 'dvd_rel_year',
'dvd_rel_month', 'dvd_rel_day', 'imdb_rating', 'imdb_num_votes',
'critics_rating', 'critics_score', 'audience_rating', 'audience_score',
'best_pic_nom', 'best_pic_win', 'best_actor_win', 'best_actress_win',
'best_dir_win', 'top200_box', 'director', 'actor1', 'actor2', 'actor3',
'actor4', 'actor5', 'imdb_url', 'rt_url'], dtype='object')
```

[11]:
```
df.drop(['title', 'title_type', 'genre','mpaa_rating', 'studio',␣
↪'thtr_rel_year', 'thtr_rel_month', 'thtr_rel_day', 'dvd_rel_year',
       'dvd_rel_month',␣
↪'dvd_rel_day','critics_rating','audience_rating','best_pic_nom',␣
↪'best_pic_win', 'best_actor_win', 'best_actress_win',
       'best_dir_win', 'top200_box', 'director', 'actor1', 'actor2', 'actor3',␣
↪'actor4', 'actor5', 'imdb_url', 'rt_url'],axis=1,inplace=True)
```

[12]:
```
df
```

[12]:

	runtime	imdb_rating	imdb_num_votes	critics_score	audience_score
0	80.0	5.5	899	45	73
1	101.0	7.3	12285	96	81
2	84.0	7.6	22381	91	91
3	139.0	7.2	35096	80	76
4	90.0	5.1	2386	33	27
..	…	…	…	…	…
646	97.0	5.9	8345	44	26
647	82.0	6.7	46794	29	81
648	87.0	5.9	10087	80	52
649	120.0	5.8	66054	31	34
650	97.0	4.2	43574	17	51

[651 rows x 5 columns]

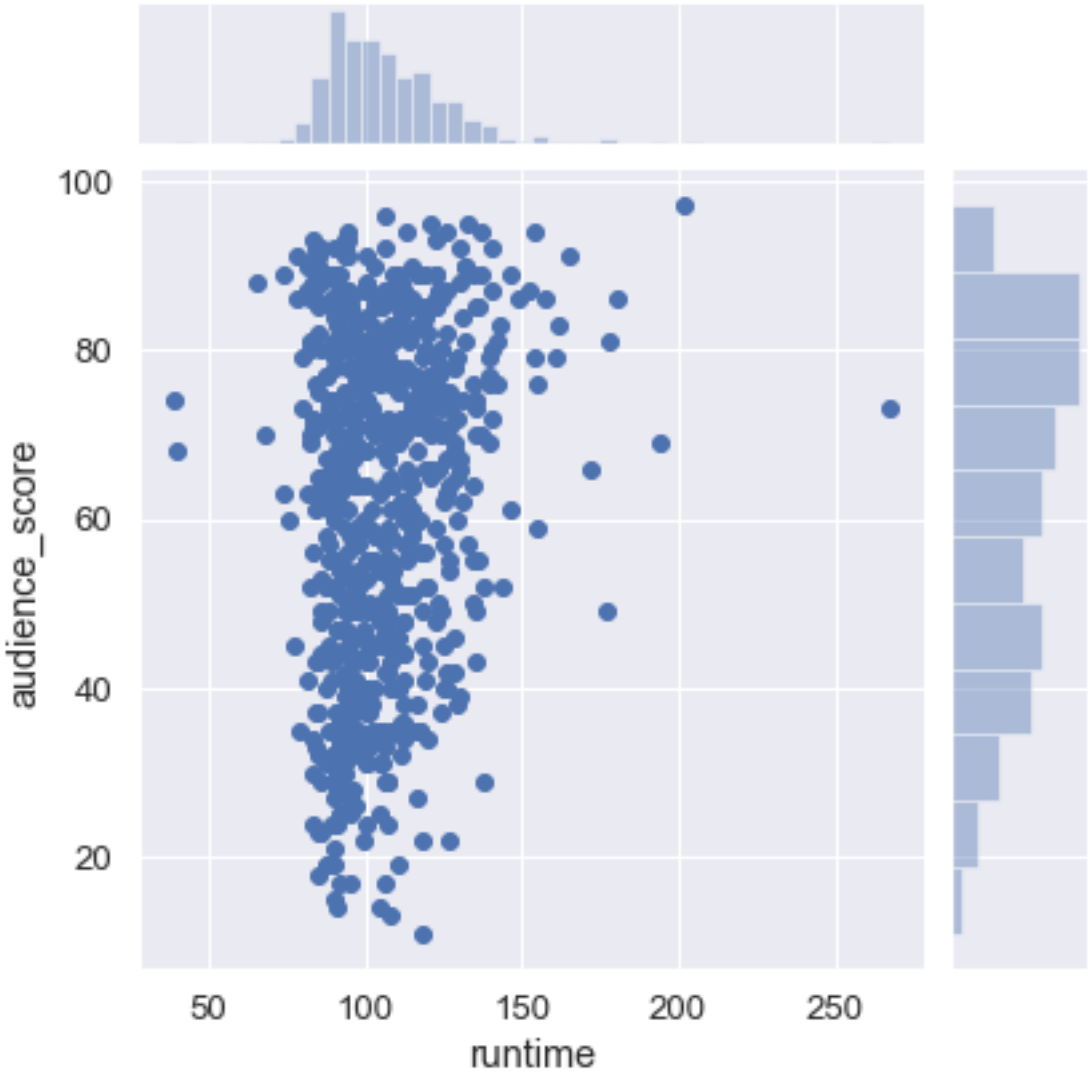
[13]:
```
df.info()
```

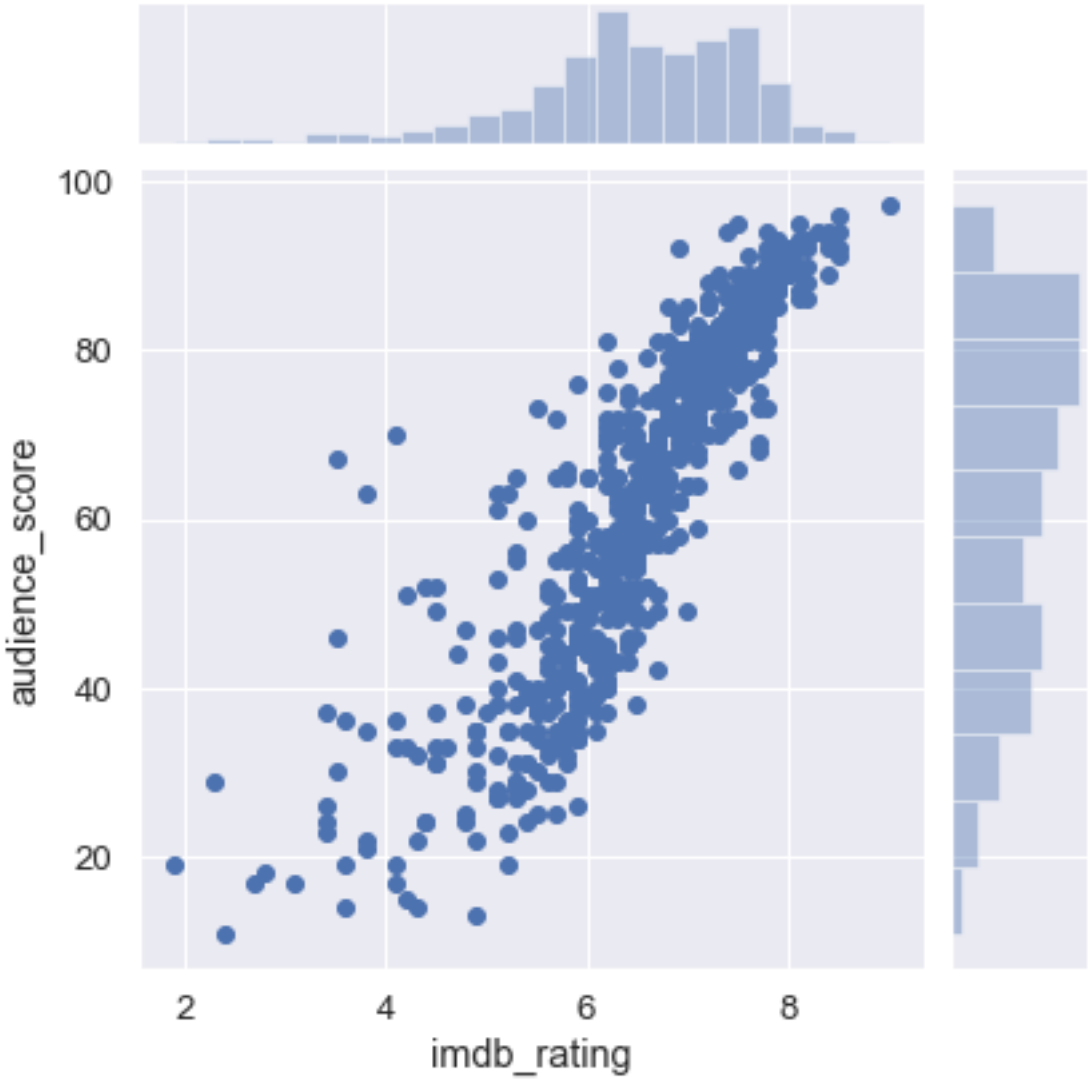
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 651 entries, 0 to 650
Data columns (total 5 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   runtime         650 non-null    float64
```

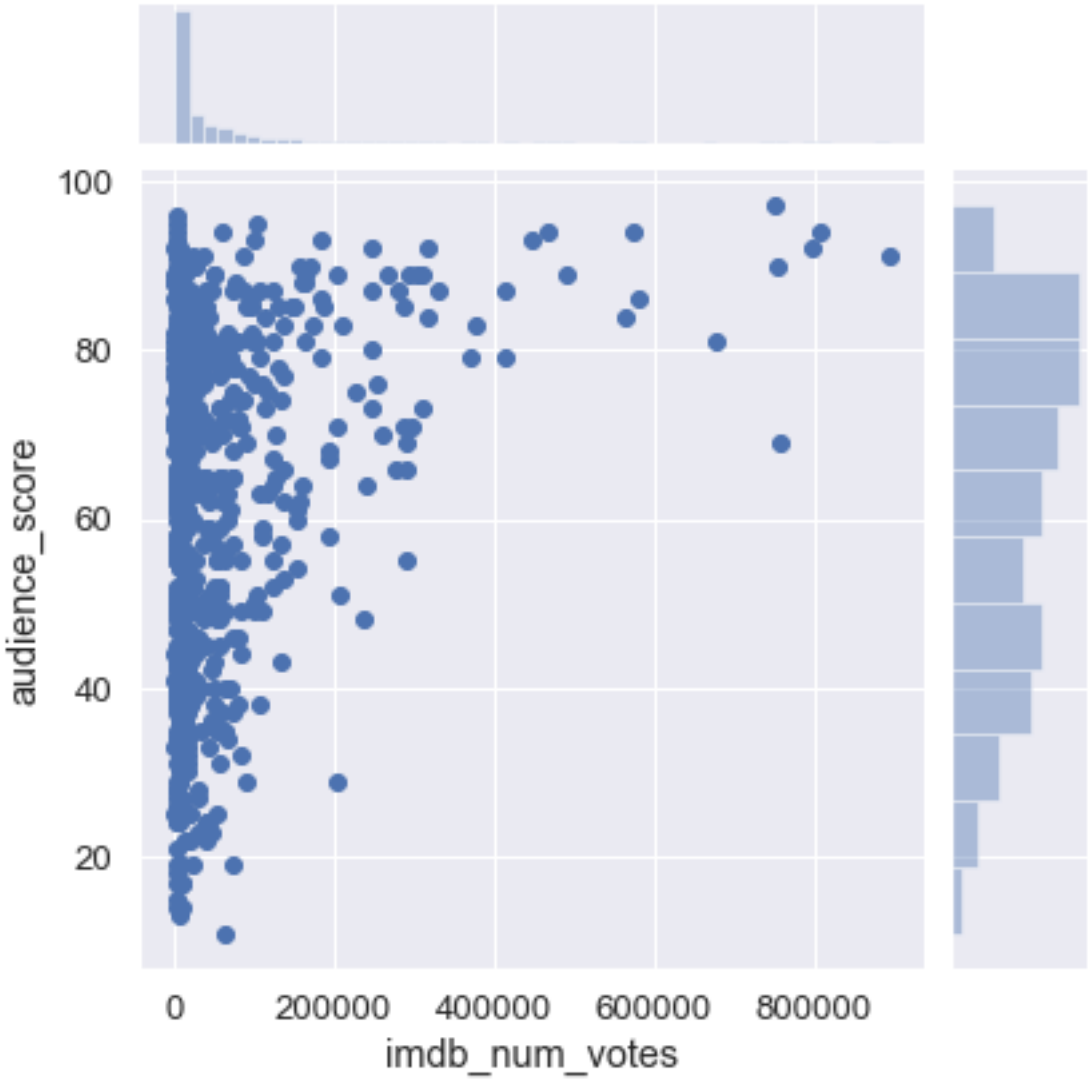
```
 1   imdb_rating     651 non-null    float64
 2   imdb_num_votes  651 non-null    int64
 3   critics_score   651 non-null    int64
 4   audience_score  651 non-null    int64
dtypes: float64(2), int64(3)
memory usage: 25.6 KB
```

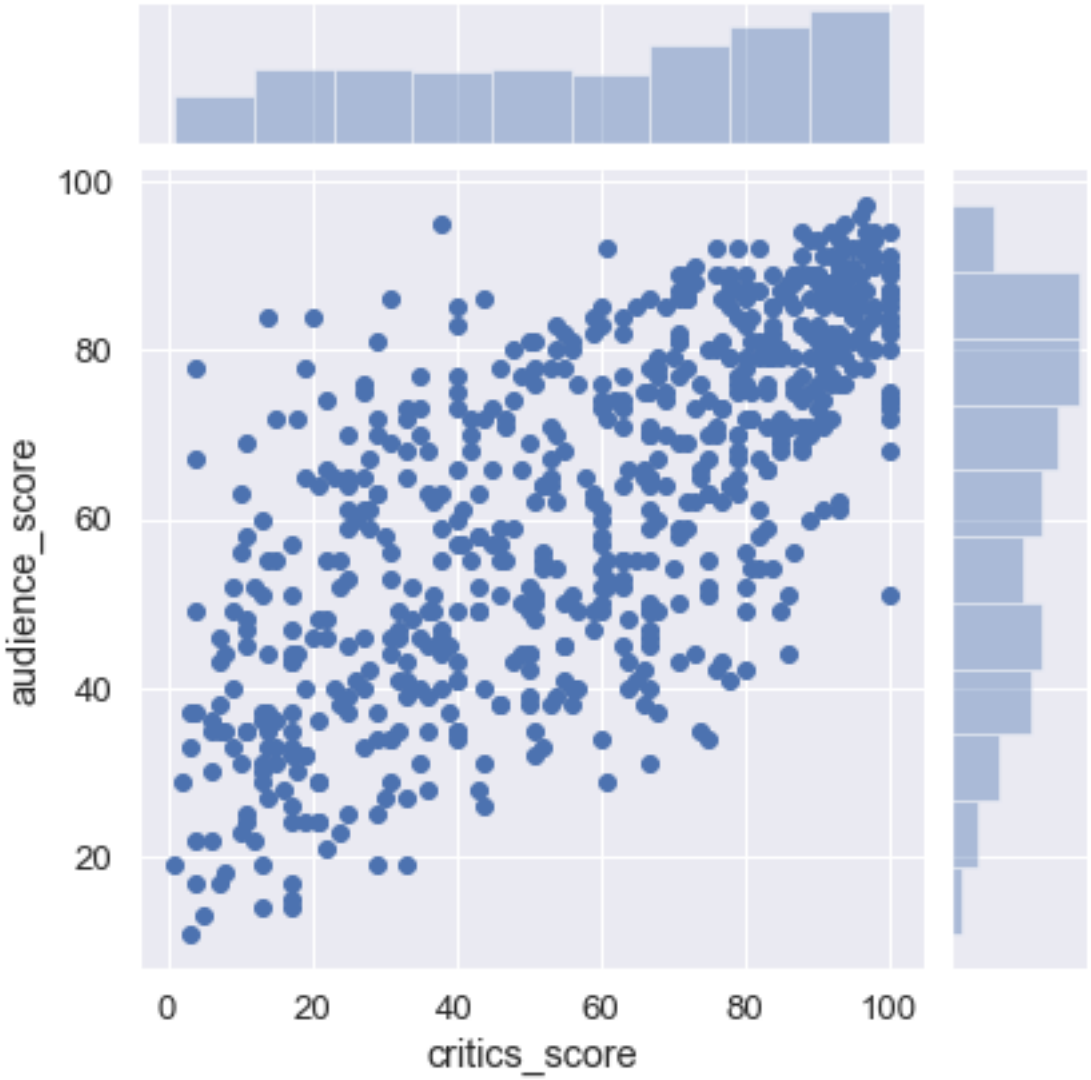
0.0.7 Bivariate Data Exploration

[14]:
```
sns.jointplot(x='runtime', y='audience_score',data=df, kind='scatter')

sns.jointplot(x='imdb_rating', y='audience_score',data=df, kind='scatter')

sns.jointplot(x='imdb_num_votes', y='audience_score',data=df, kind='scatter')

sns.jointplot(x='critics_score', y='audience_score',data=df, kind='scatter')

plt.show()
```


audience_score
100
80
60
40
20
50
100
150
200
250
runtime


100
80
60
40
20
audience_score
2
4
6
8
imdb_rating


100
80
60
40
20
audience_score
0
200000
400000
600000
800000
imdb_num_votes



0.0.8 Correlation

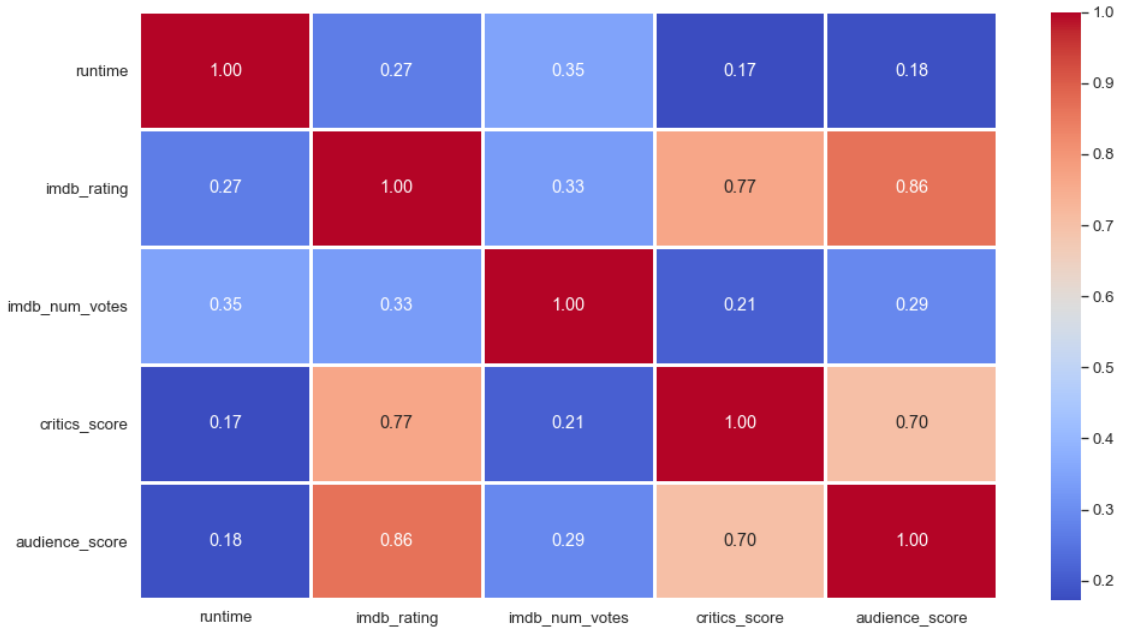
[15]:
```
df.corr()
```

[15]:

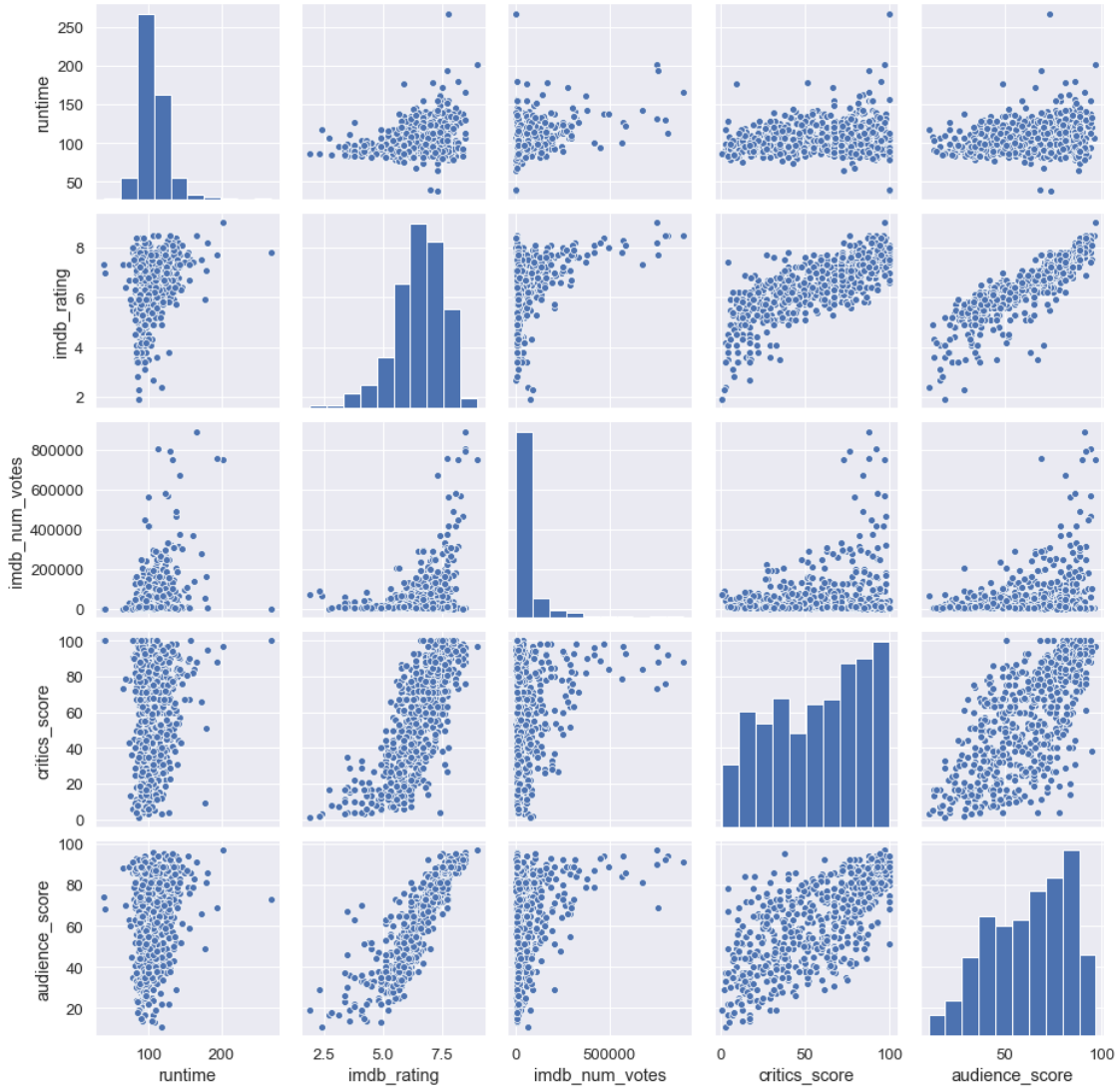
	runtime	imdb_rating	imdb_num_votes	critics_score	audience_score
runtime	1.000000	0.268240	0.347215	0.172499	0.180963
imdb_rating	0.268240	1.000000	0.331152	0.765036	0.864865
imdb_num_votes	0.347215	0.331152	1.000000	0.209251	0.289813
critics_score	0.172499	0.765036	0.209251	1.000000	

```
0.704276
audience_score   0.180963      0.864865         0.289813         0.704276
1.000000
```

[16]:
```
plt.figure(figsize=(16,9))
sns.heatmap(df.corr(),cmap="coolwarm",annot=True,fmt='.2f',linewidths=2)
plt.show()
```



[17]:
```
sns.pairplot(df)
plt.show()
```



[ ]:

0.0.9 Data Preprocessing

0.0.10 Treat Missing Values

[18]:
```
df.isnull().sum()
```

[18]:
```
runtime           1
imdb_rating       0
imdb_num_votes    0
critics_score     0
audience_score    0
dtype: int64
```

[19]:
```
imputer = mdi.MeanMedianImputer(imputation_method='median',variables=None)
```

[20]:
```
imputer.fit(df)
```

[20]:
```
MeanMedianImputer(variables=['runtime', 'imdb_rating', 'imdb_num_votes',
                             'critics_score', 'audience_score'])
```

[21]:
```
df = imputer.transform(df)
```

[22]:
```
df.isnull().sum()
```

[22]:
```
runtime           0
imdb_rating       0
imdb_num_votes    0
critics_score     0
audience_score    0
dtype: int64
```

0.0.11 Treat Duplicate Values

[23]:
```
df.duplicated(keep='first').sum()
```

[23]:
```
1
```

0.0.12 Create and save processed dataset

[24]:
```
#df.to_csv("moviestrain.csv",index=False)
```

[25]:
```
df.shape
```

[25]:
```
(651, 5)
```

0.1 Predict Audience Score

0.1.1 Train Test Split

[26]:
```
X = df.iloc[:,0:4]
y = df.iloc[:,4]
```

[27]:
```
X.values, y.values
```

[27]:
```
(array([[8.0000e+01, 5.5000e+00, 8.9900e+02, 4.5000e+01],
        [1.0100e+02, 7.3000e+00, 1.2285e+04, 9.6000e+01],
        [8.4000e+01, 7.6000e+00, 2.2381e+04, 9.1000e+01],
        …,
        [8.7000e+01, 5.9000e+00, 1.0087e+04, 8.0000e+01],
        [1.2000e+02, 5.8000e+00, 6.6054e+04, 3.1000e+01],
```

```
        [9.7000e+01, 4.2000e+00, 4.3574e+04, 1.7000e+01]]),
 array([73, 81, 91, 76, 27, 86, 76, 47, 89, 66, 75, 46, 89, 53, 36, 64, 80,
        92, 24, 19, 73, 86, 42, 71, 77, 41, 81, 43, 61, 91, 71, 77, 85, 70,
        57, 55, 70, 51, 94, 81, 64, 88, 64, 40, 94, 62, 64, 49, 17, 40, 88,
        75, 56, 52, 83, 72, 59, 38, 35, 43, 54, 44, 96, 89, 58, 52, 90, 55,
        52, 85, 43, 30, 38, 75, 55, 39, 59, 47, 24, 87, 83, 81, 35, 34, 92,
        55, 54, 14, 76, 40, 82, 87, 41, 75, 43, 52, 25, 74, 93, 90, 80, 34,
        37, 86, 33, 45, 81, 48, 24, 35, 91, 69, 75, 43, 35, 78, 78, 64, 40,
        88, 43, 65, 46, 55, 75, 67, 54, 41, 75, 76, 74, 87, 37, 86, 79, 42,
        63, 38, 51, 84, 44, 33, 74, 78, 59, 74, 55, 52, 89, 82, 22, 85, 66,
        61, 33, 43, 50, 49, 84, 62, 63, 83, 65, 74, 61, 70, 62, 69, 63, 39,
        32, 79, 87, 70, 71, 86, 33, 70, 87, 68, 83, 85, 39, 44, 17, 58, 11,
        41, 85, 49, 79, 54, 41, 77, 79, 85, 80, 89, 31, 46, 92, 62, 36, 80,
        57, 55, 55, 90, 75, 40, 71, 86, 73, 61, 45, 70, 56, 56, 91, 81, 46,
        65, 21, 49, 48, 90, 31, 84, 77, 48, 45, 58, 73, 27, 85, 86, 83, 25,
        61, 78, 40, 51, 40, 87, 66, 52, 75, 40, 38, 49, 63, 83, 81, 61, 80,
        46, 64, 19, 78, 80, 33, 51, 81, 65, 90, 85, 71, 90, 50, 51, 87, 69,
        87, 84, 86, 71, 82, 72, 59, 76, 45, 63, 29, 72, 40, 76, 84, 71, 42,
        88, 85, 75, 76, 66, 55, 76, 42, 74, 79, 64, 48, 23, 61, 66, 45, 54,
        49, 59, 60, 48, 58, 28, 31, 87, 68, 47, 37, 83, 84, 74, 79, 44, 84,
        27, 51, 64, 41, 35, 78, 72, 66, 37, 82, 72, 63, 89, 56, 62, 46, 38,
        92, 89, 82, 41, 72, 83, 87, 29, 45, 34, 79, 97, 84, 63, 44, 68, 61,
        86, 50, 65, 49, 83, 57, 40, 69, 63, 85, 79, 57, 53, 65, 66, 35, 95,
        33, 19, 17, 89, 86, 55, 72, 84, 82, 94, 55, 88, 68, 89, 76, 43, 62,
        36, 60, 89, 74, 78, 71, 82, 89, 91, 44, 56, 45, 70, 89, 35, 13, 30,
        68, 49, 63, 81, 47, 51, 50, 78, 93, 35, 29, 31, 85, 31, 85, 49, 69,
        80, 79, 43, 37, 86, 55, 52, 75, 49, 74, 58, 65, 67, 60, 83, 57, 37,
        29, 78, 67, 76, 49, 51, 92, 78, 87, 60, 32, 67, 47, 39, 24, 93, 24,
        57, 47, 77, 72, 72, 49, 78, 29, 70, 94, 14, 70, 30, 35, 73, 28, 37,
        92, 76, 81, 58, 72, 66, 23, 70, 81, 60, 52, 49, 46, 79, 86, 63, 88,
        86, 34, 59, 87, 50, 62, 25, 82, 85, 26, 38, 45, 70, 86, 28, 81, 32,
        77, 72, 72, 53, 60, 49, 36, 59, 49, 73, 59, 49, 65, 64, 68, 64, 19,
        93, 76, 77, 31, 72, 78, 44, 52, 44, 73, 85, 46, 77, 80, 71, 91, 42,
        61, 37, 67, 65, 54, 76, 15, 79, 49, 86, 77, 67, 87, 44, 65, 78, 87,
        22, 35, 80, 79, 32, 44, 73, 69, 71, 39, 82, 46, 68, 87, 85, 89, 71,
        62, 68, 80, 34, 81, 80, 89, 72, 86, 29, 59, 81, 71, 94, 54, 79, 82,
        87, 87, 71, 88, 50, 70, 92, 70, 38, 44, 80, 66, 55, 79, 69, 91, 38,
        67, 58, 89, 88, 52, 89, 55, 53, 65, 33, 74, 78, 73, 89, 35, 70, 48,
        18, 41, 22, 81, 35, 73, 95, 37, 40, 72, 55, 65, 81, 81, 73, 74, 35,
        26, 81, 52, 34, 51], dtype=int64))
```

[28]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
↪random_state=0)
```

[29]:
```
X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

[29]:
```
((520, 4), (131, 4), (520,), (131,))
```

0.1.2 Feature Scaling

[30]:
```
X_train
```

[30]:
```
     runtime  imdb_rating  imdb_num_votes  critics_score
432    119.0          6.8           11103             87
399     93.0          8.0           36909             98
346     94.0          7.9           38076            100
604     89.0          6.1           26301             49
603    116.0          5.7           48718             53
..       …            …               …              …
9      119.0          6.6           12496             83
359    108.0          6.8           37506             79
192    112.0          5.9             725             33
629     85.0          2.8            3790              8
559     90.0          7.7           56919             55

[520 rows x 4 columns]
```

[31]:
```
scaler = StandardScaler()
```

[32]:
```
X_train_scaled = scaler.fit_transform(X_train)
```

[33]:
```
X_test_scaled = scaler.transform(X_test)
```

[34]:
```
X_train_scaled
```

[34]:
```
array([[ 0.64158617,  0.26977263, -0.41831233,  0.99130441],
       [-0.64082734,  1.37362916, -0.20085178,  1.37252406],
       [-0.59150375,  1.28164111, -0.19101777,  1.44183673],
       …,
       [ 0.29632099, -0.55811977, -0.50576507, -0.88013755],
       [-1.03541612, -3.40974913, -0.4799371 , -1.74654587],
       [-0.78879813,  1.09766503, -0.03223264, -0.11769824]])
```

[35]:
```
X_test_scaled
```

[35]:
```
array([[-6.40827343e-01, -7.42095855e-01, -3.50409625e-01,
        -1.43463888e+00],
       [-9.36768922e-01, -4.23764153e+00,  1.05123288e-01,
        -1.98914020e+00],
       [-5.91503746e-01, -1.90167591e-01,  2.08558505e-02,
         7.48710078e-01],
       [ 7.40233361e-01,  1.77784585e-01, -1.54802686e-02,
         2.98177754e-01],
       [ 7.40233361e-01,  8.57965411e-02, -4.80442708e-01,
        -3.25636233e-01],
```

```
[-6.40827343e-01, -1.47800021e+00, -3.81335775e-01,
 -1.81585854e+00],
[ 1.18414573e+00, -2.82155635e-01, -4.44325684e-01,
 -4.64261563e-01],
[ 9.90266054e-02, -8.34083899e-01, -4.36008488e-01,
  1.94208756e-01],
[-9.36768922e-01, -4.66131723e-01, -4.91372195e-01,
  9.56648074e-01],
[ 3.79412281e-04, -5.58119767e-01,  1.85329035e-01,
 -5.68230561e-01],
[-4.89441843e-02,  7.29712849e-01, -2.19972043e-03,
  4.36803084e-01],
[ 4.97030088e-02, -9.81795470e-02, -4.83712285e-01,
  1.59552424e-01],
[-7.88798133e-01,  7.29712849e-01, -4.99327041e-01,
  9.21991741e-01],
[-1.47591377e-01,  4.53748717e-01, -3.70684371e-01,
 -2.56323568e-01],
[-8.87445326e-01, -1.90167591e-01, -4.83569030e-01,
  1.19924240e+00],
[ 3.79412281e-04, -1.90167591e-01, -3.00750825e-01,
 -5.33574229e-01],
[-3.44885764e-01, -2.76583282e+00, -2.77872215e-01,
 -8.10824890e-01],
[-9.82677808e-02,  5.45736761e-01,  5.69790791e-01,
  3.32834087e-01],
[-9.86092519e-01, -1.84595238e+00, -4.85785264e-01,
 -1.88517120e+00],
[-3.44885764e-01,  9.13688937e-01, -2.39370423e-01,
  1.26855507e+00],
[ 4.44291781e-01,  8.57965411e-02,  4.17072915e-01,
 -5.33574229e-01],
[ 5.92262571e-01,  8.21700893e-01, -4.39876364e-01,
  1.02596074e+00],
[-8.38121729e-01, -6.50107811e-01, -3.54547157e-01,
  2.28865089e-01],
[ 2.96320992e-01,  2.69772629e-01, -3.59004904e-01,
  5.55834258e-02],
[ 4.93615378e-01,  1.09766503e+00, -4.28415992e-01,
  9.21991741e-01],
[ 8.88204150e-01,  4.53748717e-01, -2.84832706e-01,
 -8.30419047e-02],
[-2.46238571e-01, -3.74143679e-01, -3.86307554e-01,
  2.98177754e-01],
[ 1.48350202e-01,  1.09766503e+00, -3.37120651e-01,
  1.16458607e+00],
[-4.43532957e-01,  3.61760673e-01, -4.05646931e-01,
```

```
  9.21991741e-01],
[-9.36768922e-01, -5.58119767e-01, -4.26873898e-01,
  7.48710078e-01],
[-2.02188805e+00,  7.29712849e-01, -5.10332368e-01,
  5.06115750e-01],
[ 6.90909764e-01,  9.13688937e-01, -3.29662982e-01,
  1.59552424e-01],
[-5.91503746e-01,  9.13688937e-01, -4.87200957e-01,
  2.98177754e-01],
[-7.39474536e-01,  1.18965307e+00, -4.97860788e-01,
 -6.37543226e-01],
[-8.38121729e-01,  6.37724805e-01, -4.94195154e-01,
 -4.98917896e-01],
[ 1.08549854e+00,  5.45736761e-01, -4.68771668e-01,
  1.02596074e+00],
[ 2.96320992e-01, -6.50107811e-01, -4.12118670e-01,
 -1.57326421e+00],
[ 1.18414573e+00, -4.66131723e-01, -4.80316307e-01,
 -1.52354570e-01],
[ 1.23346933e+00, -9.81795470e-02,  7.91523700e-01,
 -2.56323568e-01],
[-4.43532957e-01, -5.58119767e-01, -4.41553286e-01,
 -4.98917896e-01],
[-6.40827343e-01, -9.81795470e-02, -2.80838429e-01,
 -1.17698237e-01],
[-2.95562167e-01, -4.66131723e-01, -4.91304781e-01,
 -4.29605231e-01],
[ 1.97673799e-01,  1.18965307e+00, -4.23208265e-01,
  1.12992974e+00],
[-4.43532957e-01, -8.34083899e-01, -3.69347581e-02,
 -1.19204455e+00],
[-9.86092519e-01, -1.66197630e+00, -3.97902753e-01,
 -1.74654587e+00],
[-9.82677808e-02,  9.13688937e-01,  3.47936846e-02,
  2.09270932e-02],
[-2.95562167e-01,  1.18965307e+00,  2.98227483e+00,
  1.16458607e+00],
[-4.43532957e-01,  4.53748717e-01, -3.34837003e-01,
  1.59552424e-01],
[-1.13406331e+00,  1.00567698e+00,  1.52675403e-01,
 -2.90979900e-01],
[ 4.93615378e-01, -1.20203608e+00, -4.13643911e-01,
 -7.76168557e-01],
[-1.47591377e-01,  1.18965307e+00, -3.24421548e-01,
  4.71459417e-01],
[ 2.96320992e-01, -6.50107811e-01, -4.96571496e-01,
 -7.06855892e-01],
```

```
[ 6.41586168e-01, -6.19150296e-03,  1.26144951e-02,
 -2.21667235e-01],
[ 3.79412281e-04, -3.50173718e+00, -5.05461710e-01,
 -1.43463888e+00],
[-1.03541612e+00, -4.66131723e-01, -4.97717533e-01,
 -8.80137555e-01],
[-5.42180150e-01, -1.20203608e+00, -4.45463295e-01,
 -1.64257687e+00],
[-6.90150940e-01, -4.66131723e-01, -4.87150396e-01,
  9.02397584e-02],
[-7.39474536e-01, -7.42095855e-01, -4.64988060e-01,
 -1.19204455e+00],
[ 2.96320992e-01,  1.18965307e+00,  1.56605100e+00,
  1.19924240e+00],
[ 2.46997395e-01, -2.67384478e+00, -4.65156595e-01,
 -1.50395154e+00],
[ 7.89556957e-01,  1.28164111e+00, -5.00532065e-01,
  1.23389873e+00],
[-7.39474536e-01, -1.93794043e+00, -1.93604778e-01,
 -1.36532621e+00],
[ 6.41586168e-01, -7.42095855e-01, -4.37567436e-01,
 -1.12273188e+00],
[ 1.28279292e+00, -2.82155635e-01, -4.29022718e-01,
  6.44741080e-01],
[ 1.67738170e+00,  2.69772629e-01, -4.23663309e-01,
 -4.98917896e-01],
[ 9.37527747e-01, -2.82155635e-01, -2.26721881e-01,
 -8.10824890e-01],
[-2.46238571e-01, -9.81795470e-02, -5.07124842e-02,
 -3.60292566e-01],
[-1.03541612e+00,  4.53748717e-01, -4.50831131e-01,
  2.98177754e-01],
[ 5.92262571e-01, -5.58119767e-01, -4.94212007e-01,
  2.98177754e-01],
[-3.44885764e-01, -6.19150296e-03,  7.70810764e-01,
 -1.12273188e+00],
[-7.39474536e-01, -9.26071943e-01,  1.08477132e-01,
 -1.53860787e+00],
[-9.36768922e-01, -1.01805999e+00, -4.54479911e-01,
 -8.10824890e-01],
[ 2.46997395e-01, -4.66131723e-01,  3.62728844e-01,
 -2.56323568e-01],
[-9.86092519e-01, -1.20203608e+00, -1.22280819e-01,
 -1.19204455e+00],
[ 3.79412281e-04,  3.61760673e-01, -4.58482614e-01,
  9.02397584e-02],
[-5.42180150e-01, -2.82155635e-01, -3.03919280e-01,
```

```
 -1.17698237e-01],
[ 2.46997395e-01,  3.61760673e-01, -4.99841072e-01,
  5.06115750e-01],
[-8.87445326e-01, -5.58119767e-01, -4.77906258e-01,
 -1.43463888e+00],
[-1.47591377e-01,  7.29712849e-01, -4.39733109e-01,
 -8.30419047e-02],
[-2.46238571e-01, -5.58119767e-01, -4.03363283e-01,
 -1.05341922e+00],
[ 1.03617494e+00,  2.69772629e-01,  9.46741257e-02,
  1.06061707e+00],
[-1.47591377e-01,  8.57965411e-02,  6.84838061e-02,
  7.48710078e-01],
[-5.91503746e-01,  1.77784585e-01, -4.77518628e-01,
 -6.37543226e-01],
[ 1.97673799e-01, -7.42095855e-01, -4.89113827e-01,
 -1.22670088e+00],
[-5.42180150e-01, -6.19150296e-03, -4.68485159e-01,
 -3.94948898e-01],
[ 8.38880554e-01,  5.45736761e-01,  2.74601958e-01,
 -6.37543226e-01],
[-4.89441843e-02,  1.09766503e+00, -4.15160724e-01,
  9.56648074e-01],
[ 4.93615378e-01,  6.37724805e-01,  1.61723504e+00,
  1.19924240e+00],
[ 9.86851344e-01,  1.00567698e+00,  2.26568982e+00,
  4.36803084e-01],
[ 3.79412281e-04,  1.83356938e+00, -4.91970494e-01,
  1.30321140e+00],
[-2.95562167e-01,  5.45736761e-01, -3.60749240e-01,
  5.55834258e-02],
[-5.91503746e-01,  1.28164111e+00, -1.91017768e-01,
  1.44183673e+00],
[-7.88798133e-01,  6.37724805e-01, -1.34853520e-01,
  6.10084748e-01],
[ 3.94968185e-01, -1.90167591e-01, -1.69251488e-01,
 -1.87010903e-01],
[-2.95562167e-01, -1.90167591e-01, -4.85304939e-01,
 -8.80137555e-01],
[-5.42180150e-01, -2.82155635e-01, -3.61895277e-01,
 -9.14793887e-01],
[ 3.45644588e-01, -6.50107811e-01, -4.92535085e-01,
 -1.26135721e+00],
[-2.46238571e-01,  7.29712849e-01, -4.08351915e-01,
  1.30321140e+00],
[ 4.44291781e-01, -3.74143679e-01, -4.93799097e-01,
 -1.57326421e+00],
```

```
[ 5.92262571e-01, -5.58119767e-01, -3.05655189e-01,
 -9.14793887e-01],
[ 7.89556957e-01,  6.37724805e-01,  3.81208693e-01,
  3.32834087e-01],
[ 1.97673799e-01,  1.77784585e-01,  1.93995404e+00,
  4.71459417e-01],
[ 1.43076371e+00,  1.09766503e+00,  2.09615217e+00,
  1.59552424e-01],
[ 3.45644588e-01, -6.19150296e-03, -4.30522678e-01,
 -1.52354570e-01],
[ 7.40233361e-01,  2.69772629e-01,  9.63341942e-02,
  2.63521421e-01],
[-7.39474536e-01, -6.19150296e-03, -4.91582863e-01,
 -7.76168557e-01],
[-8.87445326e-01,  9.13688937e-01, -5.04458928e-01,
  1.09527340e+00],
[ 7.89556957e-01,  1.28164111e+00,  3.27277533e-01,
  1.19924240e+00],
[-6.40827343e-01, -3.74143679e-01, -2.95450403e-01,
 -8.80137555e-01],
[-9.36768922e-01,  4.53748717e-01, -3.88981894e-02,
  3.32834087e-01],
[-1.18338691e+00,  1.00567698e+00, -4.82726356e-01,
  1.16458607e+00],
[-3.44885764e-01, -5.58119767e-01, -4.99293334e-01,
  2.09270932e-02],
[ 9.90266054e-02,  8.21700893e-01, -1.74796286e-01,
  1.16458607e+00],
[-6.90150940e-01, -5.58119767e-01, -1.39007905e-01,
 -1.81585854e+00],
[ 2.71317722e+00,  8.21700893e-01,  2.59590020e+00,
  8.18022743e-01],
[-1.03541612e+00, -1.29402412e+00, -4.31272658e-01,
 -1.36532621e+00],
[-7.88798133e-01, -7.42095855e-01, -4.83451056e-01,
 -1.15738822e+00],
[-1.96914974e-01, -5.58119767e-01, -4.90226158e-01,
 -6.37543226e-01],
[ 4.44291781e-01,  9.13688937e-01, -4.27691292e-01,
  4.36803084e-01],
[ 2.12129406e+00,  8.21700893e-01,  2.88522939e-01,
  2.98177754e-01],
[ 4.97030088e-02, -1.47800021e+00, -4.54867541e-01,
 -9.49450220e-01],
[ 2.96320992e-01,  5.45736761e-01, -4.53342300e-01,
  9.21991741e-01],
[-6.90150940e-01,  8.21700893e-01, -4.22559405e-01,
```

```
        1.26855507e+00],
       [-2.95562167e-01, -1.84595238e+00, -3.70102926e-01,
        -1.67723320e+00],
       [ 1.82535249e+00,  1.09766503e+00,  2.65506437e+00,
        1.09527340e+00],
       [ 1.52941091e+00,  4.53748717e-01, -4.29924379e-01,
        -1.52354570e-01],
       [-2.46238571e-01,  1.77784585e-01, -4.33581586e-01,
        6.10084748e-01],
       [-8.87445326e-01,  2.69772629e-01, -3.73962374e-01,
        6.44741080e-01],
       [-4.43532957e-01, -2.12191652e+00, -1.44687530e-01,
        -1.43463888e+00],
       [-4.43532957e-01, -3.74143679e-01, -4.88127898e-01,
        -8.30419047e-02],
       [ 2.96320992e-01,  8.57965411e-02, -4.70516004e-01,
        -6.37543226e-01]])
```

0.1.3 Model Training

0.1.4 Using PyCaret

[36]:
```
exp_reg = setup(data = df, target = 'audience_score', session_id=0,␣
↪normalize=True, train_size=0.8)
```

```
Setup Succesfully Completed!
```

```
<pandas.io.formats.style.Styler at 0x2398d721588>
```

[37]:
```
compare_models()
```

[37]:
```
<pandas.io.formats.style.Styler at 0x2398700c088>
```

[38]:
```
cat = create_model('catboost')
```

	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	6.8670	98.3357	9.9164	0.7679	0.2008	0.1336
1	6.5469	83.3773	9.1311	0.8106	0.1841	0.1350
2	6.2076	73.1855	8.5548	0.7758	0.1963	0.1404
3	7.0753	82.2519	9.0693	0.8329	0.2108	0.1706
4	6.8862	85.2206	9.2315	0.7333	0.1697	0.1216
5	6.0639	64.9538	8.0594	0.8510	0.1603	0.1175
6	6.4169	116.9859	10.8160	0.6794	0.2517	0.1210
7	7.2507	88.5254	9.4088	0.8136	0.2335	0.1830
8	6.4853	71.9568	8.4827	0.8093	0.1503	0.1129
9	6.6723	69.2329	8.3206	0.8462	0.2066	0.1614

```
Mean   6.6472   83.4026   9.0991   0.7920   0.1964   0.1397
SD     0.3573   14.6910   0.7807   0.0512   0.0301   0.0229
```

[39]:
```
print(cat)
```

```
<catboost.core.CatBoostRegressor object at 0x000002398FA577C8>
```

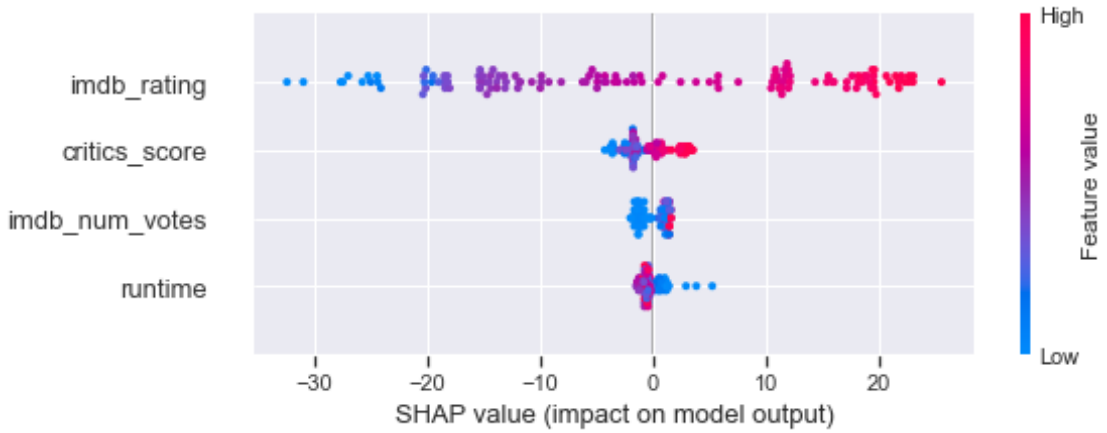
[40]:
```
tuned_cat = tune_model('catboost', optimize='mse')
```

```
          MAE       MSE      RMSE      R2    RMSLE    MAPE
0      6.5976   93.8717    9.6887  0.7784   0.2050  0.1321
1      6.5135   85.6355    9.2539  0.8054   0.1966  0.1428
2      5.4581   61.9992    7.8740  0.8100   0.1989  0.1365
3      7.5797   92.6546    9.6257  0.8117   0.2377  0.1936
4      7.9040  116.5350   10.7951  0.6352   0.2062  0.1411
5      5.5099   55.6068    7.4570  0.8725   0.1509  0.1081
6      6.5872  120.8227   10.9919  0.6689   0.2530  0.1246
7      7.1994   80.9686    8.9983  0.8296   0.2261  0.1802
8      6.0139   65.4295    8.0888  0.8266   0.1401  0.1025
9      7.0064   74.2622    8.6175  0.8351   0.2342  0.1839
Mean   6.6370   84.7786    9.1391  0.7873   0.2049  0.1445
SD     0.7755   20.7906    1.1204  0.0718   0.0345  0.0299
```

[41]:
```
print(tuned_cat)
```

```
<catboost.core.CatBoostRegressor object at 0x000002398FA5C648>
```

[42]:
```
interpret_model(tuned_cat)
```



[43]:
```
predict_model(tuned_cat)
```

```
                  Model     MAE     MSE   RMSE      R2   RMSLE    MAPE
0  CatBoost Regressor  6.5575  75.446  8.686  0.7964  0.1759  0.1295
```

[43]:

	runtime	imdb_rating	imdb_num_votes	critics_score	audience_score	Label
0	-0.660154	-0.731689	-0.342490	-1.433634	44	43.2409
1	-0.969186	-4.237505	0.140006	-1.997388	19	28.2623
2	-0.608649	-0.178139	0.050751	0.786150	49	58.1592
3	0.781995	0.190894	0.012264	0.328099	70	63.5129
4	0.781995	0.098636	-0.480220	-0.306125	66	58.0552
..	…	…	…	…	…	…
126	-0.248112	0.190894	-0.430585	0.645211	71	63.8282
127	-0.917681	0.283152	-0.367437	0.680446	73	70.9544
128	-0.454133	-2.115564	-0.124591	-1.433634	51	31.5633
129	-0.454133	-0.362656	-0.488360	-0.059482	40	46.4926
130	0.318447	0.098636	-0.469706	-0.623237	57	57.5242

[131 rows x 6 columns]