Data Description

GroupLens Research has collected and made available rating data sets from the MovieLens web site (http://movielens.org). The data sets were collected over various periods of time, depending on the size of the set.

100,000 ratings and 3,600 tag applications applied to 9,000 movies by 600 users. Last updated 9/2018.

In [34]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
import datetime

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

from pycaret.regression import *

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 warnings
warnings.filterwarnings('ignore')

pd.set_option('display.max_columns',None)
#pd.set_option('display.max_rows',None)

np.random.seed(0)
np.set_printoptions(suppress=True)
```

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

In [3]:
```
ratings = pd.read_csv("ratings.csv")
```

In [4]:
```
movies.head()
```

Out[4]:

	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 [5]:
```
ratings.head()
```

Out[5]:

	userId	movieId	rating	timestamp
0	1	1	4.0	964982703
1	1	3	4.0	964981247
2	1	6	4.0	964982224
3	1	47	5.0	964983815
4	1	50	5.0	964982931

In [6]:
```
df = pd.merge(ratings,movies, on='movieId')
```

In [7]:
```
df
```

Out[7]:

	userId	movieId	rating	timestamp	title	genres
0	1	1	4.0	964982703	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	5	1	4.0	847434962	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
2	7	1	4.5	1106635946	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
3	15	1	2.5	1510577970	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
4	17	1	4.5	1305696483	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
...	...	...	...	...	...	...
100831	610	160341	2.5	1479545749	Bloodmoon (1997)	Action|Thriller
100832	610	160527	4.5	1479544998	Sympathy for the Underdog (1971)	Action|Crime|Drama
100833	610	160836	3.0	1493844794	Hazard (2005)	Action|Drama|Thriller
100834	610	163937	3.5	1493848789	Blair Witch (2016)	Horror|Thriller
100835	610	163981	3.5	1493850155	31 (2016)	Horror

100836 rows × 6 columns

In [8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 100836 entries, 0 to 100835
Data columns (total 6 columns):
 #   Column     Non-Null Count   Dtype
---  ------     --------------   -----
 0   userId     100836 non-null  int64
 1   movieId    100836 non-null  int64
 2   rating     100836 non-null  float64
 3   timestamp  100836 non-null  int64
 4   title      100836 non-null  object
 5   genres     100836 non-null  object
dtypes: float64(1), int64(3), object(2)
memory usage: 5.4+ MB
```

In [9]:
```
df.groupby('title')['rating'].mean().sort_values(ascending=False).head()
```

Out[9]:
```
title
Karlson Returns (1970)                           5.0
Winter in Prostokvashino (1984)                  5.0
My Love (2006)                                   5.0
Sorority House Massacre II (1990)                5.0
Winnie the Pooh and the Day of Concern (1972)    5.0
Name: rating, dtype: float64
```

In [10]:
```
df.groupby('title')['rating'].count().sort_values(ascending=False).head()
```

Out[10]:
```
title
Forrest Gump (1994)                 329
Shawshank Redemption, The (1994)    317
Pulp Fiction (1994)                 307
Silence of the Lambs, The (1991)    279
Matrix, The (1999)                  278
Name: rating, dtype: int64
```

In [11]:
```
rating = pd.DataFrame(df.groupby('title')['rating'].mean())
rating
```

Out[11]:

title	rating
'71 (2014)	4.000000
'Hellboy': The Seeds of Creation (2004)	4.000000
'Round Midnight (1986)	3.500000
'Salem's Lot (2004)	5.000000
'Til There Was You (1997)	4.000000
...	...
eXistenZ (1999)	3.863636
xXx (2002)	2.770833
xXx: State of the Union (2005)	2.000000
¡Three Amigos! (1986)	3.134615
À nous la liberté (Freedom for Us) (1931)	1.000000

9719 rows × 1 columns

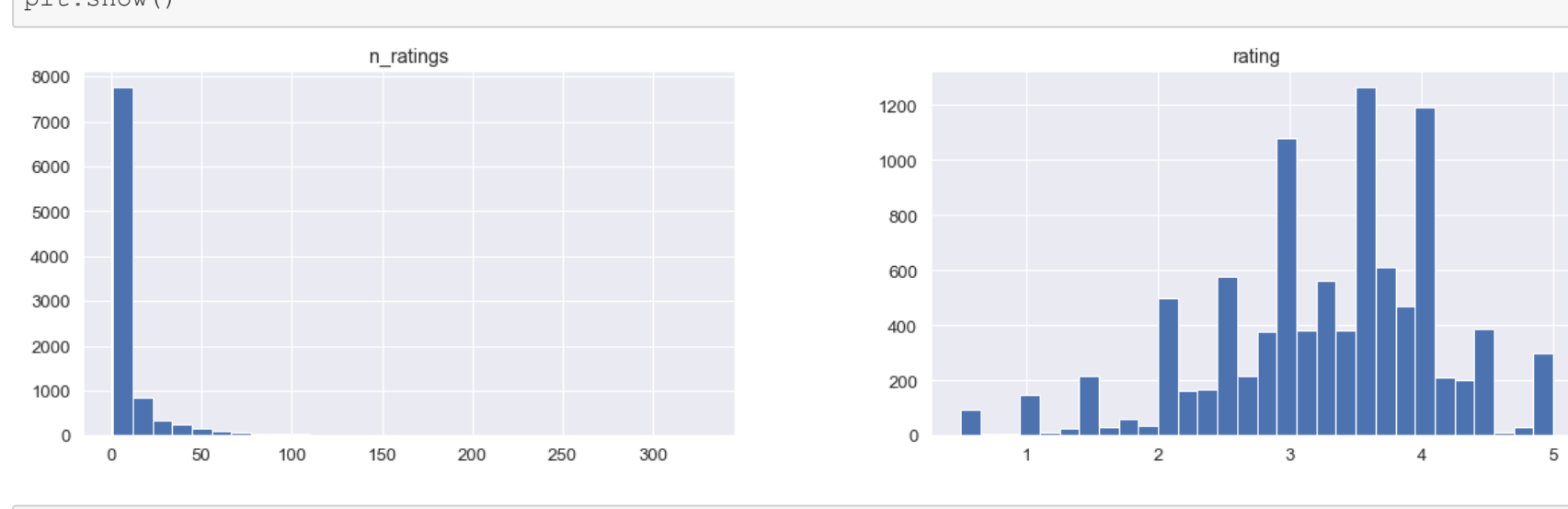
In [12]:
```
rating['n_ratings'] = pd.DataFrame(df.groupby('title')['rating'].count())
rating
```

Out[12]:

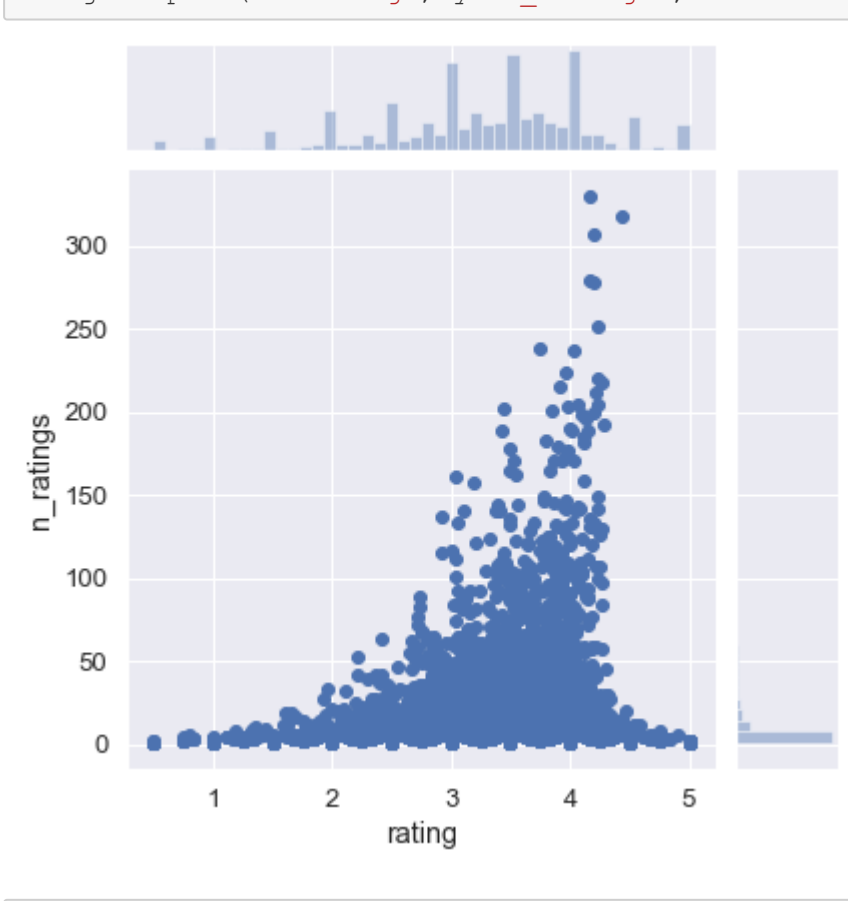
title	rating	n_ratings
'71 (2014)	4.000000	1
'Hellboy': The Seeds of Creation (2004)	4.000000	1
'Round Midnight (1986)	3.500000	2
'Salem's Lot (2004)	5.000000	1
'Til There Was You (1997)	4.000000	2
...	...	...
eXistenZ (1999)	3.863636	22
xXx (2002)	2.770833	24
xXx: State of the Union (2005)	2.000000	5
¡Three Amigos! (1986)	3.134615	26
À nous la liberté (Freedom for Us) (1931)	1.000000	1

9719 rows × 2 columns

In [13]:
```
rating.hist(figsize=(20,5), bins=30)
plt.show()
```



In [14]:
```
sns.jointplot(x='rating', y='n_ratings',data=rating);
```



In [15]:
```
df.head()
```

Out[15]:

	userId	movieId	rating	timestamp	title	genres
0	1	1	4.0	964982703	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	5	1	4.0	847434962	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
2	7	1	4.5	1106635946	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
3	15	1	2.5	1510577970	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
4	17	1	4.5	1305696483	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy

In [16]:
```
rating_matrix = df.pivot_table(values='rating',index='userId',columns='title')
```

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

Out[17]:

title	'71 (2014)	'Hellboy': The Seeds of Creation (2004)	'Round Midnight (1986)	'Salem's Lot (2004)	'Til There Was You (1997)	'Tis the Season for Love (2015)	'burbs, The (1989)	'night Mother (1986)	(500) Days of Summer (2009)	*batteries not included (1987)	...All the Marbles (1981)	...And Justice for All (1979)	00 Schneider - Jagd auf Nihil Baxter (1994)	1-900 (06) (1994)	(
userId															
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

In [18]:
```
rating.sort_values('n_ratings', ascending = False).head()
```

Out[18]:

title	rating	n_ratings
Forrest Gump (1994)	4.164134	329
Shawshank Redemption, The (1994)	4.429022	317
Pulp Fiction (1994)	4.197068	307
Silence of the Lambs, The (1991)	4.161290	279
Matrix, The (1999)	4.192446	278

In [19]:
```
forrestgumprating = rating_matrix['Forrest Gump (1994)']
```

In [20]:
```
forrestgumprating
```

Out[20]:
```
userId
1      4.0
2      NaN
3      NaN
4      NaN
5      NaN
      ...
606    4.0
607    NaN
608    3.0
609    4.0
610    3.0
Name: Forrest Gump (1994), Length: 610, dtype: float64
```

In [21]:
```
similar_to_FG = rating_matrix.corrwith(forrestgumprating)
```

In [22]:
```
similar_to_FG
```

Out[22]:
```
title
'71 (2014)                                        NaN
'Hellboy': The Seeds of Creation (2004)           NaN
'Round Midnight (1986)                            NaN
'Salem's Lot (2004)                               NaN
'Til There Was You (1997)                         NaN
                                                ...
eXistenZ (1999)                              0.011189
xXx (2002)                                   0.306817
xXx: State of the Union (2005)               0.383482
¡Three Amigos! (1986)                        0.449692
À nous la liberté (Freedom for Us) (1931)         NaN
Length: 9719, dtype: float64
```

In [23]:
```
corr_FG = pd.DataFrame(similar_to_FG, columns=['correlation'])
```

In [24]:
```
corr_FG
```

Out[24]:

title	correlation
'71 (2014)	NaN
'Hellboy': The Seeds of Creation (2004)	NaN
'Round Midnight (1986)	NaN
'Salem's Lot (2004)	NaN
'Til There Was You (1997)	NaN
...	...
eXistenZ (1999)	0.011189
xXx (2002)	0.306817
xXx: State of the Union (2005)	0.383482
¡Three Amigos! (1986)	0.449692
À nous la liberté (Freedom for Us) (1931)	NaN

9719 rows × 1 columns

In [25]:
```
corr_FG.dropna(inplace=True)
```

In [26]:
```
corr_FG
```

Out[26]:

title	correlation
'burbs, The (1989)	0.197712
(500) Days of Summer (2009)	0.234095
*batteries not included (1987)	0.892710
...And Justice for All (1979)	0.928571
10 Cent Pistol (2015)	-1.000000
...	...
[REC]² (2009)	-0.052414
eXistenZ (1999)	0.011189
xXx (2002)	0.306817
xXx: State of the Union (2005)	0.383482
¡Three Amigos! (1986)	0.449692

5460 rows × 1 columns

In [27]:
```
corr_FG.sort_values('correlation', ascending=False)
```

Out[27]:

title	correlation
Lost & Found (1999)	1.0
Century of the Self, The (2002)	1.0
The 5th Wave (2016)	1.0
Play Time (a.k.a. Playtime) (1967)	1.0
Memories (Memorîzu) (1995)	1.0
...	...
Jupiter Ascending (2015)	-1.0
The Cobbler (2015)	-1.0
The Amazing Screw-On Head (2006)	-1.0
Captain America (1990)	-1.0
Jack Whitehall: At Large (2017)	-1.0

5460 rows × 1 columns

In [28]:
```
corr_matrix = corr_FG.join(rating['n_ratings'])
```

In [29]:
```
corr_matrix
```

Out[29]:

title	correlation	n_ratings
'burbs, The (1989)	0.197712	17
(500) Days of Summer (2009)	0.234095	42
*batteries not included (1987)	0.892710	7
...And Justice for All (1979)	0.928571	3
10 Cent Pistol (2015)	-1.000000	2
...	...	...
[REC]² (2009)	-0.052414	3
eXistenZ (1999)	0.011189	22
xXx (2002)	0.306817	24
xXx: State of the Union (2005)	0.383482	5
¡Three Amigos! (1986)	0.449692	26

5460 rows × 2 columns

In [30]:
```
corr_matrix[corr_matrix['n_ratings'] > 50].sort_values('correlation', ascending=False)
```

Out[30]:

title	correlation	n_ratings
Forrest Gump (1994)	1.000000	329
Mr. Holland's Opus (1995)	0.652144	80
Pocahontas (1995)	0.550118	68
Grumpier Old Men (1995)	0.534682	52
Caddyshack (1980)	0.520328	52
...	...	...
Harry Potter and the Prisoner of Azkaban (2004)	-0.131888	93
Ed Wood (1994)	-0.163673	70
Zoolander (2001)	-0.182810	54
Best in Show (2000)	-0.241700	53
Arachnophobia (1990)	-0.265720	53

437 rows × 2 columns

In [31]:
```
rating.shape
```

Out[31]: (9719, 2)

In [32]:
```
corr_matrix.shape
```

Out[32]: (5460, 2)

In [35]:
```
rating
```

Out[35]:

title	rating	n_ratings
'71 (2014)	4.000000	1
'Hellboy': The Seeds of Creation (2004)	4.000000	1
'Round Midnight (1986)	3.500000	2
'Salem's Lot (2004)	5.000000	1
'Til There Was You (1997)	4.000000	2
...	...	...
eXistenZ (1999)	3.863636	22
xXx (2002)	2.770833	24
xXx: State of the Union (2005)	2.000000	5
¡Three Amigos! (1986)	3.134615	26
À nous la liberté (Freedom for Us) (1931)	1.000000	1

9719 rows × 2 columns

In [36]:
```
rating.reset_index(inplace=True)
```

In [37]:
```
rating
```

Out[37]:

	title	rating	n_ratings
0	'71 (2014)	4.000000	1
1	'Hellboy': The Seeds of Creation (2004)	4.000000	1
2	'Round Midnight (1986)	3.500000	2
3	'Salem's Lot (2004)	5.000000	1
4	'Til There Was You (1997)	4.000000	2
...	...	...	...
9714	eXistenZ (1999)	3.863636	22
9715	xXx (2002)	2.770833	24
9716	xXx: State of the Union (2005)	2.000000	5
9717	¡Three Amigos! (1986)	3.134615	26
9718	À nous la liberté (Freedom for Us) (1931)	1.000000	1

9719 rows × 3 columns

In [38]:
```
rating.drop(["title"],axis=1,inplace=True)
```

In [39]:
```
rating
```

Out[39]:

	rating	n_ratings
0	4.000000	1
1	4.000000	1
2	3.500000	2
3	5.000000	1
4	4.000000	2
...	...	...
9714	3.863636	22
9715	2.770833	24
9716	2.000000	5
9717	3.134615	26
9718	1.000000	1

9719 rows × 2 columns

In [40]:
```
exp_reg = setup(data = rating, target = 'rating', session_id=123, normalize=True)
```

Setup Succesfully Completed.

	Description	Value
0	session_id	123
1	Transform Target	False
2	Transform Target Method	None
3	Original Data	(9719, 2)
4	Missing Values	False
5	Numeric Features	1
6	Categorical Features	0
7	Ordinal Features	False
8	High Cardinality Features	False
9	High Cardinality Method	None
10	Sampled Data	(9719, 2)
11	Transformed Train Set	(6803, 1)
12	Transformed Test Set	(2916, 1)
13	Numeric Imputer	mean
14	Categorical Imputer	constant
15	Normalize	True
16	Normalize Method	zscore
17	Transformation	False
18	Transformation Method	None
19	PCA	False
20	PCA Method	None
21	PCA Components	None
22	Ignore Low Variance	False
23	Combine Rare Levels	False
24	Rare Level Threshold	None
25	Numeric Binning	False
26	Remove Outliers	False
27	Outliers Threshold	None
28	Remove Multicollinearity	False
29	Multicollinearity Threshold	None
30	Clustering	False
31	Clustering Iteration	None
32	Polynomial Features	False
33	Polynomial Degree	None
34	Trignometry Features	False
35	Polynomial Threshold	None
36	Group Features	False
37	Feature Selection	False
38	Features Selection Threshold	None
39	Feature Interaction	False
40	Feature Ratio	False
41	Interaction Threshold	None

In [41]:
```
compare_models()
```

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
0	Linear Regression	0.6718	0.7428	0.8615	0.0157	0.2341	0.2983	0.0063
1	Ridge Regression	0.6718	0.7428	0.8615	0.0157	0.2341	0.2983	0.0024
2	Least Angle Regression	0.6718	0.7428	0.8615	0.0157	0.2341	0.2983	0.0050
3	Orthogonal Matching Pursuit	0.6718	0.7428	0.8615	0.0157	0.2341	0.2983	0.0022
4	Bayesian Ridge	0.6718	0.7428	0.8615	0.0157	0.2341	0.2983	0.0080
5	Gradient Boosting Regressor	0.6724	0.7437	0.8620	0.0146	0.2342	0.2986	0.1113
6	Light Gradient Boosting Machine	0.6728	0.7445	0.8625	0.0135	0.2343	0.2988	0.1959
7	CatBoost Regressor	0.6734	0.7447	0.8626	0.0132	0.2343	0.2989	2.9399
8	Random Sample Consensus	0.6689	0.7447	0.8626	0.0131	0.2355	0.3017	0.1392
9	Random Forest	0.6743	0.7459	0.8633	0.0116	0.2344	0.2990	0.2983
10	Extreme Gradient Boosting	0.6749	0.7466	0.8637	0.0106	0.2345	0.2993	0.3493
11	Extra Trees Regressor	0.6750	0.7468	0.8638	0.0104	0.2345	0.2993	0.1613
12	Decision Tree	0.6752	0.7469	0.8639	0.0103	0.2346	0.2994	0.0022
13	Huber Regressor	0.6684	0.7478	0.8644	0.0089	0.2365	0.3039	0.0199
14	AdaBoost Regressor	0.6777	0.7481	0.8645	0.0088	0.2327	0.2930	0.0470
15	Lasso Regression	0.6818	0.7554	0.8688	-0.0010	0.2363	0.3038	0.0047
16	Elastic Net	0.6818	0.7554	0.8688	-0.0010	0.2363	0.3038	0.0066
17	Lasso Least Angle Regression	0.6818	0.7554	0.8688	-0.0010	0.2363	0.3038	0.0052
18	Support Vector Machine	0.6659	0.7587	0.8707	-0.0054	0.2394	0.3095	1.5300
19	TheilSen Regressor	0.7381	0.8867	0.9412	-0.1753	0.2576	0.3236	2.9504
20	K Neighbors Regressor	0.7417	0.9583	0.9745	-0.2699	0.2607	0.3354	0.0154
21	Passive Aggressive Regressor	0.8897	1.2994	1.1206	-0.7261	0.2810	0.3420	0.0071

Out[41]: LinearRegression(copy_X=True, fit_intercept=True, n_jobs=-1, normalize=False)

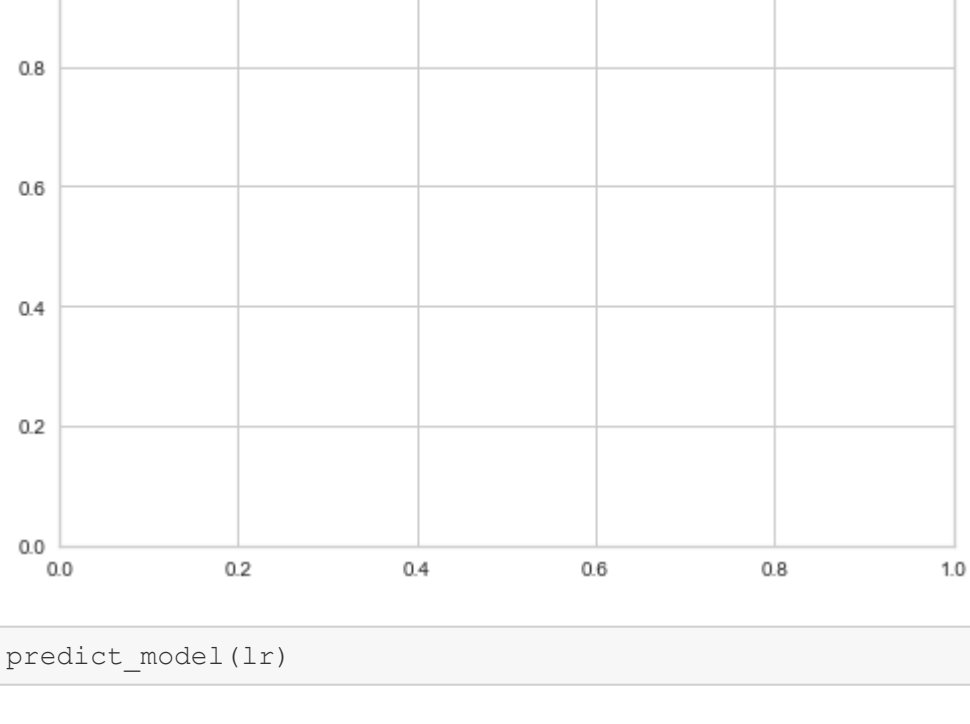
In [42]:
```
lr = create_model('lr')
```

	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	0.7134	0.8499	0.9219	0.0158	0.2598	0.3533
1	0.6540	0.7025	0.8382	0.0271	0.2239	0.2724
2	0.6731	0.7724	0.8789	0.0166	0.2453	0.3204
3	0.6752	0.7457	0.8635	0.0053	0.2316	0.2890
4	0.6884	0.7709	0.8780	0.0133	0.2377	0.3068
5	0.6621	0.7103	0.8428	0.0199	0.2297	0.2931
6	0.6630	0.7391	0.8597	0.0162	0.2368	0.3069
7	0.6731	0.7334	0.8564	0.0124	0.2281	0.2864
8	0.6593	0.7094	0.8422	0.0190	0.2247	0.2755
9	0.6561	0.6945	0.8333	0.0117	0.2232	0.2789
Mean	0.6718	0.7428	0.8615	0.0157	0.2341	0.2983
SD	0.0171	0.0440	0.0251	0.0055	0.0109	0.0234

In [43]:
```
print(lr)
```

LinearRegression(copy_X=True, fit_intercept=True, n_jobs=-1, normalize=False)

In [44]:
```
evaluate_model(lr)
```



In [45]:
```
predict_model(lr)
```

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	Linear Regression	0.6671	0.7491	0.8655	0.015	0.2387	0.3067

Out[45]:

	n_ratings	rating	Label
0	-0.418438	4.000000	3.2154
1	-0.373806	3.750000	3.2203
2	-0.418438	4.500000	3.2154
3	1.902465	3.688679	3.4724
4	-0.150642	3.857143	3.2450
...	...	...	...
2911	-0.418438	4.000000	3.2154
2912	-0.418438	4.500000	3.2154
2913	-0.418438	3.000000	3.2154
2914	-0.418438	5.000000	3.2154
2915	-0.418438	3.500000	3.2154

2916 rows × 3 columns

Quite far off the predicted values

In [ ]: