```
In [21]: # Importing all necessary packages

import numpy as np
import pandas as pd
import seaborn as sns
import itertools
from matplotlib import pyplot
%matplotlib inline

import missingno

import warnings
warnings.filterwarnings('ignore')

from datetime import datetime

from sklearn.model_selection import train_test_split, StratifiedKFold

from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.ensemble import BaggingRegressor

from catboost import CatBoostClassifier, CatBoostRegressor, Pool, cv

import xgboost as xgb

from sklearn.metrics import mean_squared_error

from sklearn.preprocessing import OneHotEncoder, LabelEncoder

from scipy.special import boxcox,inv_boxcox
```

Reading Datasets

```
In [22]: train_file = pd.read_csv("/home/jiten/Desktop/ML Submissions/MachineHack/VideoGames/Data/Train.csv")

test_file = pd.read_csv("/home/jiten/Desktop/ML Submissions/MachineHack/VideoGames/Data/Test.csv")
```

```
In [23]: len(train_file)
Out[23]: 3506
```

```
In [24]: len(test_file)
Out[24]: 1503
```

```
In [25]: train_file.describe()
```

Out[25]:

	ID	YEAR	CRITICS_POINTS	USER_POINTS	SalesInMillions
count	3506.000000	3506.000000	3506.000000	3506.000000	3506.000000
mean	2282.233600	2008.990302	3.790831	0.405824	2.171021
std	1287.273422	4.304252	3.141781	0.455541	2.495396
min	1.000000	1997.000000	0.568966	0.000341	0.001524
25%	1220.250000	2006.000000	1.738095	0.065966	0.965679
50%	2262.500000	2009.000000	2.766667	0.233333	1.866140
75%	3404.750000	2012.000000	4.621528	0.598333	2.792029
max	4523.000000	2019.000000	23.250000	2.325000	84.226041

```
In [26]: train_file.head()
```

Out[26]:

	ID	CONSOLE	YEAR	CATEGORY	PUBLISHER	RATING	CRITICS_POINTS	USER_POINTS	SalesInMillions
0	2860	ds	2008	role-playing	Nintendo	E	2.833333	0.303704	1.779257
1	731	wii	2012	simulation	Konami Digital Entertainment	E10+	13.200000	1.640000	0.215050
2	495	pc	2019	shooter	Activision	M	4.562500	0.006410	0.534402
3	2641	ps2	2002	sports	Electronic Arts	E	4.181818	0.326923	1.383964
4	811	ps3	2013	action	Activision	M	2.259259	0.032579	0.082671

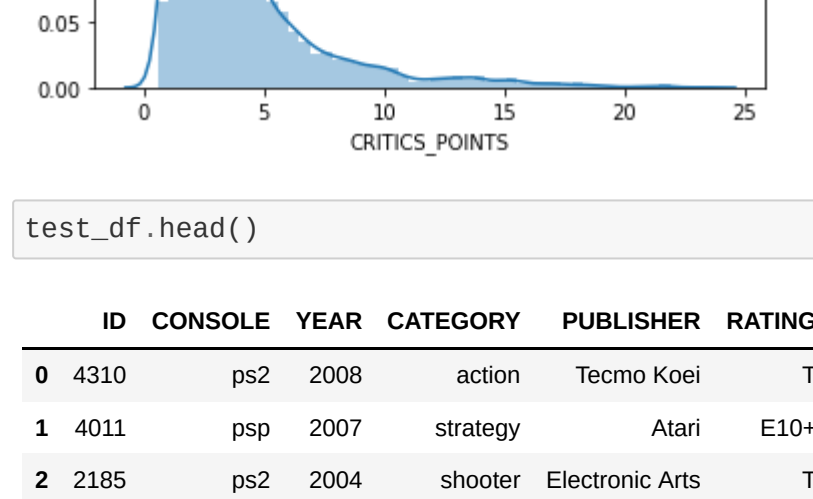
```
In [27]: train_df = train_file.copy()

test_df = test_file.copy()

train_df = train_df.drop_duplicates()

len(train_df)
Out[27]: 3506
```

```
In [28]: train_df.head()
```

Out[28]:

	ID	CONSOLE	YEAR	CATEGORY	PUBLISHER	RATING	CRITICS_POINTS	USER_POINTS	SalesInMillions
0	2860	ds	2008	role-playing	Nintendo	E	2.833333	0.303704	1.779257
1	731	wii	2012	simulation	Konami Digital Entertainment	E10+	13.200000	1.640000	0.215050
2	495	pc	2019	shooter	Activision	M	4.562500	0.006410	0.534402
3	2641	ps2	2002	sports	Electronic Arts	E	4.181818	0.326923	1.383964
4	811	ps3	2013	action	Activision	M	2.259259	0.032579	0.082671

```
In [29]: sns.distplot(train_file.CRITICS_POINTS)
Out[29]: <matplotlib.axes._subplots.AxesSubplot at 0x7f28b7734dd0>
```




```
In [30]: test_df.head()
```

Out[30]:

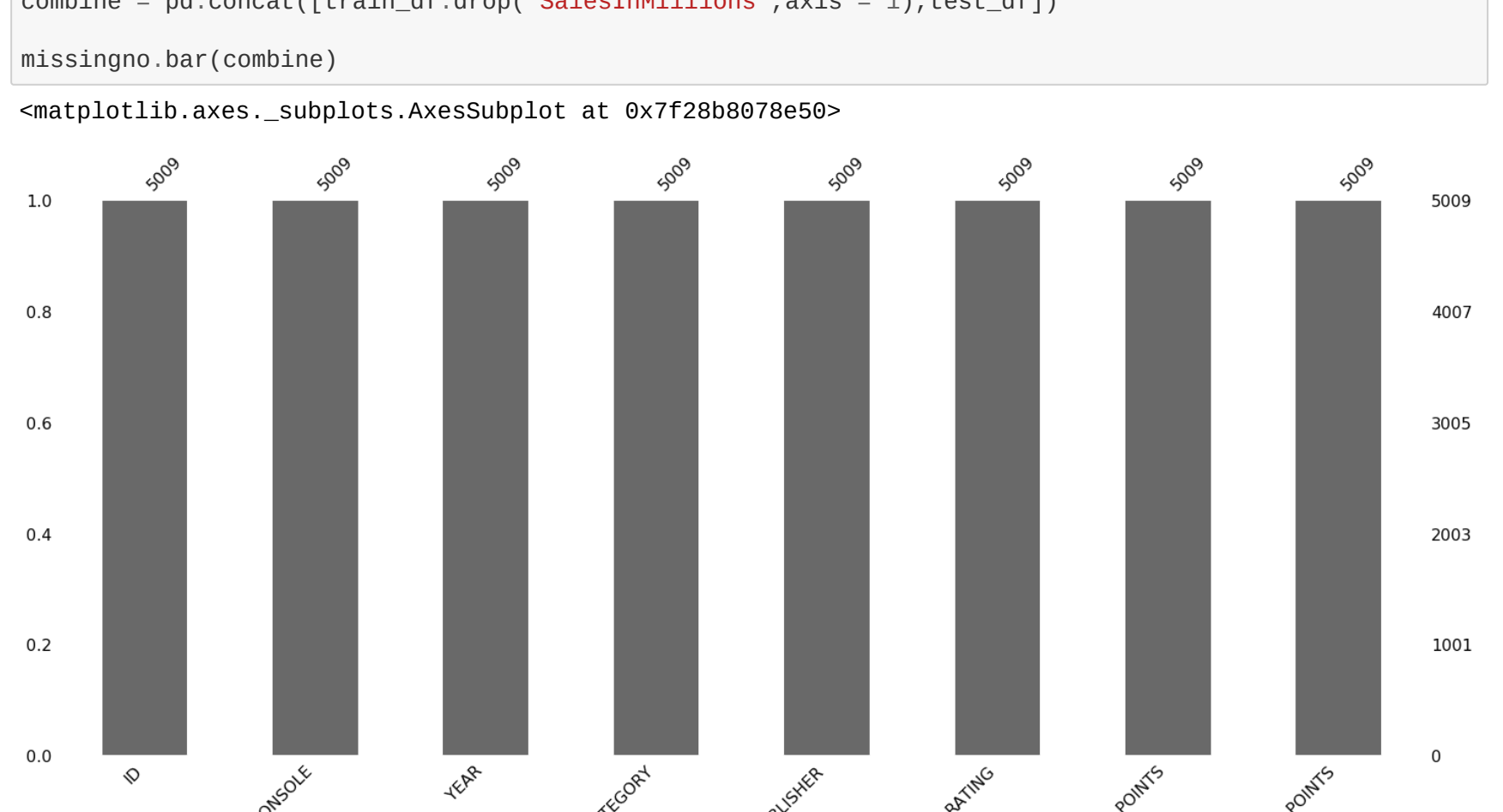
	ID	CONSOLE	YEAR	CATEGORY	PUBLISHER	RATING	CRITICS_POINTS	USER_POINTS
0	4310	ps2	2008	action	Tecmo Koei	T	3.928571	0.482353
1	4011	psp	2007	strategy	Atari	E10+	5.538462	0.071795
2	2185	ps2	2004	shooter	Electronic Arts	T	3.034483	0.062044
3	1644	x	2006	action	Electronic Arts	E	2.913043	0.880000
4	188	3ds	2011	racing	Ubisoft	E10+	1.162162	0.183333

```
In [31]: min(train_df.ID)
Out[31]: 1
```

```
In [32]: min(test_df.ID)
Out[32]: 0
```

Combining Train and Test sets for EDA

```
In [33]: combine = pd.concat([train_df.drop("SalesInMillions",axis = 1),test_df])

missingno.bar(combine)
Out[33]: <matplotlib.axes._subplots.AxesSubplot at 0x7f28b8078e50>
```



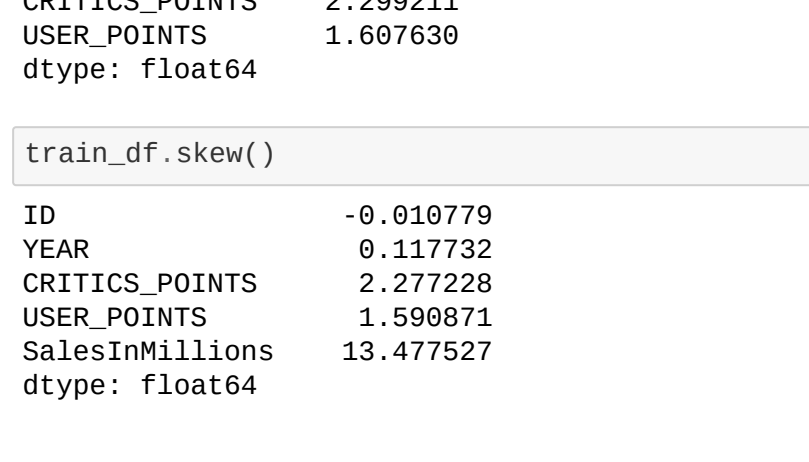

```
In [34]: combine.skew()
Out[34]: ID               -0.025973
YEAR              0.060762
CRITICS_POINTS    2.299211
USER_POINTS       1.607630
dtype: float64
```

```
In [35]: train_df.skew()
Out[35]: ID                -0.010779
YEAR               0.117732
CRITICS_POINTS     2.277228
USER_POINTS        1.590871
SalesInMillions   13.477527
dtype: float64
```

Univartiate Analysis of Features

Output Variable

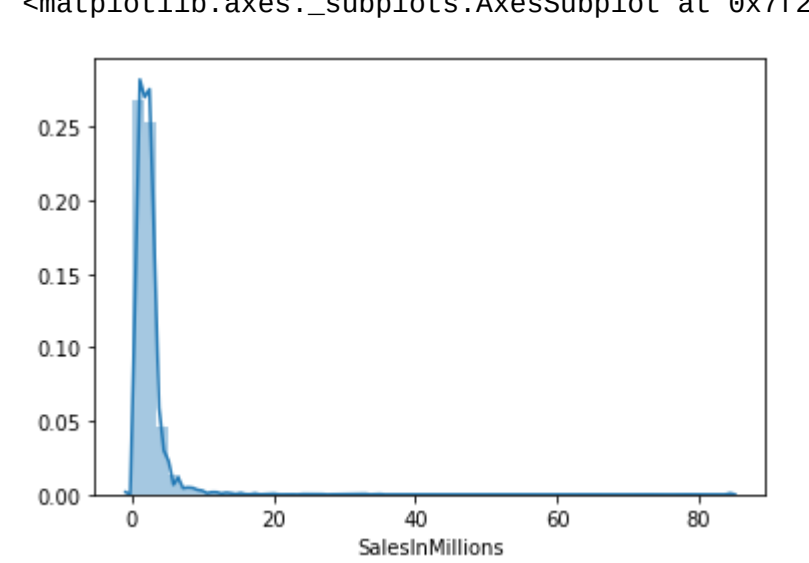
```
In [36]: sns.distplot(train_df.SalesInMillions)
Out[36]: <matplotlib.axes._subplots.AxesSubplot at 0x7f28b7179cd0>
```




```
In [37]: def modify_a_col(col,x):

    col = boxcox(col, x)

    print(col.skew())

    sns.distplot(col)
```

Reduced Skewness of Output Variable by applying boxcox transformation

```
In [38]: modify_a_col(train_df["SalesInMillions"],0.27)

train_df["SalesInMillions"] = boxcox(train_df["SalesInMillions"], 0.27)

-0.0009362893164543252
```



ID col cannot be ignored since it has repeated values

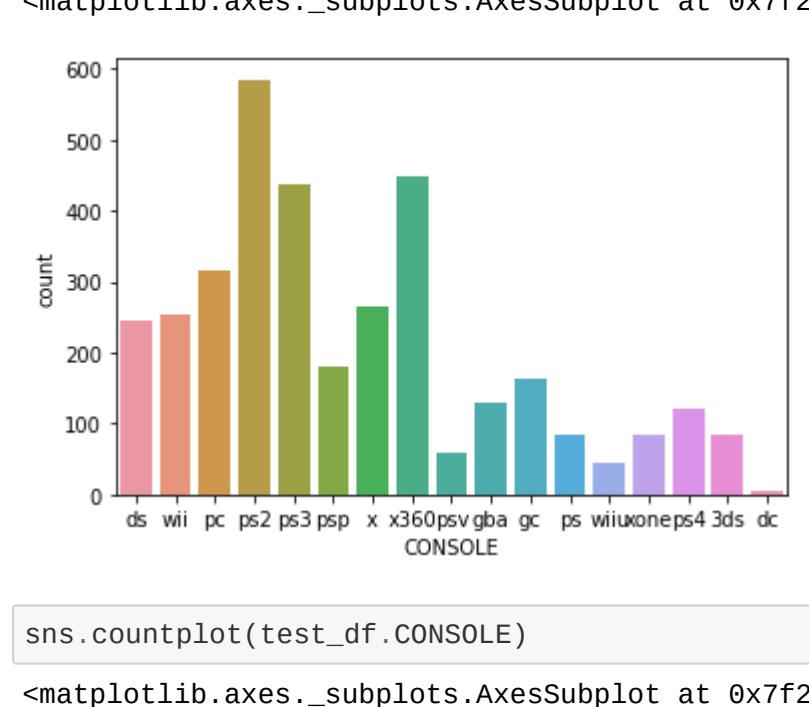
```
In [39]: train_file.columns
Out[39]: Index(['ID', 'CONSOLE', 'YEAR', 'CATEGORY', 'PUBLISHER', 'RATING',
       'CRITICS_POINTS', 'USER_POINTS', 'SalesInMillions'],
      dtype='object')
```

CONSOLE COLUMN

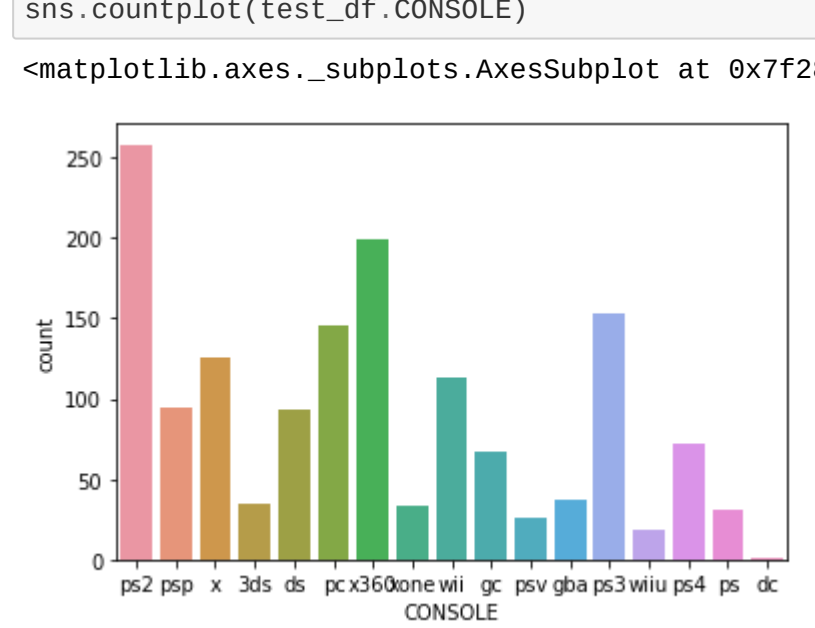
```
In [42]: if set(train_df.CONSOLE.unique()) == set(test_df.CONSOLE.unique()):

    print("All consoles are perfect !")

All consoles are perfect !
```

```
In [43]: print(set(train_df.CONSOLE))

{'x360', 'psp', 'ps2', 'ds', 'gc', 'ps', 'wiiu', 'ps4', '3ds', 'dc', 'xone', 'x', 'psv', 'pc', 'gba',
'wii', 'ps3'}
```

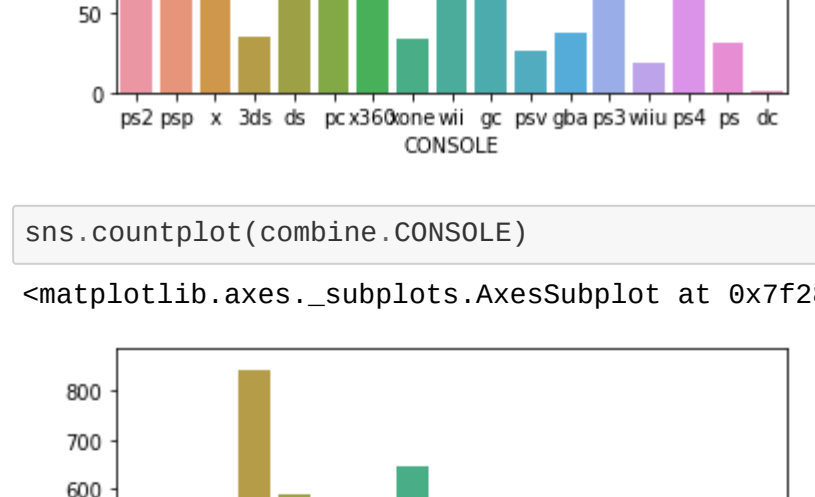
```
In [44]: sns.countplot(train_df.CONSOLE)
Out[44]: <matplotlib.axes._subplots.AxesSubplot at 0x7f28b7165950>
```




```
In [45]: sns.countplot(test_df.CONSOLE)
Out[45]: <matplotlib.axes._subplots.AxesSubplot at 0x7f28b6f975d0>
```




```
In [46]: sns.countplot(combine.CONSOLE)
Out[46]: <matplotlib.axes._subplots.AxesSubplot at 0x7f28b6ec2710>
```




```
In [47]: train_df.CONSOLE.value_counts()
Out[47]: ps2     585
x360    448
ps3     438
pc      315
x       266
wii     254
ds      246
psp     180
gc      164
gba     129
ps4     121
xone     85
3ds      84
ps       83
psv      59
wiiu     45
dc        4
Name: CONSOLE, dtype: int64
```

```
In [48]: test_df.CONSOLE.value_counts()
Out[48]: ps2     258
x360    199
ps3     153
pc      146
x       125
wii     113
psp      94
ds       93
ps4      72
gc       67
gba      37
3ds      35
xone     34
ps       31
psv      26
wiiu     19
dc        1
Name: CONSOLE, dtype: int64
```

```
In [49]: train_df.head()
```

Out[49]:

	ID	CONSOLE	YEAR	CATEGORY	PUBLISHER	RATING	CRITICS_POINTS	USER_POINTS	SalesInMillions
0	2860	ds	2008	role-playing	Nintendo	E	2.833333	0.303704	0.623434
1	731	wii	2012	simulation	Konami Digital Entertainment	E10+	13.200000	1.640000	-1.257900
2	495	pc	2019	shooter	Activision	M	4.562500	0.006410	-0.576469
3	2641	ps2	2002	sports	Electronic Arts	E	4.181818	0.326923	0.339633
4	811	ps3	2013	action	Activision	M	2.259259	0.032579	-1.814317

```
In [50]: x = train_df.YEAR.unique()

print(list(set(x)))

[1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 201
3, 2014, 2015, 2016, 2017, 2018, 2019]
```

```
In [51]: y = test_df.YEAR.unique()

print(list(set(y)))

[1988, 1994, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 201
1, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019]
```

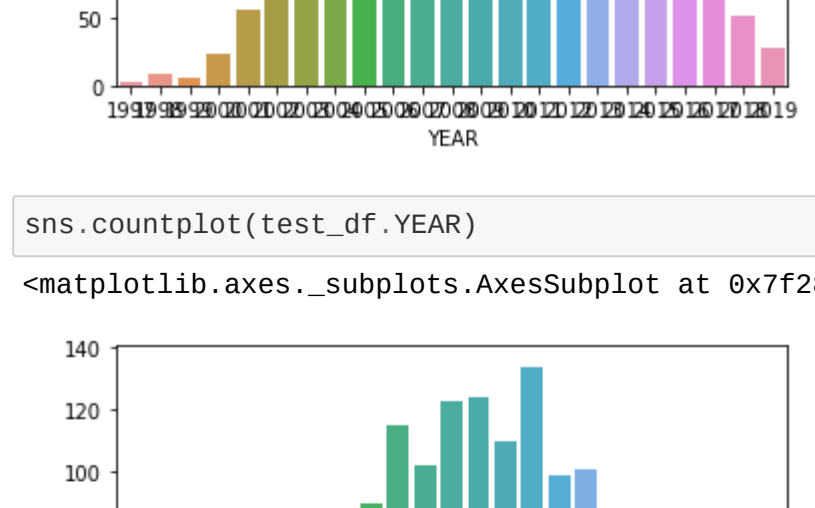
```
In [52]: print(train_df.YEAR.unique().sort() == test_df.YEAR.unique().sort())

True
```

```
In [53]: bins = [1950,2004, 2014,5000]

labels = ["age-1","age-2","age-3"]
```

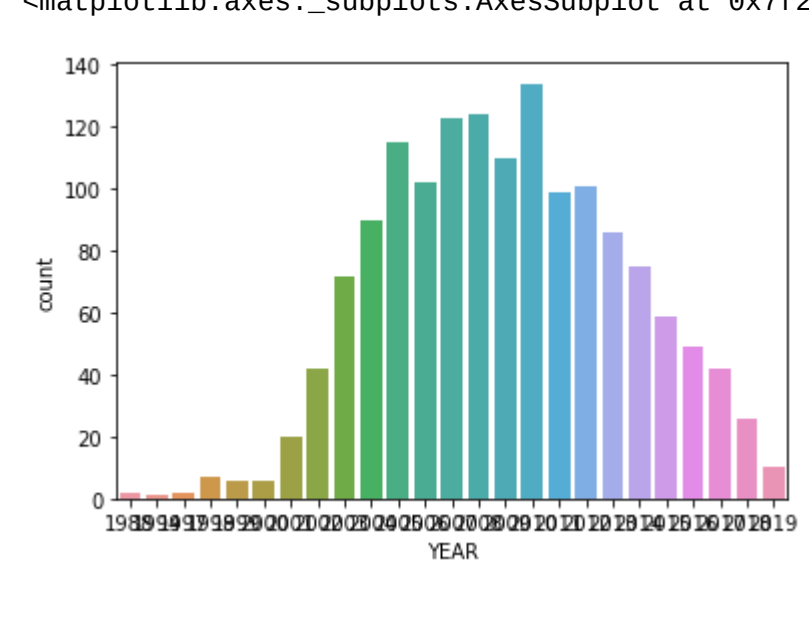
<= 2003, 2004 - 2013, >= 2014

```
In [ ]:
```

```
In [54]: sns.countplot(train_df.YEAR)
Out[54]: <matplotlib.axes._subplots.AxesSubplot at 0x7f28b6e1b390>
```




```
In [55]: sns.countplot(test_df.YEAR)
Out[55]: <matplotlib.axes._subplots.AxesSubplot at 0x7f28b6d633d0>
```



Binning Year Feature did not improve any accuracy

```
In [ ]: # train_df["YEAR"] = pd.cut(train_df["YEAR"], bins=bins, labels=labels)
# test_df["YEAR"] = pd.cut(test_df["YEAR"], bins=bins, labels=labels)
# combine["YEAR"] = pd.cut(combine["YEAR"], bins=bins, labels=labels)
```

No modifications were made to CATEGORY feature

```
In [63]: train_df.head()
```

Out[63]:

	ID	CONSOLE	YEAR	CATEGORY	PUBLISHER	RATING	CRITICS_POINTS	USER_POINTS	SalesInMillions
0	2860	ds	2008	role-playing	Nintendo	E	2.833333	0.303704	0.623434
1	731	wii	2012	simulation	Konami Digital Entertainment	E10+	13.200000	1.640000	-1.257900
2	495	pc	2019	shooter	Activision	M	4.562500	0.006410	-0.576469
3	2641	ps2	2002	sports	Electronic Arts	E	4.181818	0.326923	0.339633
4	811	ps3	2013	action	Activision	M	2.259259	0.032579	-1.814317

```
In [65]: print(train_df.CATEGORY.unique().sort() == test_df.CATEGORY.unique().sort())

True
```

```
In [66]: len(set(train_df.CATEGORY))
Out[66]: 12
```

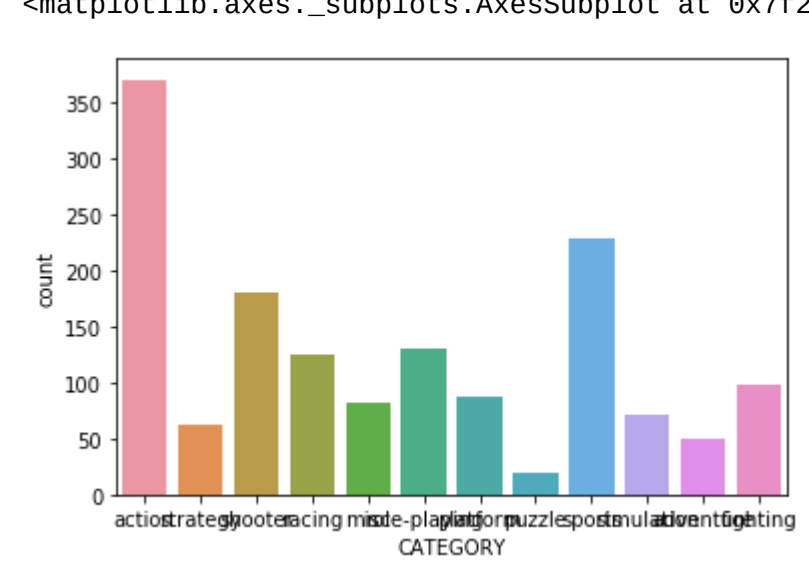
```
In [67]: set(train_df.CATEGORY)
Out[67]: {'action',
 'adventure',
 'fighting',
 'misc',
 'platform',
 'puzzle',
 'racing',
 'role-playing',
 'shooter',
 'simulation',
 'sports',
 'strategy'}
```

No modifications were made to PUBLISHER feature

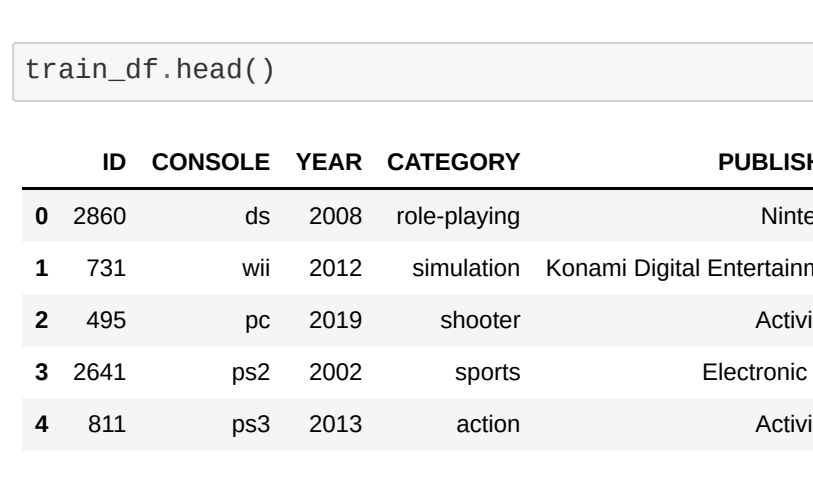
```
In [70]: len(set(train_df.PUBLISHER))
Out[70]: 204
```

```
In [71]: set(train_df.PUBLISHER)
Out[71]: {'1C Company',
 '3Do',
 '505 Games',
 '506 Games',
 '5P',
 'Acclaim Entertainment',
 'Acquire',
 'Activision',
 'Activision Blizzard',
 'Agatsuma Entertainment',
 'Agetec',
 'Aksys Games',
 'Aq Interactive',
 'Aqua Plus',
 'Arc System Works',
 'Ascaron Entertainment Gmbh',
 'Ascii Entertainment',
 'Aspyr',
 'Atari',
 'Atlus',
 'Avalon Interactive',
 'Bam! Entertainment',
 'Banpresto',
 'Bethesda Softworks',
 'Bigben Interactive',
 'Bitcomposer Games',
 'Black Bean Games',
 'Black Label Games',
 'Blue Byte',
 'Capcom',
 'Cave',
 'Cdv Software Entertainment',
 'City Interactive',
 'Cloud Imperium Games Corporation',
 'Codemasters',
 'Compile Heart',
 'Crave Entertainment',
 'Crimson Cow',
 'D3Publisher',
 'Deep Silver',
 'Destination Software, Inc',
 'Destineer',
 'Devolver Digital',
 'Disney Interactive Studios',
 'Dreamcatcher Interactive',
 'Dtp Entertainment',
 'Dusenberry Martin Racing',
 'Ea Games',
 'Eidos Interactive',
 'Electronic Arts',
 'Empire Interactive',
 'En Masse Entertainment',
 'Encore',
 'Enix Corporation',
 'Enterbrain',
 'Evolved Games',
 'Flashpoint Games',
 'Focus Home Interactive',
 'From Software',
 'Funcom',
 'Furyu',
 'Game Factory',
 'Gamebridge',
 'Gamecock',
 'Gaslamp Games',
 'Gathering Of Developers',
 'Genki',
 'Ghostlight',
 'Global Star',
 'Gotham Games',
 'Graffiti',
 'Groove Games',
 'Gt Interactive',
 'Gungho',
 'Harmonix Music Systems',
 'Hasbro Interactive',
 'Havas Interactive',
 'Her Interactive',
 'Hip Interactive',
 'Home Entertainment Suppliers',
 'Hudson Soft',
 'Human Entertainment',
 'Iceberg Interactive',
 'Idea Factory',
 'Ignition Entertainment',
 'Illusion Softworks',
 'Indie Games',
 'Infogrames',
 'Insomniac Games',
 'Interplay',
 'Jaleco',
 'Jowood Productions',
 'Kalypso Media',
 'Kemco',
 'Koch Media',
 'Konami Digital Entertainment',
 'Kool Kizz',
 'Lexicon Entertainment',
 'Little Orbit',
 'Lucasarts',
 'Mad Catz',
 'Majesco Entertainment',
 'Marvelous Entertainment',
 'Marvelous Interactive',
 'Mastertronic',
 'Mastiff',
 'Mc2 Entertainment',
 'Metro 3D',
 'Microids',
 'Microsoft Game Studios',
 'Midas Interactive Entertainment',
 'Midway Games',
 'Milestone S.R.L',
 'Milestone S.R.L.',
 'Mindscape',
 'Monster Games',
 'Moss',
 'Mtv Games',
 'Namco Bandai Games',
 'Naturalmotion',
 'Ncsoft',
 'Nintendo',
 'Nippon Ichi Software',
 'Nobilis',
 'Nordic Games',
 'Novalogic',
 'O-Games',
 'Oovee Game Studios',
 'Pacific Century Cyber Works',
 'Paradox Interactive',
 'Phantagram',
 'Pinnacle',
 'Play It',
 'Playlogic Game Factory',
 'Pm Studios',
 'Popcap Games',
 'Pqube',
 'Rage Software',
 'Rebellion',
 'Rebellion Developments',
 'Redoctane',
 'Revolution Software',
 'Rising Star Games',
 'Rondomedia',
 'Sammy Corporation',
 'Scholastic Inc.',
 'Sci',
 'Screenlife',
 'Sega',
 'Slightly Mad Studios',
 'Snk',
 'Sold Out',
 'Sony Computer Entertainment',
 'Sony Computer Entertainment America',
 'Sony Computer Entertainment Europe',
 'Sony Online Entertainment',
 'Southpeak Games',
 'Spike',
 'Square Ea',
 'Square Enix',
 'Squaresoft',
 'Stainless Games',
 'Sting',
 'Strategy First',
 'Success',
 'Sunsoft',
 'Swing! Entertainment',
 'System 3',
 'System 3 Arcade Software',
 'Takara Tomy',
 'Take-Two Interactive',
 'Talonsoft',
 'Tdk Mediactive',
 'Team17 Software',
 'Tecmo Koei',
 'Telltale Games',
 'Thq',
 'Thq Nordic',
 'Tomy Corporation',
 'Topware Interactive',
 'Touchstone',
 'Trion Worlds',
 'Tripwire Interactive',
 'Tru Blu Entertainment',
 'Ubisoft',
 'Ubisoft Annecy',
 'Universal Interactive',
 'Unknown',
 'Valcon Games',
 'Valve',
 'Valve Software',
 'Virgin Interactive',
 'Visco',
 'Vivendi Games',
 'Warner Bros. Interactive Entertainment',
 'White Park Bay Software',
 'Xicat Interactive',
 'Xplosiv',
 'Xs Games',
 'Xseed Games',
 'Yacht Club Games',
 'Zoo Digital Publishing',
 'Zoo Games',
 'Zushi Games'}
```

```
In [72]: sns.countplot(train_df.CATEGORY)
Out[72]: <matplotlib.axes._subplots.AxesSubplot at 0x7f28b6c73d90>
```




```
In [73]: sns.countplot(test_df.CATEGORY)
Out[73]: <matplotlib.axes._subplots.AxesSubplot at 0x7f28b6bfaf50>
```




```
In [74]: train_df.head()
```

Out[74]:

	ID	CONSOLE	YEAR	CATEGORY	PUBLISHER	RATING	CRITICS_POINTS	USER_POINTS	SalesInMillions
0	2860	ds	2008	role-playing	Nintendo	E	2.833333	0.303704	0.623434
1	731	wii	2012	simulation	Konami Digital Entertainment	E10+	13.200000	1.640000	-1.257900
2	495	pc	2019	shooter	Activision	M	4.562500	0.006410	-0.576469
3	2641	ps2	2002	sports	Electronic Arts	E	4.181818	0.326923	0.339633
4	811	ps3	2013	action	Activision	M	2.259259	0.032579	-1.814317

Both train and test sets contain same PUBLISHER values

```
In [76]: if set(train_df.PUBLISHER.sort_values() == set(test_df.PUBLISHER.sort_values())):

    print(True)

True
```

```
In [77]: len(set(train_df.PUBLISHER.sort_values()))
Out[77]: 204
```

RATING COLUMN

some values present in train set which were absent in test set and hence they were dropped

```
In [81]: set(train_df.RATING.sort_values())
Out[81]: {'E', 'E10+', 'K-A', 'M', 'RP', 'T'}
```

```
In [82]: set(test_df.RATING.sort_values())
Out[82]: {'AO', 'E', 'E10+', 'M', 'T'}
```

Removing rows with "K-A" and "RP"

```
In [83]: train_df[train_df.RATING == "K-A"]
```

Out[83]:

	ID	CONSOLE	YEAR	CATEGORY	PUBLISHER	RATING	CRITICS_POINTS	USER_POINTS	SalesInMillions
286	2771	ps	1998	misc	Sony Computer Entertainment	K-A	6.133333	0.154167	-0.147765

```
In [84]: train_df[train_df.RATING == 'RP']
```

Out[84]:

	ID	CONSOLE	YEAR	CATEGORY	PUBLISHER	RATING	CRITICS_POINTS	USER_POINTS	SalesInMillions
374	3696	pc	2011	strategy	Paradox Interactive	RP	5.25	0.251852	1.121748
748	591	pc	2016	strategy	Gaslamp Games	RP	7.25	0.760000	-0.849052

In [85]: `test_df[test_df.RATING == "AO"]`

Out[85]:

	ID	CONSOLE	YEAR	CATEGORY	PUBLISHER	RATING	CRITICS_POINTS	USER_POINTS
623	1543	x	2008	action	Take-Two Interactive	AO	1.603448	0.073109

In [86]: `train_df = train_df[(train_df.RATING != "RP") & (train_df.RATING != "K-A")]`

In [88]: `set(train_df.RATING.sort_values())`

Out[88]: {'E', 'E10+', 'M', 'T'}

In [89]: `train_df.head()`

Out[89]:

	ID	CONSOLE	YEAR	CATEGORY	PUBLISHER	RATING	CRITICS_POINTS	USER_POINTS	SalesInMillions
0	2860	ds	2008	role-playing	Nintendo	E	2.833333	0.303704	0.623434
1	731	wii	2012	simulation	Konami Digital Entertainment	E10+	13.200000	1.640000	-1.257900
2	495	pc	2019	shooter	Activision	M	4.562500	0.006410	-0.576469
3	2641	ps2	2002	sports	Electronic Arts	E	4.181818	0.326923	0.339633
4	811	ps3	2013	action	Activision	M	2.259259	0.032579	-1.814317

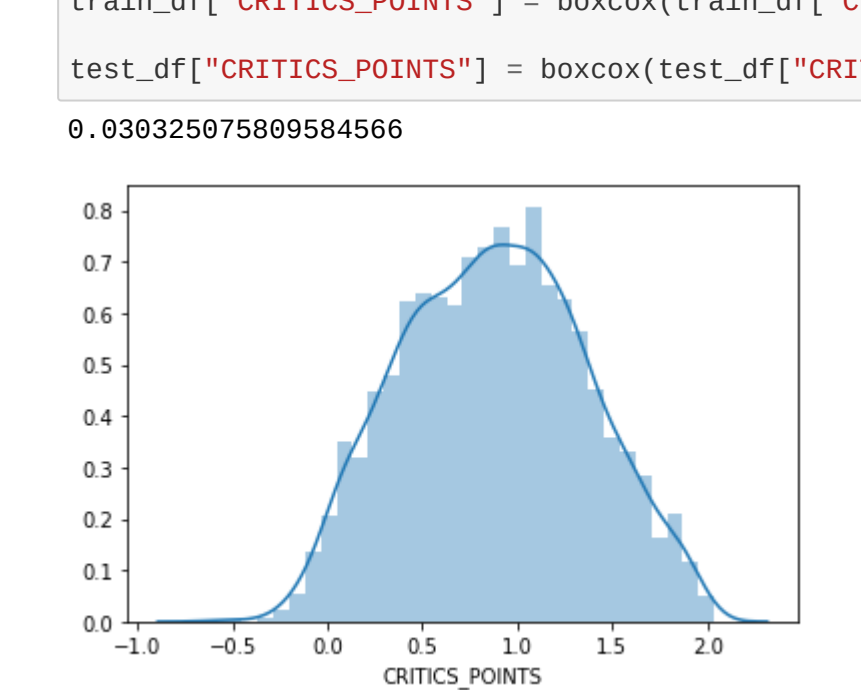
In [90]: `combine.skew()`

Out[90]:
```
ID               -0.025973
YEAR              0.060762
CRITICS_POINTS    2.299211
USER_POINTS       1.607630
dtype: float64
```

Reduced skewness of CRITICS POINTS by applying boxcox transformation

In [91]:
```python
modify_a_col(combine.CRITICS_POINTS,-0.3)

train_df["CRITICS_POINTS"] = boxcox(train_df["CRITICS_POINTS"], -0.3)

test_df["CRITICS_POINTS"] = boxcox(test_df["CRITICS_POINTS"], -0.3)
```

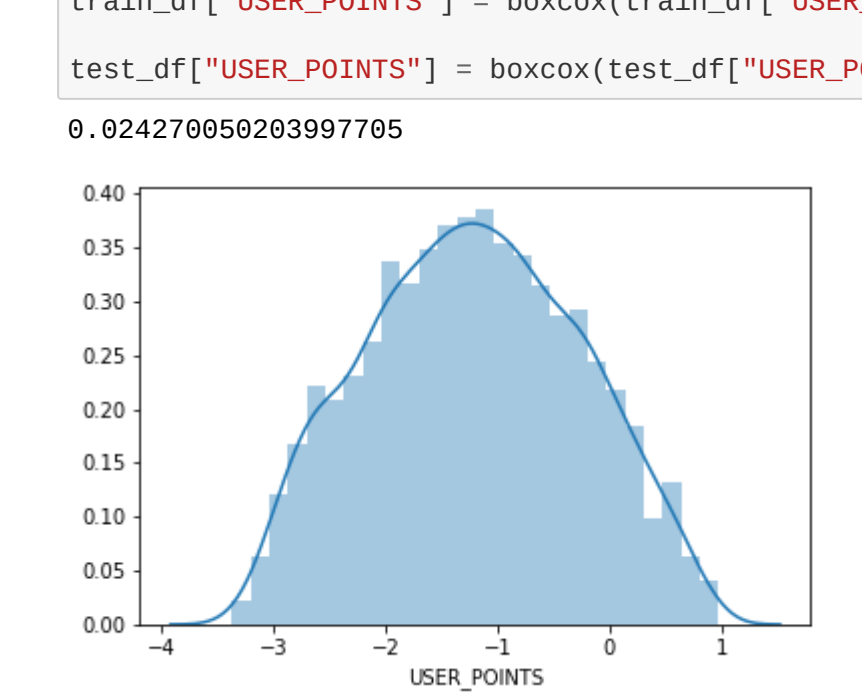
0.030325075809584566



Reduced skewness of USER POINTS by applying boxcox transformation

In [92]:
```python
modify_a_col(combine.USER_POINTS,0.26)

train_df["USER_POINTS"] = boxcox(train_df["USER_POINTS"], 0.26)

test_df["USER_POINTS"] = boxcox(test_df["USER_POINTS"], 0.26)
```

0.024270050203997705



In [93]: `train_df.RATING.unique()`

Out[93]: `array(['E', 'E10+', 'M', 'T'], dtype=object)`

In [94]:
```python
rating_dict = {
    "E" : 1,
    "E10+" : 5,
    "M" : 10,
    "T" : 15
}

# train_df["RATING"] = train_df["RATING"].map(rating_dict)
# train_df["RATING"] = test_df["RATING"].map(rating_dict)
# combine["RATING"] = combine["RATING"].map(rating_dict)
```

I tried Feature Engineering but that decreased model accuracy

In [97]:
```python
# train_df["Skills"] = inv_boxcox(train_df.USER_POINTS,0.26) * train_df.RATING

# test_df["Skills"] = inv_boxcox(test_df.USER_POINTS, 0.26) * test_df.RATING

# test_df["Skills"] = test_df["Skills"].fillna(np.mean(train_df.Skills))

# combine["Skills"] = inv_boxcox(combine.USER_POINTS, 0.26) * combine.RATING

# combine["Skills"] = combine["Skills"].fillna(np.mean(train_df.Skills))

# train_df["Skills"] = boxcox(train_df.Skills, 0.20)
# test_df["Skills"] = boxcox(test_df.Skills, 0.20)
# combine["Skills"] = boxcox(combine.Skills, 0.20)
```

In [98]: `train_df.skew()`

Out[98]:
```
ID               -0.010394
YEAR              0.121266
CRITICS_POINTS    0.018155
USER_POINTS       0.018782
SalesInMillions  -0.001442
dtype: float64
```

In [99]: `# modify_a_col(train_df.Skills, 0.20)`

In [100]: `print(train_df.CONSOLE.unique())`

```
['ps2' 'psp' 'x' '3ds' 'ds' 'pc' 'x360' 'xone' 'wii' 'gc' 'psv' 'gba'
 'ps3' 'wiiu' 'ps4' 'dc']
```

In [101]: `len(train_df.CONSOLE.unique())`

Out[101]: 17

In [102]: `print(train_df.CONSOLE.unique())`

```
['ds' 'wii' 'pc' 'ps2' 'ps3' 'psp' 'x' 'x360' 'psv' 'gba' 'gc' 'ps' 'wiiu'
 'xone' 'ps4' '3ds' 'dc']
```

In [65]:
```python
console_dict = {

    "x" : "x-type",
    "xone" : "x-type",
    "x360" : "x-type",
    "psp" : "p-type",
    "ps2" : "p-type",
    "ps3" : "p-type",
    "ps4" : "p-type",
    "ps" : "p-type",
    "psv" : "p-type",
    "pc" : "pc",
    "wii" : "w-type",
    "wiiu" : "w-type",
    "ds" : "d-type",
    "3ds" : "d-type",
    "dc" : "d-type",
    "gba" : "g-type",
    "gc" : "g-type",
}

print(len(console_dict.keys()))

print(set(console_dict.keys()) == set(train_df.CONSOLE.unique()))

# train_df["CONSOLE"] = train_df.CONSOLE.map(console_dict)
# test_df["CONSOLE"] = test_df.CONSOLE.map(console_dict)
# combine["CONSOLE"] = combine.CONSOLE.map(console_dict)
```

```
17
True
```

In [66]:
```python
# train_df["How_much_Old"] = train_df["YEAR"].apply(lambda x : 2020 - x)
# test_df["How_much_Old"] = test_df["YEAR"].apply(lambda x : 2020 - x)
# combine["How_much_Old"] = combine["YEAR"].apply(lambda x : 2020 - x)

# train_df = train_df.drop("YEAR",axis = 1)
# test_df = test_df.drop("YEAR",axis = 1)
# combine = combine.drop("YEAR",axis = 1)
```

In [103]:
```python
def move_target_to_last(dataset, col_name):

    x = dataset[col_name]

    dataset = dataset.drop(col_name, axis = 1)

    dataset[col_name] = x

    return dataset
```

In [104]: `train_df = move_target_to_last(train_df, "SalesInMillions")`

In [105]: `train_df.head()`

Out[105]:

	ID	CONSOLE	YEAR	CATEGORY	PUBLISHER	RATING	CRITICS_POINTS	USER_POINTS	SalesInMillions
0	2860	ds	2008	role-playing	Nintendo	E	0.894459	-1.024760	0.623434
1	731	wii	2012	simulation	Konami Digital Entertainment	E10+	1.796218	0.527919	-1.257900
2	495	pc	2019	shooter	Activision	M	1.219271	-2.811454	-0.576469
3	2641	ps2	2002	sports	Electronic Arts	E	1.163286	-0.970196	0.339633
4	811	ps3	2013	action	Activision	M	0.723042	-2.267121	-1.814317

In [106]: `test_df.head()`

Out[106]:

	ID	CONSOLE	YEAR	CATEGORY	PUBLISHER	RATING	CRITICS_POINTS	USER_POINTS
0	4310	ps2	2008	action	Tecmo Koei	T	1.122234	-0.664144
1	4011	psp	2007	strategy	Atari	E10+	1.338705	-1.907006
2	2185	ps2	2004	shooter	Electronic Arts	T	0.944129	-1.979223
3	1644	x	2006	action	Electronic Arts	E	0.914674	-0.125732
4	188	3ds	2011	racing	Ubisoft	E10+	0.146945	-1.371757

In [107]: `train_df.corr()`

Out[107]:

	ID	YEAR	CRITICS_POINTS	USER_POINTS	SalesInMillions
ID	1.000000	-0.052969	0.007047	0.061334	-0.032628
YEAR	-0.052969	1.000000	-0.181344	-0.415385	0.004861
CRITICS_POINTS	0.007047	-0.181344	1.000000	0.521520	-0.119885
USER_POINTS	0.061334	-0.415385	0.521520	1.000000	-0.205847
SalesInMillions	-0.032628	0.004861	-0.119885	-0.205847	1.000000

In [108]:
```python
new_train = train_df.copy()

new_test = test_df.copy()

new_train = new_train.drop("CRITICS_POINTS",axis = 1)

new_test = new_test.drop("CRITICS_POINTS", axis = 1)
```

In [109]: `train_df.head()`

Out[109]:

	ID	CONSOLE	YEAR	CATEGORY	PUBLISHER	RATING	CRITICS_POINTS	USER_POINTS	SalesInMillions
0	2860	ds	2008	role-playing	Nintendo	E	0.894459	-1.024760	0.623434
1	731	wii	2012	simulation	Konami Digital Entertainment	E10+	1.796218	0.527919	-1.257900
2	495	pc	2019	shooter	Activision	M	1.219271	-2.811454	-0.576469
3	2641	ps2	2002	sports	Electronic Arts	E	1.163286	-0.970196	0.339633
4	811	ps3	2013	action	Activision	M	0.723042	-2.267121	-1.814317

In [110]:
```python
class Jiten(object):
    def get_final_error(self, error, weight):
        return np.sqrt(boxcox(error,0.27) / (weight + 1e-38))

    def is_max_optimal(self):
        return False

    def evaluate(self, approxes, target, weight):
        assert len(approxes) == 1
        assert len(target) == len(approxes[0])

        approx = approxes[0]

        error_sum = 0.0
        weight_sum = 0.0

        for i in range(len(approx)):
            w = 1.0 if weight is None else weight[i]
            weight_sum += w
            error_sum += inv_boxcox(w,0.27) * ((inv_boxcox(approx[i],0.27) - inv_boxcox(target[i],0.
27))**2)

        return error_sum, weight_sum
```

Since we have transformed output variable we need to change eval metric so that catboost model performance increase

Model score increased from 1.66 to 1.62 without any tuning

In [111]:
```python
# class RmseObjective(object):
#     def calc_ders_range(self, approxes, targets, weights):
#         assert len(approxes) == len(targets)
#         if weights is not None:
#             assert len(weights) == len(approxes)

#         targets = inv_boxcox(targets, 0.27)

#         approxes = inv_boxcox(approxes, 0.27)

#         result = []

#         for index in range(len(targets)):
#             der1 = targets[index] - approxes[index]
#             der2 = -1

#             if weights is not None:
#                 der1 *= weights[index]
#                 der2 *= weights[index]

#             result.append((der1, der2))
#         return result
```

Getting Validation Test

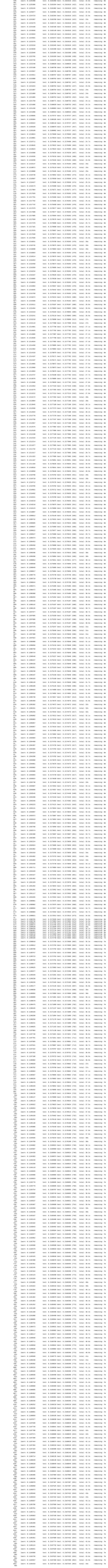
In [112]:
```python
X_train, X_test, y_train, y_test = train_test_split(train_df.drop(["SalesInMillions"],axis = 1), tra
in_df["SalesInMillions"], test_size = 0.2,random_state=0)
```

In [113]:
```python
cat_columns=train_df.select_dtypes(include=['object','category']).columns.tolist()

cat_columns
```

Out[113]: ['CONSOLE', 'CATEGORY', 'PUBLISHER', 'RATING']

Model Training

In [114]:
```python
model = CatBoostRegressor(use_best_model=True,learning_rate=0.01,n_estimators=10000,objective="RMSE"
,cat_features=cat_columns,eval_metric=Jiten())
```



```
8105:	learn: 0.0923556	test: 0.2147572	best: 0.2145365 (6667)	total: 8m 25s	remaining: 1m 58s
8106:	learn: 0.0923526	test: 0.2147571	best: 0.2145365 (6667)	total: 8m 25s	remaining: 1m 58s
8107:	learn: 0.0923505	test: 0.2147572	best: 0.2145365 (6667)	total: 8m 25s	remaining: 1m 58s
8108:	learn: 0.0923468	test: 0.2147586	best: 0.2145365 (6667)	total: 8m 25s	remaining: 1m 57s
8109:	learn: 0.0923463	test: 0.2147586	best: 0.2145365 (6667)	total: 8m 25s	remaining: 1m 57s
8110:	learn: 0.0923450	test: 0.2147605	best: 0.2145365 (6667)	total: 8m 26s	remaining: 1m 57s
8111:	learn: 0.0923433	test: 0.2147639	best: 0.2145365 (6667)	total: 8m 26s	remaining: 1m 57s
8112:	learn: 0.0923420	test: 0.2147636	best: 0.2145365 (6667)	total: 8m 26s	remaining: 1m 57s
8113:	learn: 0.0923376	test: 0.2147638	best: 0.2145365 (6667)	total: 8m 26s	remaining: 1m 57s
8114:	learn: 0.0923374	test: 0.2147642	best: 0.2145365 (6667)	total: 8m 26s	remaining: 1m 57s
8115:	learn: 0.0923361	test: 0.2147651	best: 0.2145365 (6667)	total: 8m 26s	remaining: 1m 57s
8116:	learn: 0.0923346	test: 0.2147702	best: 0.2145365 (6667)	total: 8m 26s	remaining: 1m 57s
8117:	learn: 0.0923322	test: 0.2147690	best: 0.2145365 (6667)	total: 8m 26s	remaining: 1m 57s
8118:	learn: 0.0923309	test: 0.2147679	best: 0.2145365 (6667)	total: 8m 26s	remaining: 1m 57s
8119:	learn: 0.0923309	test: 0.2147679	best: 0.2145365 (6667)	total: 8m 26s	remaining: 1m 57s
8120:	learn: 0.0923297	test: 0.2147690	best: 0.2145365 (6667)	total: 8m 26s	remaining: 1m 57s
8121:	learn: 0.0923284	test: 0.2147689	best: 0.2145365 (6667)	total: 8m 26s	remaining: 1m 57s
8122:	learn: 0.0923280	test: 0.2147679	best: 0.2145365 (6667)	total: 8m 26s	remaining: 1m 57s
8123:	learn: 0.0923243	test: 0.2147684	best: 0.2145365 (6667)	total: 8m 26s	remaining: 1m 57s
8124:	learn: 0.0923216	test: 0.2147658	best: 0.2145365 (6667)	total: 8m 26s	remaining: 1m 56s
8125:	learn: 0.0923192	test: 0.2147656	best: 0.2145365 (6667)	total: 8m 27s	remaining: 1m 56s
8126:	learn: 0.0923132	test: 0.2147653	best: 0.2145365 (6667)	total: 8m 27s	remaining: 1m 56s
8127:	learn: 0.0923124	test: 0.2147658	best: 0.2145365 (6667)	total: 8m 27s	remaining: 1m 56s
8128:	learn: 0.0923118	test: 0.2147655	best: 0.2145365 (6667)	total: 8m 27s	remaining: 1m 56s
8129:	learn: 0.0923095	test: 0.2147669	best: 0.2145365 (6667)	total: 8m 27s	remaining: 1m 56s
8130:	learn: 0.0923083	test: 0.2147648	best: 0.2145365 (6667)	total: 8m 27s	remaining: 1m 56s
8131:	learn: 0.0923085	test: 0.2147650	best: 0.2145365 (6667)	total: 8m 27s	remaining: 1m 56s
8132:	learn: 0.0923081	test: 0.2147654	best: 0.2145365 (6667)	total: 8m 27s	remaining: 1m 56s
8133:	learn: 0.0923061	test: 0.2147654	best: 0.2145365 (6667)	total: 8m 27s	remaining: 1m 56s
8134:	learn: 0.0923059	test: 0.2147653	best: 0.2145365 (6667)	total: 8m 27s	remaining: 1m 56s
8135:	learn: 0.0923060	test: 0.2147653	best: 0.2145365 (6667)	total: 8m 27s	remaining: 1m 56s
8136:	learn: 0.0923031	test: 0.2147659	best: 0.2145365 (6667)	total: 8m 27s	remaining: 1m 56s
8137:	learn: 0.0923002	test: 0.2147657	best: 0.2145365 (6667)	total: 8m 27s	remaining: 1m 56s
8138:	learn: 0.0922996	test: 0.2147653	best: 0.2145365 (6667)	total: 8m 27s	remaining: 1m 56s
8139:	learn: 0.0922963	test: 0.2147646	best: 0.2145365 (6667)	total: 8m 28s	remaining: 1m 56s
8140:	learn: 0.0922957	test: 0.2147660	best: 0.2145365 (6667)	total: 8m 28s	remaining: 1m 56s
8141:	learn: 0.0922913	test: 0.2147672	best: 0.2145365 (6667)	total: 8m 28s	remaining: 1m 55s
8142:	learn: 0.0922895	test: 0.2147675	best: 0.2145365 (6667)	total: 8m 28s	remaining: 1m 55s
8143:	learn: 0.0922879	test: 0.2147672	best: 0.2145365 (6667)	total: 8m 28s	remaining: 1m 55s
8144:	learn: 0.0922860	test: 0.2147698	best: 0.2145365 (6667)	total: 8m 28s	remaining: 1m 55s
8145:	learn: 0.0922832	test: 0.2147750	best: 0.2145365 (6667)	total: 8m 28s	remaining: 1m 55s
8146:	learn: 0.0922819	test: 0.2147747	best: 0.2145365 (6667)	total: 8m 28s	remaining: 1m 55s
8147:	learn: 0.0922785	test: 0.2147740	best: 0.2145365 (6667)	total: 8m 28s	remaining: 1m 55s
8148:	learn: 0.0922761	test: 0.2147754	best: 0.2145365 (6667)	total: 8m 28s	remaining: 1m 55s
8149:	learn: 0.0922722	test: 0.2147757	best: 0.2145365 (6667)	total: 8m 29s	remaining: 1m 55s
8150:	learn: 0.0922720	test: 0.2147759	best: 0.2145365 (6667)	total: 8m 29s	remaining: 1m 55s
8151:	learn: 0.0922700	test: 0.2147764	best: 0.2145365 (6667)	total: 8m 29s	remaining: 1m 55s
8152:	learn: 0.0922698	test: 0.2147773	best: 0.2145365 (6667)	total: 8m 29s	remaining: 1m 55s
8153:	learn: 0.0922668	test: 0.2147772	best: 0.2145365 (6667)	total: 8m 29s	remaining: 1m 55s
8154:	learn: 0.0922667	test: 0.2147773	best: 0.2145365 (6667)	total: 8m 29s	remaining: 1m 55s
8155:	learn: 0.0922660	test: 0.2147773	best: 0.2145365 (6667)	total: 8m 29s	remaining: 1m 55s
8156:	learn: 0.0922638	test: 0.2147788	best: 0.2145365 (6667)	total: 8m 29s	remaining: 1m 55s
8157:	learn: 0.0922628	test: 0.2147803	best: 0.2145365 (6667)	total: 8m 29s	remaining: 1m 55s
8158:	learn: 0.0922583	test: 0.2147823	best: 0.2145365 (6667)	total: 8m 29s	remaining: 1m 54s
8159:	learn: 0.0922540	test: 0.2147814	best: 0.2145365 (6667)	total: 8m 29s	remaining: 1m 54s
8160:	learn: 0.0922524	test: 0.2147816	best: 0.2145365 (6667)	total: 8m 29s	remaining: 1m 54s
8161:	learn: 0.0922507	test: 0.2147827	best: 0.2145365 (6667)	total: 8m 29s	remaining: 1m 54s
8162:	learn: 0.0922494	test: 0.2147839	best: 0.2145365 (6667)	total: 8m 29s	remaining: 1m 54s
8163:	learn: 0.0922466	test: 0.2147831	best: 0.2145365 (6667)	total: 8m 29s	remaining: 1m 54s
8164:	learn: 0.0922444	test: 0.2147822	best: 0.2145365 (6667)	total: 8m 29s	remaining: 1m 54s
8165:	learn: 0.0922396	test: 0.2147823	best: 0.2145365 (6667)	total: 8m 29s	remaining: 1m 54s
8166:	learn: 0.0922340	test: 0.2147779	best: 0.2145365 (6667)	total: 8m 30s	remaining: 1m 54s
8167:	learn: 0.0922326	test: 0.2147779	best: 0.2145365 (6667)	total: 8m 30s	remaining: 1m 54s
8168:	learn: 0.0922283	test: 0.2147764	best: 0.2145365 (6667)	total: 8m 30s	remaining: 1m 54s
8169:	learn: 0.0922242	test: 0.2147750	best: 0.2145365 (6667)	total: 8m 30s	remaining: 1m 54s
8170:	learn: 0.0922225	test: 0.2147765	best: 0.2145365 (6667)	total: 8m 30s	remaining: 1m 54s
8171:	learn: 0.0922187	test: 0.2147742	best: 0.2145365 (6667)	total: 8m 30s	remaining: 1m 54s
8172:	learn: 0.0922161	test: 0.2147738	best: 0.2145365 (6667)	total: 8m 30s	remaining: 1m 54s
8173:	learn: 0.0922109	test: 0.2147766	best: 0.2145365 (6667)	total: 8m 30s	remaining: 1m 54s
8174:	learn: 0.0922082	test: 0.2147754	best: 0.2145365 (6667)	total: 8m 30s	remaining: 1m 53s
8175:	learn: 0.0922034	test: 0.2147736	best: 0.2145365 (6667)	total: 8m 30s	remaining: 1m 53s
8176:	learn: 0.0922022	test: 0.2147741	best: 0.2145365 (6667)	total: 8m 30s	remaining: 1m 53s
8177:	learn: 0.0922015	test: 0.2147733	best: 0.2145365 (6667)	total: 8m 30s	remaining: 1m 53s
8178:	learn: 0.0922005	test: 0.2147737	best: 0.2145365 (6667)	total: 8m 30s	remaining: 1m 53s
8179:	learn: 0.0922006	test: 0.2147737	best: 0.2145365 (6667)	total: 8m 30s	remaining: 1m 53s
8180:	learn: 0.0921981	test: 0.2147743	best: 0.2145365 (6667)	total: 8m 30s	remaining: 1m 53s
8181:	learn: 0.0921957	test: 0.2147751	best: 0.2145365 (6667)	total: 8m 31s	remaining: 1m 53s
8182:	learn: 0.0921939	test: 0.2147773	best: 0.2145365 (6667)	total: 8m 31s	remaining: 1m 53s
8183:	learn: 0.0921905	test: 0.2147786	best: 0.2145365 (6667)	total: 8m 31s	remaining: 1m 53s
8184:	learn: 0.0921895	test: 0.2147783	best: 0.2145365 (6667)	total: 8m 31s	remaining: 1m 53s
8185:	learn: 0.0921868	test: 0.2147784	best: 0.2145365 (6667)	total: 8m 31s	remaining: 1m 53s
8186:	learn: 0.0921853	test: 0.2147784	best: 0.2145365 (6667)	total: 8m 31s	remaining: 1m 53s
8187:	learn: 0.0921822	test: 0.2147789	best: 0.2145365 (6667)	total: 8m 31s	remaining: 1m 53s
8188:	learn: 0.0921798	test: 0.2147792	best: 0.2145365 (6667)	total: 8m 31s	remaining: 1m 53s
8189:	learn: 0.0921787	test: 0.2147804	best: 0.2145365 (6667)	total: 8m 31s	remaining: 1m 53s
8190:	learn: 0.0921771	test: 0.2147811	best: 0.2145365 (6667)	total: 8m 31s	remaining: 1m 53s
8191:	learn: 0.0921748	test: 0.2147800	best: 0.2145365 (6667)	total: 8m 32s	remaining: 1m 53s
8192:	learn: 0.0921747	test: 0.2147801	best: 0.2145365 (6667)	total: 8m 32s	remaining: 1m 52s
8193:	learn: 0.0921719	test: 0.2147808	best: 0.2145365 (6667)	total: 8m 32s	remaining: 1m 52s
8194:	learn: 0.0921684	test: 0.2147792	best: 0.2145365 (6667)	total: 8m 32s	remaining: 1m 52s
8195:	learn: 0.0921654	test: 0.2147805	best: 0.2145365 (6667)	total: 8m 32s	remaining: 1m 52s
8196:	learn: 0.0921644	test: 0.2147802	best: 0.2145365 (6667)	total: 8m 32s	remaining: 1m 52s
8197:	learn: 0.0921637	test: 0.2147789	best: 0.2145365 (6667)	total: 8m 32s	remaining: 1m 52s
8198:	learn: 0.0921620	test: 0.2147786	best: 0.2145365 (6667)	total: 8m 32s	remaining: 1m 52s
8199:	learn: 0.0921603	test: 0.2147781	best: 0.2145365 (6667)	total: 8m 32s	remaining: 1m 52s
8200:	learn: 0.0921566	test: 0.2147805	best: 0.2145365 (6667)	total: 8m 32s	remaining: 1m 52s
8201:	learn: 0.0921565	test: 0.2147808	best: 0.2145365 (6667)	total: 8m 32s	remaining: 1m 52s
8202:	learn: 0.0921562	test: 0.2147815	best: 0.2145365 (6667)	total: 8m 32s	remaining: 1m 52s
8203:	learn: 0.0921520	test: 0.2147860	best: 0.2145365 (6667)	total: 8m 32s	remaining: 1m 52s
8204:	learn: 0.0921515	test: 0.2147867	best: 0.2145365 (6667)	total: 8m 32s	remaining: 1m 52s
8205:	learn: 0.0921495	test: 0.2147905	best: 0.2145365 (6667)	total: 8m 32s	remaining: 1m 52s
8206:	learn: 0.0921441	test: 0.2147916	best: 0.2145365 (6667)	total: 8m 32s	remaining: 1m 52s
8207:	learn: 0.0921417	test: 0.2147907	best: 0.2145365 (6667)	total: 8m 32s	remaining: 1m 51s
8208:	learn: 0.0921390	test: 0.2147917	best: 0.2145365 (6667)	total: 8m 33s	remaining: 1m 51s
8209:	learn: 0.0921376	test: 0.2147909	best: 0.2145365 (6667)	total: 8m 33s	remaining: 1m 51s
8210:	learn: 0.0921367	test: 0.2147915	best: 0.2145365 (6667)	total: 8m 33s	remaining: 1m 51s
8211:	learn: 0.0921348	test: 0.2147911	best: 0.2145365 (6667)	total: 8m 33s	remaining: 1m 51s
8212:	learn: 0.0921305	test: 0.2147883	best: 0.2145365 (6667)	total: 8m 33s	remaining: 1m 51s
8213:	learn: 0.0921272	test: 0.2147884	best: 0.2145365 (6667)	total: 8m 33s	remaining: 1m 51s
8214:	learn: 0.0921239	test: 0.2147883	best: 0.2145365 (6667)	total: 8m 33s	remaining: 1m 51s
8215:	learn: 0.0921210	test: 0.2147878	best: 0.2145365 (6667)	total: 8m 33s	remaining: 1m 51s
8216:	learn: 0.0921204	test: 0.2147881	best: 0.2145365 (6667)	total: 8m 33s	remaining: 1m 51s
8217:	learn: 0.0921174	test: 0.2147881	best: 0.2145365 (6667)	total: 8m 33s	remaining: 1m 51s
8218:	learn: 0.0921134	test: 0.2147925	best: 0.2145365 (6667)	total: 8m 33s	remaining: 1m 51s
8219:	learn: 0.0921101	test: 0.2147917	best: 0.2145365 (6667)	total: 8m 34s	remaining: 1m 51s
8220:	learn: 0.0921091	test: 0.2147917	best: 0.2145365 (6667)	total: 8m 34s	remaining: 1m 51s
8221:	learn: 0.0921055	test: 0.2147929	best: 0.2145365 (6667)	total: 8m 34s	remaining: 1m 51s
8222:	learn: 0.0921044	test: 0.2147867	best: 0.2145365 (6667)	total: 8m 34s	remaining: 1m 51s
8223:	learn: 0.0921025	test: 0.2147890	best: 0.2145365 (6667)	total: 8m 34s	remaining: 1m 51s
8224:	learn: 0.0921000	test: 0.2147877	best: 0.2145365 (6667)	total: 8m 34s	remaining: 1m 50s
8225:	learn: 0.0920980	test: 0.2147879	best: 0.2145365 (6667)	total: 8m 34s	remaining: 1m 50s
8226:	learn: 0.0920954	test: 0.2147873	best: 0.2145365 (6667)	total: 8m 34s	remaining: 1m 50s
8227:	learn: 0.0920931	test: 0.2147911	best: 0.2145365 (6667)	total: 8m 34s	remaining: 1m 50s
8228:	learn: 0.0920903	test: 0.2147912	best: 0.2145365 (6667)	total: 8m 34s	remaining: 1m 50s
8229:	learn: 0.0920887	test: 0.2147923	best: 0.2145365 (6667)	total: 8m 34s	remaining: 1m 50s
8230:	learn: 0.0920813	test: 0.2147934	best: 0.2145365 (6667)	total: 8m 34s	remaining: 1m 50s
8231:	learn: 0.0920775	test: 0.2147941	best: 0.2145365 (6667)	total: 8m 35s	remaining: 1m 50s
8232:	learn: 0.0920750	test: 0.2147899	best: 0.2145365 (6667)	total: 8m 35s	remaining: 1m 50s
8233:	learn: 0.0920745	test: 0.2147918	best: 0.2145365 (6667)	total: 8m 35s	remaining: 1m 50s
8234:	learn: 0.0920734	test: 0.2147913	best: 0.2145365 (6667)	total: 8m 35s	remaining: 1m 50s
8235:	learn: 0.0920694	test: 0.2147893	best: 0.2145365 (6667)	total: 8m 35s	remaining: 1m 50s
8236:	learn: 0.0920668	test: 0.2147898	best: 0.2145365 (6667)	total: 8m 35s	remaining: 1m 50s
8237:	learn: 0.0920645	test: 0.2147903	best: 0.2145365 (6667)	total: 8m 35s	remaining: 1m 50s
8238:	learn: 0.0920623	test: 0.2147906	best: 0.2145365 (6667)	total: 8m 35s	remaining: 1m 50s
8239:	learn: 0.0920605	test: 0.2147937	best: 0.2145365 (6667)	total: 8m 35s	remaining: 1m 50s
8240:	learn: 0.0920591	test: 0.2147940	best: 0.2145365 (6667)	total: 8m 35s	remaining: 1m 50s
8241:	learn: 0.0920567	test: 0.2147937	best: 0.2145365 (6667)	total: 8m 35s	remaining: 1m 49s
8242:	learn: 0.0920555	test: 0.2147972	best: 0.2145365 (6667)	total: 8m 35s	remaining: 1m 49s
8243:	learn: 0.0920528	test: 0.2147969	best: 0.2145365 (6667)	total: 8m 35s	remaining: 1m 49s
8244:	learn: 0.0920515	test: 0.2147979	best: 0.2145365 (6667)	total: 8m 35s	remaining: 1m 49s
8245:	learn: 0.0920502	test: 0.2147974	best: 0.2145365 (6667)	total: 8m 35s	remaining: 1m 49s
8246:	learn: 0.0920482	test: 0.2147979	best: 0.2145365 (6667)	total: 8m 35s	remaining: 1m 49s
8247:	learn: 0.0920464	test: 0.2147992	best: 0.2145365 (6667)	total: 8m 36s	remaining: 1m 49s
8248:	learn: 0.0920444	test: 0.2148001	best: 0.2145365 (6667)	total: 8m 36s	remaining: 1m 49s
8249:	learn: 0.0920405	test: 0.2148000	best: 0.2145365 (6667)	total: 8m 36s	remaining: 1m 49s
8250:	learn: 0.0920397	test: 0.2148003	best: 0.2145365 (6667)	total: 8m 36s	remaining: 1m 49s
8251:	learn: 0.0920358	test: 0.2148029	best: 0.2145365 (6667)	total: 8m 36s	remaining: 1m 49s
8252:	learn: 0.0920349	test: 0.2148029	best: 0.2145365 (6667)	total: 8m 36s	remaining: 1m 49s
8253:	learn: 0.0920316	test: 0.2148031	best: 0.2145365 (6667)	total: 8m 36s	remaining: 1m 49s
8254:	learn: 0.0920304	test: 0.2147999	best: 0.2145365 (6667)	total: 8m 36s	remaining: 1m 49s
8255:	learn: 0.0920297	test: 0.2148001	best: 0.2145365 (6667)	total: 8m 36s	remaining: 1m 49s
8256:	learn: 0.0920266	test: 0.2148006	best: 0.2145365 (6667)	total: 8m 36s	remaining: 1m 49s
8257:	learn: 0.0920253	test: 0.2148000	best: 0.2145365 (6667)	total: 8m 36s	remaining: 1m 49s
8258:	learn: 0.0920224	test: 0.2148014	best: 0.2145365 (6667)	total: 8m 36s	remaining: 1m 48s
8259:	learn: 0.0920221	test: 0.2148014	best: 0.2145365 (6667)	total: 8m 36s	remaining: 1m 48s
8260:	learn: 0.0920213	test: 0.2148005	best: 0.2145365 (6667)	total: 8m 37s	remaining: 1m 48s
8261:	learn: 0.0920163	test: 0.2147993	best: 0.2145365 (6667)	total: 8m 37s	remaining: 1m 48s
8262:	learn: 0.0920155	test: 0.2147995	best: 0.2145365 (6667)	total: 8m 37s	remaining: 1m 48s
8263:	learn: 0.0920137	test: 0.2147996	best: 0.2145365 (6667)	total: 8m 37s	remaining: 1m 48s
8264:	learn: 0.0920091	test: 0.2147988	best: 0.2145365 (6667)	total: 8m 37s	remaining: 1m 48s
8265:	learn: 0.0920077	test: 0.2148012	best: 0.2145365 (6667)	total: 8m 37s	remaining: 1m 48s
8266:	learn: 0.0920052	test: 0.2147996	best: 0.2145365 (6667)	total: 8m 37s	remaining: 1m 48s
8267:	learn: 0.0920048	test: 0.2147997	best: 0.2145365 (6667)	total: 8m 37s	remaining: 1m 48s
8268:	learn: 0.0920039	test: 0.2148014	best: 0.2145365 (6667)	total: 8m 38s	remaining: 1m 48s
8269:	learn: 0.0920011	test: 0.2148004	best: 0.2145365 (6667)	total: 8m 38s	remaining: 1m 48s
8270:	learn: 0.0919977	test: 0.2148020	best: 0.2145365 (6667)	total: 8m 38s	remaining: 1m 48s
8271:	learn: 0.0919954	test: 0.2148024	best: 0.2145365 (6667)	total: 8m 38s	remaining: 1m 48s
8272:	learn: 0.0919919	test: 0.2148019	best: 0.2145365 (6667)	total: 8m 38s	remaining: 1m 48s
8273:	learn: 0.0919909	test: 0.2148015	best: 0.2145365 (6667)	total: 8m 38s	remaining: 1m 48s
8274:	learn: 0.0919881	test: 0.2148019	best: 0.2145365 (6667)	total: 8m 38s	remaining: 1m 48s
8275:	learn: 0.0919852	test: 0.2147989	best: 0.2145365 (6667)	total: 8m 38s	remaining: 1m 47s
8276:	learn: 0.0919847	test: 0.2147985	best: 0.2145365 (6667)	total: 8m 38s	remaining: 1m 47s
8277:	learn: 0.0919827	test: 0.2147984	best: 0.2145365 (6667)	total: 8m 38s	remaining: 1m 47s
8278:	learn: 0.0919805	test: 0.2147972	best: 0.2145365 (6667)	total: 8m 38s	remaining: 1m 47s
8279:	learn: 0.0919790	test: 0.2147970	best: 0.2145365 (6667)	total: 8m 38s	remaining: 1m 47s
8280:	learn: 0.0919740	test: 0.2147970	best: 0.2145365 (6667)	total: 8m 38s	remaining: 1m 47s
8281:	learn: 0.0919733	test: 0.2147980	best: 0.2145365 (6667)	total: 8m 38s	remaining: 1m 47s
8282:	learn: 0.0919698	test: 0.2147937	best: 0.2145365 (6667)	total: 8m 38s	remaining: 1m 47s
8283:	learn: 0.0919649	test: 0.2147947	best: 0.2145365 (6667)	total: 8m 38s	remaining: 1m 47s
8284:	learn: 0.0919606	test: 0.2147956	best: 0.2145365 (6667)	total: 8m 38s	remaining: 1m 47s
8285:	learn: 0.0919560	test: 0.2147958	best: 0.2145365 (6667)	total: 8m 38s	remaining: 1m 47s
8286:	learn: 0.0919555	test: 0.2147949	best: 0.2145365 (6667)	total: 8m 39s	remaining: 1m 47s
8287:	learn: 0.0919547	test: 0.2147950	best: 0.2145365 (6667)	total: 8m 39s	remaining: 1m 47s
8288:	learn: 0.0919532	test: 0.2147945	best: 0.2145365 (6667)	total: 8m 39s	remaining: 1m 47s
8289:	learn: 0.0919519	test: 0.2147941	best: 0.2145365 (6667)	total: 8m 39s	remaining: 1m 47s
8290:	learn: 0.0919491	test: 0.2147942	best: 0.2145365 (6667)	total: 8m 39s	remaining: 1m 47s
8291:	learn: 0.0919452	test: 0.2147924	best: 0.2145365 (6667)	total: 8m 39s	remaining: 1m 47s
8292:	learn: 0.0919421	test: 0.2147898	best: 0.2145365 (6667)	total: 8m 39s	remaining: 1m 47s
8293:	learn: 0.0919413	test: 0.2147912	best: 0.2145365 (6667)	total: 8m 39s	remaining: 1m 46s
8294:	learn: 0.0919396	test: 0.2147908	best: 0.2145365 (6667)	total: 8m 40s	remaining: 1m 46s
8295:	learn: 0.0919389	test: 0.2147910	best: 0.2145365 (6667)	total: 8m 40s	remaining: 1m 46s
8296:	learn: 0.0919383	test: 0.2147919	best: 0.2145365 (6667)	total: 8m 40s	remaining: 1m 46s
8297:	learn: 0.0919368	test: 0.2147887	best: 0.2145365 (6667)	total: 8m 40s	remaining: 1m 46s
8298:	learn: 0.0919354	test: 0.2147871	best: 0.2145365 (6667)	total: 8m 40s	remaining: 1m 46s
8299:	learn: 0.0919269	test: 0.2147829	best: 0.2145365 (6667)	total: 8m 40s	remaining: 1m 46s
8300:	learn: 0.0919223	test: 0.2147846	best: 0.2145365 (6667)	total: 8m 40s	remaining: 1m 46s
8301:	learn: 0.0919206	test: 0.2147841	best: 0.2145365 (6667)	total: 8m 40s	remaining: 1m 46s
8302:	learn: 0.0919196	test: 0.2147843	best: 0.2145365 (6667)	total: 8m 40s	remaining: 1m 46s
8303:	learn: 0.0919168	test: 0.2147848	best: 0.2145365 (6667)	total: 8m 40s	remaining: 1m 46s
8304:	learn: 0.0919150	test: 0.2147858	best: 0.2145365 (6667)	total: 8m 40s	remaining: 1m 46s
8305:	learn: 0.0919150	test: 0.2147857	best: 0.2145365 (6667)	total: 8m 40s	remaining: 1m 46s
8306:	learn: 0.0919133	test: 0.2147854	best: 0.2145365 (6667)	total: 8m 40s	remaining: 1m 46s
8307:	learn: 0.0919126	test: 0.2147858	best: 0.2145365 (6667)	total: 8m 41s	remaining: 1m 46s
8308:	learn: 0.0919113	test: 0.2147859	best: 0.2145365 (6667)	total: 8m 41s	remaining: 1m 46s
8309:	learn: 0.0919098	test: 0.2147812	best: 0.2145365 (6667)	total: 8m 41s	remaining: 1m 46s
8310:	learn: 0.0919069	test: 0.2147813	best: 0.2145365 (6667)	total: 8m 41s	remaining: 1m 45s
8311:	learn: 0.0919022	test: 0.2147825	best: 0.2145365 (6667)	total: 8m 41s	remaining: 1m 45s
8312:	learn: 0.0918980	test: 0.2147807	best: 0.2145365 (6667)	total: 8m 41s	remaining: 1m 45s
8313:	learn: 0.0918949	test: 0.2147815	best: 0.2145365 (6667)	total: 8m 41s	remaining: 1m 45s
8314:	learn: 0.0918929	test: 0.2147822	best: 0.2145365 (6667)	total: 8m 41s	remaining: 1m 45s
8315:	learn: 0.0918888	test: 0.2147841	best: 0.2145365 (6667)	total: 8m 41s	remaining: 1m 45s
8316:	learn: 0.0918864	test: 0.2147855	best: 0.2145365 (6667)	total: 8m 41s	remaining: 1m 45s
8317:	learn: 0.0918829	test: 0.2147852	best: 0.2145365 (6667)	total: 8m 41s	remaining: 1m 45s
8318:	learn: 0.0918818	test: 0.2147847	best: 0.2145365 (6667)	total: 8m 41s	remaining: 1m 45s
8319:	learn: 0.0918797	test: 0.2147846	best: 0.2145365 (6667)	total: 8m 41s	remaining: 1m 45s
8320:	learn: 0.0918782	test: 0.2147842	best: 0.2145365 (6667)	total: 8m 41s	remaining: 1m 45s
8321:	learn: 0.0918767	test: 0.2147848	best: 0.2145365 (6667)	total: 8m 41s	remaining: 1m 45s
8322:	learn: 0.0918754	test: 0.2147870	best: 0.2145365 (6667)	total: 8m 41s	remaining: 1m 45s
8323:	learn: 0.0918743	test: 0.2147876	best: 0.2145365 (6667)	total: 8m 42s	remaining: 1m 45s
8324:	learn: 0.0918718	test: 0.2147877	best: 0.2145365 (6667)	total: 8m 42s	remaining: 1m 45s
8325:	learn: 0.0918708	test: 0.2147875	best: 0.2145365 (6667)	total: 8m 42s	remaining: 1m 45s
8326:	learn: 0.0918679	test: 0.2147913	best: 0.2145365 (6667)	total: 8m 42s	remaining: 1m 45s
8327:	learn: 0.0918678	test: 0.2147959	best: 0.2145365 (6667)	total: 8m 42s	remaining: 1m 44s
8328:	learn: 0.0918647	test: 0.2147952	best: 0.2145365 (6667)	total: 8m 42s	remaining: 1m 44s
8329:	learn: 0.0918632	test: 0.2147940	best: 0.2145365 (6667)	total: 8m 42s	remaining: 1m 44s
8330:	learn: 0.0918614	test: 0.2147951	best: 0.2145365 (6667)	total: 8m 42s	remaining: 1m 44s
8331:	learn: 0.0918578	test: 0.2147953	best: 0.2145365 (6667)	total: 8m 42s	remaining: 1m 44s
8332:	learn: 0.0918557	test: 0.2147951	best: 0.2145365 (6667)	total: 8m 42s	remaining: 1m 44s
8333:	learn: 0.0918540	test: 0.2147948	best: 0.2145365 (6667)	total: 8m 43s	remaining: 1m 44s
8334:	learn: 0.0918502	test: 0.2147952	best: 0.2145365 (6667)	total: 8m 43s	remaining: 1m 44s
8335:	learn: 0.0918487	test: 0.2147946	best: 0.2145365 (6667)	total: 8m 43s	remaining: 1m 44s
8336:	learn: 0.0918483	test: 0.2147950	best: 0.2145365 (6667)	total: 8m 43s	remaining: 1m 44s
8337:	learn: 0.0918470	test: 0.2147954	best: 0.2145365 (6667)	total: 8m 43s	remaining: 1m 44s
8338:	learn: 0.0918454	test: 0.2147925	best: 0.2145365 (6667)	total: 8m 43s	remaining: 1m 44s
8339:	learn: 0.0918445	test: 0.2147927	best: 0.2145365 (6667)	total: 8m 43s	remaining: 1m 44s
8340:	learn: 0.0918431	test: 0.2147928	best: 0.2145365 (6667)	total: 8m 43s	remaining: 1m 44s
8341:	learn: 0.0918400	test: 0.2147927	best: 0.2145365 (6667)	total: 8m 43s	remaining: 1m 44s
8342:	learn: 0.0918374	test: 0.2147923	best: 0.2145365 (6667)	total: 8m 43s	remaining: 1m 43s
8343:	learn: 0.0918337	test: 0.2147930	best: 0.2145365 (6667)	total: 8m 43s	remaining: 1m 43s
8344:	learn: 0.0918296	test: 0.2147913	best: 0.2145365 (6667)	total: 8m 43s	remaining: 1m 43s
8345:	learn: 0.0918287	test: 0.2147915	best: 0.2145365 (6667)	total: 8m 43s	remaining: 1m 43s
8346:	learn: 0.0918258	test: 0.2147917	best: 0.2145365 (6667)	total: 8m 43s	remaining: 1m 43s
8347:	learn: 0.0918237	test: 0.2147934	best: 0.2145365 (6667)	total: 8m 44s	remaining: 1m 43s
8348:	learn: 0.0918217	test: 0.2147953	best: 0.2145365 (6667)	total: 8m 44s	remaining: 1m 43s
8349:	learn: 0.0918187	test: 0.2147950	best: 0.2145365 (6667)	total: 8m 44s	remaining: 1m 43s
8350:	learn: 0.0918176	test: 0.2147959	best: 0.2145365 (6667)	total: 8m 44s	remaining: 1m 43s
8351:	learn: 0.0918162	test: 0.2147962	best: 0.2145365 (6667)	total: 8m 44s	remaining: 1m 43s
8352:	learn: 0.0918137	test: 0.2147955	best: 0.2145365 (6667)	total: 8m 44s	remaining: 1m 43s
8353:	learn: 0.0918123	test: 0.2147956	best: 0.2145365 (6667)	total: 8m 44s	remaining: 1m 43s
8354:	learn: 0.0918100	test: 0.2147954	best: 0.2145365 (6667)	total: 8m 44s	remaining: 1m 43s
8355:	learn: 0.0918090	test: 0.2147954	best: 0.2145365 (6667)	total: 8m 44s	remaining: 1m 43s
8356:	learn: 0.0918091	test: 0.2147954	best: 0.2145365 (6667)	total: 8m 44s	remaining: 1m 43s
8357:	learn: 0.0918076	test: 0.2147943	best: 0.2145365 (6667)	total: 8m 44s	remaining: 1m 43s
8358:	learn: 0.0918065	test: 0.2147950	best: 0.2145365 (6667)	total: 8m 44s	remaining: 1m 43s
8359:	learn: 0.0917995	test: 0.2147954	best: 0.2145365 (6667)	total: 8m 44s	remaining: 1m 42s
8360:	learn: 0.0917980	test: 0.2147953	best: 0.2145365 (6667)	total: 8m 44s	remaining: 1m 42s
8361:	learn: 0.0917953	test: 0.2147958	best: 0.2145365 (6667)	total: 8m 44s	remaining: 1m 42s
8362:	learn: 0.0917942	test: 0.2147955	best: 0.2145365 (6667)	total: 8m 44s	remaining: 1m 42s
8363:	learn: 0.0917913	test: 0.2147924	best: 0.2145365 (6667)	total: 8m 45s	remaining: 1m 42s
8364:	learn: 0.0917909	test: 0.2147920	best: 0.2145365 (6667)	total: 8m 45s	remaining: 1m 42s
8365:	learn: 0.0917883	test: 0.2147924	best: 0.2145365 (6667)	total: 8m 45s	remaining: 1m 42s
8366:	learn: 0.0917880	test: 0.2147920	best: 0.2145365 (6667)	total: 8m 45s	remaining: 1m 42s
8367:	learn: 0.0917873	test: 0.2147917	best: 0.2145365 (6667)	total: 8m 45s	remaining: 1m 42s
8368:	learn: 0.0917846	test: 0.2147934	best: 0.2145365 (6667)	total: 8m 45s	remaining: 1m 42s
8369:	learn: 0.0917814	test: 0.2147932	best: 0.2145365 (6667)	total: 8m 45s	remaining: 1m 42s
8370:	learn: 0.0917765	test: 0.2147948	best: 0.2145365 (6667)	total: 8m 45s	remaining: 1m 42s
8371:	learn: 0.0917739	test: 0.2147951	best: 0.2145365 (6667)	total: 8m 45s	remaining: 1m 42s
8372:	learn: 0.0917713	test: 0.2147956	best: 0.2145365 (6667)	total: 8m 45s	remaining: 1m 42s
8373:	learn: 0.0917673	test: 0.2147964	best: 0.2145365 (6667)	total: 8m 45s	remaining: 1m 42s
8374:	learn: 0.0917658	test: 0.2147982	best: 0.2145365 (6667)	total: 8m 45s	remaining: 1m 42s
8375:	learn: 0.0917648	test: 0.2147979	best: 0.2145365 (6667)	total: 8m 46s	remaining: 1m 41s
8376:	learn: 0.0917638	test: 0.2147995	best: 0.2145365 (6667)	total: 8m 46s	remaining: 1m 41s
8377:	learn: 0.0917628	test: 0.2148012	best: 0.2145365 (6667)	total: 8m 46s	remaining: 1m 41s
8378:	learn: 0.0917617	test: 0.2147950	best: 0.2145365 (6667)	total: 8m 46s	remaining: 1m 41s
8379:	learn: 0.0917598	test: 0.2147955	best: 0.2145365 (6667)	total: 8m 46s	remaining: 1m 41s
8380:	learn: 0.0917577	test: 0.2147944	best: 0.2145365 (6667)	total: 8m 46s	remaining: 1m 41s
8381:	learn: 0.0917566	test: 0.2147943	best: 0.2145365 (6667)	total: 8m 46s	remaining: 1m 41s
8382:	learn: 0.0917556	test: 0.2147952	best: 0.2145365 (6667)	total: 8m 46s	remaining: 1m 41s
8383:	learn: 0.0917544	test: 0.2147968	best: 0.2145365 (6667)	total: 8m 46s	remaining: 1m 41s
8384:	learn: 0.0917524	test: 0.2147945	best: 0.2145365 (6667)	total: 8m 46s	remaining: 1m 41s
8385:	learn: 0.0917498	test: 0.2147941	best: 0.2145365 (6667)	total: 8m 46s	remaining: 1m 41s
8386:	learn: 0.0917468	test: 0.2147941	best: 0.2145365 (6667)	total: 8m 46s	remaining: 1m 41s
8387:	learn: 0.0917431	test: 0.2147946	best: 0.2145365 (6667)	total: 8m 47s	remaining: 1m 41s
8388:	learn: 0.0917414	test: 0.2147939	best: 0.2145365 (6667)	total: 8m 47s	remaining: 1m 41s
8389:	learn: 0.0917394	test: 0.2147942	best: 0.2145365 (6667)	total: 8m 47s	remaining: 1m 41s
8390:	learn: 0.0917374	test: 0.2147944	best: 0.2145365 (6667)	total: 8m 47s	remaining: 1m 41s
8391:	learn: 0.0917351	test: 0.2147940	best: 0.2145365 (6667)	total: 8m 47s	remaining: 1m 41s
8392:	learn: 0.0917328	test: 0.2147945	best: 0.2145365 (6667)	total: 8m 47s	remaining: 1m 40s
8393:	learn: 0.0917300	test: 0.2147950	best: 0.2145365 (6667)	total: 8m 47s	remaining: 1m 40s
8394:	learn: 0.0917297	test: 0.2147950	best: 0.2145365 (6667)	total: 8m 47s	remaining: 1m 40s
8395:	learn: 0.0917264	test: 0.2147959	best: 0.2145365 (6667)	total: 8m 47s	remaining: 1m 40s
8396:	learn: 0.0917241	test: 0.2147962	best: 0.2145365 (6667)	total: 8m 47s	remaining: 1m 40s
8397:	learn: 0.0917205	test: 0.2147955	best: 0.2145365 (6667)	total: 8m 47s	remaining: 1m 40s
8398:	learn: 0.0917178	test: 0.2147943	best: 0.2145365 (6667)	total: 8m 47s	remaining: 1m 40s
8399:	learn: 0.0917168	test: 0.2147928	best: 0.2145365 (6667)	total: 8m 47s	remaining: 1m 40s
8400:	learn: 0.0917097	test: 0.2147924	best: 0.2145365 (6667)	total: 8m 47s	remaining: 1m 40s
8401:	learn: 0.0917075	test: 0.2147925	best: 0.2145365 (6667)	total: 8m 47s	remaining: 1m 40s
8402:	learn: 0.0917050	test: 0.2147921	best: 0.2145365 (6667)	total: 8m 48s	remaining: 1m 40s
8403:	learn: 0.0917041	test: 0.2147912	best: 0.2145365 (6667)	total: 8m 48s	remaining: 1m 40s
8404:	learn: 0.0917019	test: 0.2147899	best: 0.2145365 (6667)	total: 8m 48s	remaining: 1m 40s
8405:	learn: 0.0917011	test: 0.2147895	best: 0.2145365 (6667)	total: 8m 48s	remaining: 1m 40s
8406:	learn: 0.0916990	test: 0.2147917	best: 0.2145365 (6667)	total: 8m 48s	remaining: 1m 40s
8407:	learn: 0.0916988	test: 0.2147918	best: 0.2145365 (6667)	total: 8m 48s	remaining: 1m 40s
8408:	learn: 0.0916967	test: 0.2147886	best: 0.2145365 (6667)	total: 8m 48s	remaining: 1m 40s
8409:	learn: 0.0916938	test: 0.2147869	best: 0.2145365 (6667)	total: 8m 48s	remaining: 1m 39s
8410:	learn: 0.0916925	test: 0.2147869	best: 0.2145365 (6667)	total: 8m 48s	remaining: 1m 39s
8411:	learn: 0.0916917	test: 0.2147876	best: 0.2145365 (6667)	total: 8m 48s	remaining: 1m 39s
8412:	learn: 0.0916905	test: 0.2147877	best: 0.2145365 (6667)	total: 8m 48s	remaining: 1m 39s
8413:	learn: 0.0916880	test: 0.2147888	best: 0.2145365 (6667)	total: 8m 48s	remaining: 1m 39s
8414:	learn: 0.0916865	test: 0.2147876	best: 0.2145365 (6667)	total: 8m 48s	remaining: 1m 39s
8415:	learn: 0.0916859	test: 0.2147879	best: 0.2145365 (6667)	total: 8m 48s	remaining: 1m 39s
8416:	learn: 0.0916835	test: 0.2147887	best: 0.2145365 (6667)	total: 8m 49s	remaining: 1m 39s
8417:	learn: 0.0916824	test: 0.2147895	best: 0.2145365 (6667)	total: 8m 49s	remaining: 1m 39s
8418:	learn: 0.0916780	test: 0.2147895	best: 0.2145365 (6667)	total: 8m 49s	remaining: 1m 39s
8419:	learn: 0.0916765	test: 0.2147917	best: 0.2145365 (6667)	total: 8m 49s	remaining: 1m 39s
8420:	learn: 0.0916747	test: 0.2147918	best: 0.2145365 (6667)	total: 8m 49s	remaining: 1m 39s
8421:	learn: 0.0916746	test: 0.2147914	best: 0.2145365 (6667)	total: 8m 49s	remaining: 1m 39s
8422:	learn: 0.0916723	test: 0.2147913	best: 0.2145365 (6667)	total: 8m 49s	remaining: 1m 39s
8423:	learn: 0.0916714	test: 0.2147917	best: 0.2145365 (6667)	total: 8m 49s	remaining: 1m 39s
8424:	learn: 0.0916704	test: 0.2147918	best: 0.2145365 (6667)	total: 8m 49s	remaining: 1m 38s
8425:	learn: 0.0916663	test: 0.2147881	best: 0.2145365 (6667)	total: 8m 49s	remaining: 1m 38s
8426:	learn: 0.0916614	test: 0.2147875	best: 0.2145365 (6667)	total: 8m 49s	remaining: 1m 38s
8427:	learn: 0.0916585	test: 0.2147868	best: 0.2145365 (6667)	total: 8m 49s	remaining: 1m 38s
8428:	learn: 0.0916581	test: 0.2147869	best: 0.2145365 (6667)	total: 8m 49s	remaining: 1m 38s
8429:	learn: 0.0916517	test: 0.2147942	best: 0.2145365 (6667)	total: 8m 50s	remaining: 1m 38s
8430:	learn: 0.0916503	test: 0.2147938	best: 0.2145365 (6667)	total: 8m 50s	remaining: 1m 38s
8431:	learn: 0.0916491	test: 0.2147935	best: 0.2145365 (6667)	total: 8m 50s	remaining: 1m 38s
8432:	learn: 0.0916421	test: 0.2147967	best: 0.2145365 (6667)	total: 8m 50s	remaining: 1m 38s
8433:	learn: 0.0916400	test: 0.2147940	best: 0.2145365 (6667)	total: 8m 50s	remaining: 1m 38s
8434:	learn: 0.0916378	test: 0.2147971	best: 0.2145365 (6667)	total: 8m 50s	remaining: 1m 38s
8435:	learn: 0.0916371	test: 0.2147973	best: 0.2145365 (6667)	total: 8m 50s	remaining: 1m 38s
8436:	learn: 0.0916360	test: 0.2147966	best: 0.2145365 (6667)	total: 8m 50s	remaining: 1m 38s
8437:	learn: 0.0916346	test: 0.2147951	best: 0.2145365 (6667)	total: 8m 50s	remaining: 1m 38s
8438:	learn: 0.0916326	test: 0.2147945	best: 0.2145365 (6667)	total: 8m 50s	remaining: 1m 38s
8439:	learn: 0.0916285	test: 0.2147968	best: 0.2145365 (6667)	total: 8m 50s	remaining: 1m 38s
8440:	learn: 0.0916263	test: 0.2147956	best: 0.2145365 (6667)	total: 8m 50s	remaining: 1m 38s
8441:	learn: 0.0916218	test: 0.2148000	best: 0.2145365 (6667)	total: 8m 50s	remaining: 1m 37s
8442:	learn: 0.0916200	test: 0.2148044	best: 0.2145365 (6667)	total: 8m 50s	remaining: 1m 37s
8443:	learn: 0.0916176	test: 0.2148036	best: 0.2145365 (6667)	total: 8m 50s	remaining: 1m 37s
8444:	learn: 0.0916173	test: 0.2148036	best: 0.2145365 (6667)	total: 8m 50s	remaining: 1m 37s
8445:	learn: 0.0916160	test: 0.2148039	best: 0.2145365 (6667)	total: 8m 50s	remaining: 1m 37s
8446:	learn: 0.0916140	test: 0.2148087	best: 0.2145365 (6667)	total: 8m 51s	remaining: 1m 37s
8447:	learn: 0.0916121	test: 0.2148088	best: 0.2145365 (6667)	total: 8m 51s	remaining: 1m 37s
8448:	learn: 0.0916112	test: 0.2148026	best: 0.2145365 (6667)	total: 8m 51s	remaining: 1m 37s
8449:	learn: 0.0916107	test: 0.2148082	best: 0.2145365 (6667)	total: 8m 51s	remaining: 1m 37s
8450:	learn: 0.0916087	test: 0.2148080	best: 0.2145365 (6667)	total: 8m 51s	remaining: 1m 37s
8451:	learn: 0.0916063	test: 0.2147998	best: 0.2145365 (6667)	total: 8m 51s	remaining: 1m 37s
8452:	learn: 0.0916060	test: 0.2148005	best: 0.2145365 (6667)	total: 8m 51s	remaining: 1m 37s
8453:	learn: 0.0916051	test: 0.2148005	best: 0.2145365 (6667)	total: 8m 51s	remaining: 1m 37s
8454:	learn: 0.0916037	test: 0.2148004	best: 0.2145365 (6667)	total: 8m 51s	remaining: 1m 37s
8455:	learn: 0.0916016	test: 0.2147987	best: 0.2145365 (6667)	total: 8m 51s	remaining: 1m 37s
8456:	learn: 0.0916009	test: 0.2147983	best: 0.2145365 (6667)	total: 8m 52s	remaining: 1m 37s
8457:	learn: 0.0915987	test: 0.2147987	best: 0.2145365 (6667)	total: 8m 52s	remaining: 1m 37s
8458:	learn: 0.0915970	test: 0.2147965	best: 0.2145365 (6667)	total: 8m 52s	remaining: 1m 36s
8459:	learn: 0.0915966	test: 0.2147959	best: 0.2145365 (6667)	total: 8m 52s	remaining: 1m 36s
8460:	learn: 0.0915940	test: 0.2147948	best: 0.2145365 (6667)	total: 8m 52s	remaining: 1m 36s
8461:	learn: 0.0915939	test: 0.2147954	best: 0.2145365 (6667)	total: 8m 52s	remaining: 1m 36s
8462:	learn: 0.0915922	test: 0.2147951	best: 0.2145365 (6667)	total: 8m 52s	remaining: 1m 36s
8463:	learn: 0.0915896	test: 0.2147960	best: 0.2145365 (6667)	total: 8m 52s	remaining: 1m 36s
8464:	learn: 0.0915869	test: 0.2147945	best: 0.2145365 (6667)	total: 8m 52s	remaining: 1m 36s
8465:	learn: 0.0915845	test: 0.2147969	best: 0.2145365 (6667)	total: 8m 52s	remaining: 1m 36s
8466:	learn: 0.0915833	test: 0.2147971	best: 0.2145365 (6667)	total: 8m 52s	remaining: 1m 36s
8467:	learn: 0.0915824	test: 0.2147981	best: 0.2145365 (6667)	total: 8m 52s	remaining: 1m 36s
8468:	learn: 0.0915769	test: 0.2147974	best: 0.2145365 (6667)	total: 8m 52s	remaining: 1m 36s
8469:	learn: 0.0915755	test: 0.2147973	best: 0.2145365 (6667)	total: 8m 52s	remaining: 1m 36s
8470:	learn: 0.0915732	test: 0.2147989	best: 0.2145365 (6667)	total: 8m 53s	remaining: 1m 36s
8471:	learn: 0.0915730	test: 0.2147988	best: 0.2145365 (6667)	total: 8m 53s	remaining: 1m 36s
8472:	learn: 0.0915709	test: 0.2147980	best: 0.2145365 (6667)	total: 8m 53s	remaining: 1m 36s
8473:	learn: 0.0915672	test: 0.2147975	best: 0.2145365 (6667)	total: 8m 53s	remaining: 1m 36s
8474:	learn: 0.0915645	test: 0.2147960	best: 0.2145365 (6667)	total: 8m 53s	remaining: 1m 36s
8475:	learn: 0.0915633	test: 0.2147974	best: 0.2145365 (6667)	total: 8m 53s	remaining: 1m 35s
8476:	learn: 0.0915601	test: 0.2147986	best: 0.2145365 (6667)	total: 8m 53s	remaining: 1m 35s
8477:	learn: 0.0915561	test: 0.2147992	best: 0.2145365 (6667)	total: 8m 53s	remaining: 1m 35s
8478:	learn: 0.0915551	test: 0.2148002	best: 0.2145365 (6667)	total: 8m 53s	remaining: 1m 35s
8479:	learn: 0.0915495	test: 0.2148027	best: 0.2145365 (6667)	total: 8m 53s	remaining: 1m 35s
8480:	learn: 0.0915472	test: 0.2147996	best: 0.2145365 (6667)	total: 8m 53s	remaining: 1m 35s
8481:	learn: 0.0915458	test: 0.2147993	best: 0.2145365 (6667)	total: 8m 53s	remaining: 1m 35s
8482:	learn: 0.0915443	test: 0.2148035	best: 0.2145365 (6667)	total: 8m 53s	remaining: 1m 35s
8483:	learn: 0.0915427	test: 0.2148031	best: 0.2145365 (6667)	total: 8m 53s	remaining: 1m 35s
8484:	learn: 0.0915396	test: 0.2148062	best: 0.2145365 (6667)	total: 8m 54s	remaining: 1m 35s
8485:	learn: 0.0915387	test: 0.2148072	best: 0.2145365 (6667)	total: 8m 54s	remaining: 1m 35s
8486:	learn: 0.0915350	test: 0.2148022	best: 0.2145365 (6667)	total: 8m 54s	remaining: 1m 35s
8487:	learn: 0.0915327	test: 0.2148016	best: 0.2145365 (6667)	total: 8m 54s	remaining: 1m 35s
8488:	learn: 0.0915322	test: 0.2148015	best: 0.2145365 (6667)	total: 8m 54s	remaining: 1m 35s
8489:	learn: 0.0915313	test: 0.2148015	best: 0.2145365 (6667)	total: 8m 54s	remaining: 1m 35s
8490:	learn: 0.0915300	test: 0.2148010	best: 0.2145365 (6667)	total: 8m 54s	remaining: 1m 35s
8491:	learn: 0.0915257	test: 0.2147966	best: 0.2145365 (6667)	total: 8m 54s	remaining: 1m 34s
8492:	learn: 0.0915241	test: 0.2147960	best: 0.2145365 (6667)	total: 8m 54s	remaining: 1m 34s
8493:	learn: 0.0915233	test: 0.2147963	best: 0.2145365 (6667)	total: 8m 54s	remaining: 1m 34s
8494:	learn: 0.0915222	test: 0.2147953	best: 0.2145365 (6667)	total: 8m 54s	remaining: 1m 34s
8495:	learn: 0.0915201	test: 0.2147952	best: 0.2145365 (6667)	total: 8m 54s	remaining: 1m 34s
8496:	learn: 0.0915192	test: 0.2147956	best: 0.2145365 (6667)	total: 8m 55s	remaining: 1m 34s
8497:	learn: 0.0915179	test: 0.2147959	best: 0.2145365 (6667)	total: 8m 55s	remaining: 1m 34s
8498:	learn: 0.0915155	test: 0.2147963	best: 0.2145365 (6667)	total: 8m 55s	remaining: 1m 34s
8499:	learn: 0.0915136	test: 0.2147979	best: 0.2145365 (6667)	total: 8m 55s	remaining: 1m 34s
8500:	learn: 0.0915123	test: 0.2147967	best: 0.2145365 (6667)	total: 8m 55s	remaining: 1m 34s
8501:	learn: 0.0915084	test: 0.2147957	best: 0.2145365 (6667)	total: 8m 55s	remaining: 1m 34s
8502:	learn: 0.0915059	test: 0.2147961	best: 0.2145365 (6667)	total: 8m 55s	remaining: 1m 34s
8503:	learn: 0.0915038	test: 0.2147946	best: 0.2145365 (6667)	total: 8m 55s	remaining: 1m 34s
8504:	learn: 0.0915004	test: 0.2147964	best: 0.2145365 (6667)	total: 8m 55s	remaining: 1m 34s
8505:	learn: 0.0914996	test: 0.2147976	best: 0.2145365 (6667)	total: 8m 55s	remaining: 1m 34s
8506:	learn: 0.0914978	test: 0.2147981	best: 0.2145365 (6667)	total: 8m 55s	remaining: 1m 33s
8507:	learn: 0.0914944	test: 0.2147963	best: 0.2145365 (6667)	total: 8m 55s	remaining: 1m 33s
8508:	learn: 0.0914899	test: 0.2147962	best: 0.2145365 (6667)	total: 8m 55s	remaining: 1m 33s
8509:	learn: 0.0914878	test: 0.2147958	best: 0.2145365 (6667)	total: 8m 56s	remaining: 1m 33s
8510:	learn: 0.0914835	test: 0.2147929	best: 0.2145365 (6667)	total: 8m 56s	remaining: 1m 33s
8511:	learn: 0.0914820	test: 0.2147926	best: 0.2145365 (6667)	total: 8m 56s	remaining: 1m 33s
8512:	learn: 0.0914804	test: 0.2147933	best: 0.2145365 (6667)	total: 8m 56s	remaining: 1m 33s
8513:	learn: 0.0914790	test: 0.2147939	best: 0.2145365 (6667)	total: 8m 56s	remaining: 1m 33s
8514:	learn: 0.0914771	test: 0.2147949	best: 0.2145365 (6667)	total: 8m 56s	remaining: 1m 33s
8515:	learn: 0.0914760	test: 0.2147943	best: 0.2145365 (6667)	total: 8m 56s	remaining: 1m 33s
8516:	learn: 0.0914753	test: 0.2147934	best: 0.2145365 (6667)	total: 8m 56s	remaining: 1m 33s
8517:	learn: 0.0914740	test: 0.2147923	best: 0.2145365 (6667)	total: 8m 56s	remaining: 1m 33s
8518:	learn: 0.0914734	test: 0.2147920	best: 0.2145365 (6667)	total: 8m 56s	remaining: 1m 33s
8519:	learn: 0.0914726	test: 0.2147919	best: 0.2145365 (6667)	total: 8m 56s	remaining: 1m 33s
8520:	learn: 0.0914711	test: 0.2147922	best: 0.2145365 (6667)	total: 8m 56s	remaining: 1m 33s
8521:	learn: 0.0914700	test: 0.2147935	best: 0.2145365 (6667)	total: 8m 56s	remaining: 1m 33s
8522:	learn: 0.0914674	test: 0.2147942	best: 0.2145365 (6667)	total: 8m 56s	remaining: 1m 33s
8523:	learn: 0.0914667	test: 0.2147948	best: 0.2145365 (6667)	total: 8m 56s	remaining: 1m 32s
8524:	learn: 0.0914619	test: 0.2147955	best: 0.2145365 (6667)	total: 8m 56s	remaining: 1m 32s
8525:	learn: 0.0914586	test: 0.2147951	best: 0.2145365 (6667)	total: 8m 57s	remaining: 1m 32s
8526:	learn: 0.0914568	test: 0.2147940	best: 0.2145365 (6667)	total: 8m 57s	remaining: 1m 32s
8527:	learn: 0.0914539	test: 0.2147936	best: 0.2145365 (6667)	total: 8m 57s	remaining: 1m 32s
```

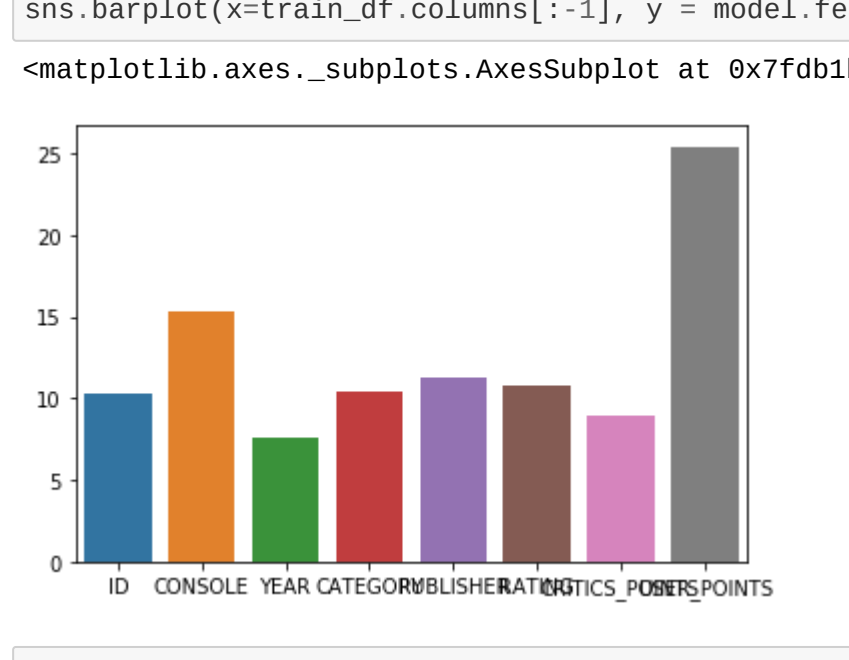
```
2.6s
9806:	learn: 0.0888085	test: 0.2148983 best: 0.2145365 (6667)	total: 10m 37s	remaining: 1
2.6s
9807:	learn: 0.0888080	test: 0.2148982 best: 0.2145365 (6667)	total: 10m 37s	remaining: 1
2.5s
9808:	learn: 0.0888052	test: 0.2149006 best: 0.2145365 (6667)	total: 10m 38s	remaining: 1
2.4s
9809:	learn: 0.0888044	test: 0.2149003 best: 0.2145365 (6667)	total: 10m 38s	remaining: 1
2.4s
9810:	learn: 0.0888021	test: 0.2149026 best: 0.2145365 (6667)	total: 10m 38s	remaining: 1
2.3s
9811:	learn: 0.0888010	test: 0.2149032 best: 0.2145365 (6667)	total: 10m 38s	remaining: 1
2.2s
9812:	learn: 0.0887986	test: 0.2149048 best: 0.2145365 (6667)	total: 10m 38s	remaining: 1
2.2s
9813:	learn: 0.0887967	test: 0.2149040 best: 0.2145365 (6667)	total: 10m 38s	remaining: 1
2.1s
9814:	learn: 0.0887965	test: 0.2149022 best: 0.2145365 (6667)	total: 10m 38s	remaining: 12
s
9815:	learn: 0.0887944	test: 0.2149045 best: 0.2145365 (6667)	total: 10m 38s	remaining: 12
s
9816:	learn: 0.0887934	test: 0.2149046 best: 0.2145365 (6667)	total: 10m 38s	remaining: 1
1.9s
9817:	learn: 0.0887912	test: 0.2149054 best: 0.2145365 (6667)	total: 10m 38s	remaining: 1
1.8s
9818:	learn: 0.0887900	test: 0.2149040 best: 0.2145365 (6667)	total: 10m 38s	remaining: 1
1.8s
9819:	learn: 0.0887885	test: 0.2149048 best: 0.2145365 (6667)	total: 10m 39s	remaining: 1
1.7s
9820:	learn: 0.0887875	test: 0.2149052 best: 0.2145365 (6667)	total: 10m 39s	remaining: 1
1.6s
9821:	learn: 0.0887861	test: 0.2149061 best: 0.2145365 (6667)	total: 10m 39s	remaining: 1
1.6s
9822:	learn: 0.0887833	test: 0.2149059 best: 0.2145365 (6667)	total: 10m 39s	remaining: 1
1.5s
9823:	learn: 0.0887829	test: 0.2149053 best: 0.2145365 (6667)	total: 10m 39s	remaining: 1
1.5s
9824:	learn: 0.0887809	test: 0.2149060 best: 0.2145365 (6667)	total: 10m 39s	remaining: 1
1.4s
9825:	learn: 0.0887796	test: 0.2149051 best: 0.2145365 (6667)	total: 10m 39s	remaining: 1
1.3s
9826:	learn: 0.0887778	test: 0.2149056 best: 0.2145365 (6667)	total: 10m 39s	remaining: 1
1.3s
9827:	learn: 0.0887769	test: 0.2149051 best: 0.2145365 (6667)	total: 10m 39s	remaining: 1
1.2s
9828:	learn: 0.0887766	test: 0.2149052 best: 0.2145365 (6667)	total: 10m 39s	remaining: 1
1.1s
9829:	learn: 0.0887743	test: 0.2149105 best: 0.2145365 (6667)	total: 10m 39s	remaining: 1
1.1s
9830:	learn: 0.0887739	test: 0.2149113 best: 0.2145365 (6667)	total: 10m 39s	remaining: 11
s
9831:	learn: 0.0887717	test: 0.2149104 best: 0.2145365 (6667)	total: 10m 40s	remaining: 1
0.9s
9832:	learn: 0.0887700	test: 0.2149102 best: 0.2145365 (6667)	total: 10m 40s	remaining: 1
0.9s
9833:	learn: 0.0887696	test: 0.2149099 best: 0.2145365 (6667)	total: 10m 40s	remaining: 1
0.8s
9834:	learn: 0.0887661	test: 0.2149051 best: 0.2145365 (6667)	total: 10m 40s	remaining: 1
0.7s
9835:	learn: 0.0887648	test: 0.2149050 best: 0.2145365 (6667)	total: 10m 40s	remaining: 1
0.7s
9836:	learn: 0.0887633	test: 0.2149062 best: 0.2145365 (6667)	total: 10m 40s	remaining: 1
0.6s
9837:	learn: 0.0887611	test: 0.2149070 best: 0.2145365 (6667)	total: 10m 40s	remaining: 1
0.5s
9838:	learn: 0.0887606	test: 0.2149073 best: 0.2145365 (6667)	total: 10m 40s	remaining: 1
0.5s
9839:	learn: 0.0887587	test: 0.2149072 best: 0.2145365 (6667)	total: 10m 40s	remaining: 1
0.4s
9840:	learn: 0.0887556	test: 0.2149072 best: 0.2145365 (6667)	total: 10m 40s	remaining: 1
0.3s
9841:	learn: 0.0887548	test: 0.2149073 best: 0.2145365 (6667)	total: 10m 40s	remaining: 1
0.3s
9842:	learn: 0.0887527	test: 0.2149083 best: 0.2145365 (6667)	total: 10m 40s	remaining: 1
0.2s
9843:	learn: 0.0887509	test: 0.2149082 best: 0.2145365 (6667)	total: 10m 40s	remaining: 1
0.2s
9844:	learn: 0.0887487	test: 0.2149096 best: 0.2145365 (6667)	total: 10m 40s	remaining: 1
0.1s
9845:	learn: 0.0887461	test: 0.2149105 best: 0.2145365 (6667)	total: 10m 40s	remaining: 10
s
9846:	learn: 0.0887443	test: 0.2149105 best: 0.2145365 (6667)	total: 10m 40s	remaining: 9.
96s
9847:	learn: 0.0887422	test: 0.2149100 best: 0.2145365 (6667)	total: 10m 40s	remaining: 9.
89s
9848:	learn: 0.0887397	test: 0.2149076 best: 0.2145365 (6667)	total: 10m 40s	remaining: 9.
82s
9849:	learn: 0.0887390	test: 0.2149078 best: 0.2145365 (6667)	total: 10m 40s	remaining: 9.
76s
9850:	learn: 0.0887364	test: 0.2149075 best: 0.2145365 (6667)	total: 10m 41s	remaining: 9.
7s
9851:	learn: 0.0887349	test: 0.2149082 best: 0.2145365 (6667)	total: 10m 41s	remaining: 9.
64s
9852:	learn: 0.0887340	test: 0.2149100 best: 0.2145365 (6667)	total: 10m 41s	remaining: 9.
57s
9853:	learn: 0.0887289	test: 0.2149072 best: 0.2145365 (6667)	total: 10m 41s	remaining: 9.
51s
9854:	learn: 0.0887280	test: 0.2149076 best: 0.2145365 (6667)	total: 10m 41s	remaining: 9.
44s
9855:	learn: 0.0887256	test: 0.2149061 best: 0.2145365 (6667)	total: 10m 41s	remaining: 9.
38s
9856:	learn: 0.0887239	test: 0.2149060 best: 0.2145365 (6667)	total: 10m 41s	remaining: 9.
31s
9857:	learn: 0.0887223	test: 0.2149039 best: 0.2145365 (6667)	total: 10m 41s	remaining: 9.
25s
9858:	learn: 0.0887202	test: 0.2149049 best: 0.2145365 (6667)	total: 10m 42s	remaining: 9.
18s
9859:	learn: 0.0887165	test: 0.2149073 best: 0.2145365 (6667)	total: 10m 42s	remaining: 9.
12s
9860:	learn: 0.0887162	test: 0.2149079 best: 0.2145365 (6667)	total: 10m 42s	remaining: 9.
05s
9861:	learn: 0.0887146	test: 0.2149091 best: 0.2145365 (6667)	total: 10m 42s	remaining: 8.
98s
9862:	learn: 0.0887129	test: 0.2149106 best: 0.2145365 (6667)	total: 10m 42s	remaining: 8.
92s
9863:	learn: 0.0887108	test: 0.2149082 best: 0.2145365 (6667)	total: 10m 42s	remaining: 8.
86s
9864:	learn: 0.0887065	test: 0.2149090 best: 0.2145365 (6667)	total: 10m 42s	remaining: 8.
79s
9865:	learn: 0.0887043	test: 0.2149104 best: 0.2145365 (6667)	total: 10m 42s	remaining: 8.
72s
9866:	learn: 0.0887032	test: 0.2149103 best: 0.2145365 (6667)	total: 10m 42s	remaining: 8.
66s
9867:	learn: 0.0886996	test: 0.2149106 best: 0.2145365 (6667)	total: 10m 42s	remaining: 8.
59s
9868:	learn: 0.0886988	test: 0.2149107 best: 0.2145365 (6667)	total: 10m 42s	remaining: 8.
53s
9869:	learn: 0.0886982	test: 0.2149108 best: 0.2145365 (6667)	total: 10m 42s	remaining: 8.
46s
9870:	learn: 0.0886967	test: 0.2149111 best: 0.2145365 (6667)	total: 10m 42s	remaining: 8.
4s
9871:	learn: 0.0886942	test: 0.2149106 best: 0.2145365 (6667)	total: 10m 42s	remaining: 8.
34s
9872:	learn: 0.0886927	test: 0.2149105 best: 0.2145365 (6667)	total: 10m 43s	remaining: 8.
27s
9873:	learn: 0.0886884	test: 0.2149103 best: 0.2145365 (6667)	total: 10m 43s	remaining: 8.
21s
9874:	learn: 0.0886833	test: 0.2149120 best: 0.2145365 (6667)	total: 10m 43s	remaining: 8.
14s
9875:	learn: 0.0886814	test: 0.2149120 best: 0.2145365 (6667)	total: 10m 43s	remaining: 8.
08s
9876:	learn: 0.0886780	test: 0.2149127 best: 0.2145365 (6667)	total: 10m 43s	remaining: 8.
01s
9877:	learn: 0.0886754	test: 0.2149113 best: 0.2145365 (6667)	total: 10m 43s	remaining: 7.
95s
9878:	learn: 0.0886736	test: 0.2149111 best: 0.2145365 (6667)	total: 10m 43s	remaining: 7.
88s
9879:	learn: 0.0886722	test: 0.2149112 best: 0.2145365 (6667)	total: 10m 43s	remaining: 7.
82s
9880:	learn: 0.0886713	test: 0.2149116 best: 0.2145365 (6667)	total: 10m 43s	remaining: 7.
75s
9881:	learn: 0.0886698	test: 0.2149104 best: 0.2145365 (6667)	total: 10m 43s	remaining: 7.
69s
9882:	learn: 0.0886656	test: 0.2149104 best: 0.2145365 (6667)	total: 10m 43s	remaining: 7.
62s
9883:	learn: 0.0886643	test: 0.2149109 best: 0.2145365 (6667)	total: 10m 43s	remaining: 7.
56s
9884:	learn: 0.0886597	test: 0.2149103 best: 0.2145365 (6667)	total: 10m 44s	remaining: 7.
49s
9885:	learn: 0.0886588	test: 0.2149107 best: 0.2145365 (6667)	total: 10m 44s	remaining: 7.
43s
9886:	learn: 0.0886578	test: 0.2149106 best: 0.2145365 (6667)	total: 10m 44s	remaining: 7.
36s
9887:	learn: 0.0886568	test: 0.2149119 best: 0.2145365 (6667)	total: 10m 44s	remaining: 7.
3s
9888:	learn: 0.0886563	test: 0.2149121 best: 0.2145365 (6667)	total: 10m 44s	remaining: 7.
23s
9889:	learn: 0.0886536	test: 0.2149144 best: 0.2145365 (6667)	total: 10m 44s	remaining: 7.
17s
9890:	learn: 0.0886525	test: 0.2149138 best: 0.2145365 (6667)	total: 10m 44s	remaining: 7.
1s
9891:	learn: 0.0886502	test: 0.2149135 best: 0.2145365 (6667)	total: 10m 44s	remaining: 7.
04s
9892:	learn: 0.0886501	test: 0.2149132 best: 0.2145365 (6667)	total: 10m 45s	remaining: 6.
98s
9893:	learn: 0.0886479	test: 0.2149126 best: 0.2145365 (6667)	total: 10m 45s	remaining: 6.
91s
9894:	learn: 0.0886464	test: 0.2149132 best: 0.2145365 (6667)	total: 10m 45s	remaining: 6.
85s
9895:	learn: 0.0886435	test: 0.2149150 best: 0.2145365 (6667)	total: 10m 45s	remaining: 6.
78s
9896:	learn: 0.0886416	test: 0.2149149 best: 0.2145365 (6667)	total: 10m 45s	remaining: 6.
72s
9897:	learn: 0.0886410	test: 0.2149149 best: 0.2145365 (6667)	total: 10m 45s	remaining: 6.
65s
9898:	learn: 0.0886403	test: 0.2149137 best: 0.2145365 (6667)	total: 10m 45s	remaining: 6.
58s
9899:	learn: 0.0886403	test: 0.2149142 best: 0.2145365 (6667)	total: 10m 45s	remaining: 6.
52s
9900:	learn: 0.0886382	test: 0.2149092 best: 0.2145365 (6667)	total: 10m 45s	remaining: 6.
46s
9901:	learn: 0.0886363	test: 0.2149073 best: 0.2145365 (6667)	total: 10m 45s	remaining: 6.
39s
9902:	learn: 0.0886332	test: 0.2149085 best: 0.2145365 (6667)	total: 10m 45s	remaining: 6.
33s
9903:	learn: 0.0886298	test: 0.2149070 best: 0.2145365 (6667)	total: 10m 45s	remaining: 6.
26s
9904:	learn: 0.0886282	test: 0.2149079 best: 0.2145365 (6667)	total: 10m 45s	remaining: 6.
19s
9905:	learn: 0.0886243	test: 0.2149081 best: 0.2145365 (6667)	total: 10m 45s	remaining: 6.
13s
9906:	learn: 0.0886230	test: 0.2149104 best: 0.2145365 (6667)	total: 10m 46s	remaining: 6.
06s
9907:	learn: 0.0886206	test: 0.2149119 best: 0.2145365 (6667)	total: 10m 46s	remaining: 6s
9908:	learn: 0.0886204	test: 0.2149123 best: 0.2145365 (6667)	total: 10m 46s	remaining: 5.
94s
9909:	learn: 0.0886183	test: 0.2149130 best: 0.2145365 (6667)	total: 10m 46s	remaining: 5.
87s
9910:	learn: 0.0886174	test: 0.2149141 best: 0.2145365 (6667)	total: 10m 46s	remaining: 5.
81s
9911:	learn: 0.0886155	test: 0.2149153 best: 0.2145365 (6667)	total: 10m 46s	remaining: 5.
74s
9912:	learn: 0.0886129	test: 0.2149148 best: 0.2145365 (6667)	total: 10m 46s	remaining: 5.
68s
9913:	learn: 0.0886123	test: 0.2149128 best: 0.2145365 (6667)	total: 10m 46s	remaining: 5.
61s
9914:	learn: 0.0886112	test: 0.2149127 best: 0.2145365 (6667)	total: 10m 46s	remaining: 5.
55s
9915:	learn: 0.0886093	test: 0.2149116 best: 0.2145365 (6667)	total: 10m 46s	remaining: 5.
48s
9916:	learn: 0.0886087	test: 0.2149113 best: 0.2145365 (6667)	total: 10m 47s	remaining: 5.
42s
9917:	learn: 0.0886065	test: 0.2149111 best: 0.2145365 (6667)	total: 10m 47s	remaining: 5.
35s
9918:	learn: 0.0886051	test: 0.2149107 best: 0.2145365 (6667)	total: 10m 47s	remaining: 5.
28s
9919:	learn: 0.0886018	test: 0.2149102 best: 0.2145365 (6667)	total: 10m 47s	remaining: 5.
22s
9920:	learn: 0.0885997	test: 0.2149112 best: 0.2145365 (6667)	total: 10m 47s	remaining: 5.
15s
9921:	learn: 0.0885984	test: 0.2149110 best: 0.2145365 (6667)	total: 10m 47s	remaining: 5.
09s
9922:	learn: 0.0885954	test: 0.2149093 best: 0.2145365 (6667)	total: 10m 47s	remaining: 5.
02s
9923:	learn: 0.0885922	test: 0.2149093 best: 0.2145365 (6667)	total: 10m 47s	remaining: 4.
96s
9924:	learn: 0.0885894	test: 0.2149137 best: 0.2145365 (6667)	total: 10m 47s	remaining: 4.
89s
9925:	learn: 0.0885857	test: 0.2149124 best: 0.2145365 (6667)	total: 10m 47s	remaining: 4.
83s
9926:	learn: 0.0885835	test: 0.2149123 best: 0.2145365 (6667)	total: 10m 47s	remaining: 4.
76s
9927:	learn: 0.0885829	test: 0.2149124 best: 0.2145365 (6667)	total: 10m 48s	remaining: 4.
7s
9928:	learn: 0.0885817	test: 0.2149131 best: 0.2145365 (6667)	total: 10m 48s	remaining: 4.
64s
9929:	learn: 0.0885803	test: 0.2149110 best: 0.2145365 (6667)	total: 10m 48s	remaining: 4.
57s
9930:	learn: 0.0885793	test: 0.2149098 best: 0.2145365 (6667)	total: 10m 48s	remaining: 4.
5s
9931:	learn: 0.0885792	test: 0.2149101 best: 0.2145365 (6667)	total: 10m 48s	remaining: 4.
44s
9932:	learn: 0.0885765	test: 0.2149123 best: 0.2145365 (6667)	total: 10m 48s	remaining: 4.
38s
9933:	learn: 0.0885763	test: 0.2149123 best: 0.2145365 (6667)	total: 10m 48s	remaining: 4.
31s
9934:	learn: 0.0885745	test: 0.2149137 best: 0.2145365 (6667)	total: 10m 48s	remaining: 4.
24s
9935:	learn: 0.0885734	test: 0.2149138 best: 0.2145365 (6667)	total: 10m 48s	remaining: 4.
18s
9936:	learn: 0.0885709	test: 0.2149142 best: 0.2145365 (6667)	total: 10m 48s	remaining: 4.
11s
9937:	learn: 0.0885681	test: 0.2149137 best: 0.2145365 (6667)	total: 10m 48s	remaining: 4.
05s
9938:	learn: 0.0885643	test: 0.2149137 best: 0.2145365 (6667)	total: 10m 48s	remaining: 3.
98s
9939:	learn: 0.0885628	test: 0.2149143 best: 0.2145365 (6667)	total: 10m 49s	remaining: 3.
92s
9940:	learn: 0.0885592	test: 0.2149121 best: 0.2145365 (6667)	total: 10m 49s	remaining: 3.
85s
9941:	learn: 0.0885579	test: 0.2149133 best: 0.2145365 (6667)	total: 10m 49s	remaining: 3.
79s
9942:	learn: 0.0885569	test: 0.2149143 best: 0.2145365 (6667)	total: 10m 49s	remaining: 3.
72s
9943:	learn: 0.0885545	test: 0.2149157 best: 0.2145365 (6667)	total: 10m 49s	remaining: 3.
66s
9944:	learn: 0.0885539	test: 0.2149164 best: 0.2145365 (6667)	total: 10m 49s	remaining: 3.
59s
9945:	learn: 0.0885517	test: 0.2149164 best: 0.2145365 (6667)	total: 10m 49s	remaining: 3.
53s
9946:	learn: 0.0885507	test: 0.2149166 best: 0.2145365 (6667)	total: 10m 50s	remaining: 3.
46s
9947:	learn: 0.0885475	test: 0.2149164 best: 0.2145365 (6667)	total: 10m 50s	remaining: 3.
4s
9948:	learn: 0.0885464	test: 0.2149166 best: 0.2145365 (6667)	total: 10m 50s	remaining: 3.
33s
9949:	learn: 0.0885454	test: 0.2149166 best: 0.2145365 (6667)	total: 10m 50s	remaining: 3.
27s
9950:	learn: 0.0885446	test: 0.2149159 best: 0.2145365 (6667)	total: 10m 50s	remaining: 3.
2s
9951:	learn: 0.0885433	test: 0.2149140 best: 0.2145365 (6667)	total: 10m 50s	remaining: 3.
14s
9952:	learn: 0.0885419	test: 0.2149126 best: 0.2145365 (6667)	total: 10m 50s	remaining: 3.
07s
9953:	learn: 0.0885405	test: 0.2149122 best: 0.2145365 (6667)	total: 10m 50s	remaining: 3.
01s
9954:	learn: 0.0885388	test: 0.2149117 best: 0.2145365 (6667)	total: 10m 50s	remaining: 2.
94s
9955:	learn: 0.0885385	test: 0.2149115 best: 0.2145365 (6667)	total: 10m 50s	remaining: 2.
88s
9956:	learn: 0.0885380	test: 0.2149146 best: 0.2145365 (6667)	total: 10m 50s	remaining: 2.
81s
9957:	learn: 0.0885378	test: 0.2149145 best: 0.2145365 (6667)	total: 10m 50s	remaining: 2.
75s
9958:	learn: 0.0885336	test: 0.2149133 best: 0.2145365 (6667)	total: 10m 50s	remaining: 2.
68s
9959:	learn: 0.0885327	test: 0.2149138 best: 0.2145365 (6667)	total: 10m 50s	remaining: 2.
61s
9960:	learn: 0.0885315	test: 0.2149134 best: 0.2145365 (6667)	total: 10m 51s	remaining: 2.
55s
9961:	learn: 0.0885288	test: 0.2149140 best: 0.2145365 (6667)	total: 10m 51s	remaining: 2.
48s
9962:	learn: 0.0885280	test: 0.2149142 best: 0.2145365 (6667)	total: 10m 51s	remaining: 2.
42s
9963:	learn: 0.0885249	test: 0.2149141 best: 0.2145365 (6667)	total: 10m 51s	remaining: 2.
35s
9964:	learn: 0.0885238	test: 0.2149134 best: 0.2145365 (6667)	total: 10m 51s	remaining: 2.
29s
9965:	learn: 0.0885204	test: 0.2149170 best: 0.2145365 (6667)	total: 10m 51s	remaining: 2.
22s
9966:	learn: 0.0885196	test: 0.2149163 best: 0.2145365 (6667)	total: 10m 52s	remaining: 2.
16s
9967:	learn: 0.0885186	test: 0.2149163 best: 0.2145365 (6667)	total: 10m 52s	remaining: 2.
09s
9968:	learn: 0.0885176	test: 0.2149164 best: 0.2145365 (6667)	total: 10m 52s	remaining: 2.
03s
9969:	learn: 0.0885172	test: 0.2149163 best: 0.2145365 (6667)	total: 10m 52s	remaining: 1.
96s
9970:	learn: 0.0885159	test: 0.2149181 best: 0.2145365 (6667)	total: 10m 52s	remaining: 1.
9s
9971:	learn: 0.0885153	test: 0.2149183 best: 0.2145365 (6667)	total: 10m 52s	remaining: 1.
83s
9972:	learn: 0.0885132	test: 0.2149181 best: 0.2145365 (6667)	total: 10m 52s	remaining: 1.
77s
9973:	learn: 0.0885102	test: 0.2149180 best: 0.2145365 (6667)	total: 10m 52s	remaining: 1.
7s
9974:	learn: 0.0885086	test: 0.2149185 best: 0.2145365 (6667)	total: 10m 52s	remaining: 1.
64s
9975:	learn: 0.0885072	test: 0.2149203 best: 0.2145365 (6667)	total: 10m 52s	remaining: 1.
57s
9976:	learn: 0.0885049	test: 0.2149190 best: 0.2145365 (6667)	total: 10m 52s	remaining: 1.
51s
9977:	learn: 0.0885037	test: 0.2149203 best: 0.2145365 (6667)	total: 10m 52s	remaining: 1.
44s
9978:	learn: 0.0885027	test: 0.2149203 best: 0.2145365 (6667)	total: 10m 52s	remaining: 1.
37s
9979:	learn: 0.0884993	test: 0.2149165 best: 0.2145365 (6667)	total: 10m 52s	remaining: 1.
31s
9980:	learn: 0.0884966	test: 0.2149152 best: 0.2145365 (6667)	total: 10m 52s	remaining: 1.
24s
9981:	learn: 0.0884943	test: 0.2149144 best: 0.2145365 (6667)	total: 10m 52s	remaining: 1.
18s
9982:	learn: 0.0884929	test: 0.2149168 best: 0.2145365 (6667)	total: 10m 53s	remaining: 1.
11s
9983:	learn: 0.0884912	test: 0.2149161 best: 0.2145365 (6667)	total: 10m 53s	remaining: 1.
05s
9984:	learn: 0.0884894	test: 0.2149173 best: 0.2145365 (6667)	total: 10m 53s	remaining: 98
2ms
9985:	learn: 0.0884885	test: 0.2149173 best: 0.2145365 (6667)	total: 10m 53s	remaining: 91
7ms
9986:	learn: 0.0884883	test: 0.2149175 best: 0.2145365 (6667)	total: 10m 53s	remaining: 85
1ms
9987:	learn: 0.0884871	test: 0.2149165 best: 0.2145365 (6667)	total: 10m 53s	remaining: 78
6ms
9988:	learn: 0.0884848	test: 0.2149171 best: 0.2145365 (6667)	total: 10m 53s	remaining: 72
0ms
9989:	learn: 0.0884826	test: 0.2149166 best: 0.2145365 (6667)	total: 10m 54s	remaining: 65
5ms
9990:	learn: 0.0884808	test: 0.2149150 best: 0.2145365 (6667)	total: 10m 54s	remaining: 58
9ms
9991:	learn: 0.0884782	test: 0.2149174 best: 0.2145365 (6667)	total: 10m 54s	remaining: 52
4ms
9992:	learn: 0.0884768	test: 0.2149171 best: 0.2145365 (6667)	total: 10m 54s	remaining: 45
8ms
9993:	learn: 0.0884739	test: 0.2149135 best: 0.2145365 (6667)	total: 10m 54s	remaining: 39
3ms
9994:	learn: 0.0884724	test: 0.2149139 best: 0.2145365 (6667)	total: 10m 54s	remaining: 32
7ms
9995:	learn: 0.0884715	test: 0.2149142 best: 0.2145365 (6667)	total: 10m 54s	remaining: 26
2ms
9996:	learn: 0.0884704	test: 0.2149145 best: 0.2145365 (6667)	total: 10m 54s	remaining: 19
6ms
9997:	learn: 0.0884695	test: 0.2149141 best: 0.2145365 (6667)	total: 10m 54s	remaining: 13
1ms
9998:	learn: 0.0884665	test: 0.2149144 best: 0.2145365 (6667)	total: 10m 54s	remaining: 6
5.5ms
9999:	learn: 0.0884656	test: 0.2149145 best: 0.2145365 (6667)	total: 10m 54s	remaining: 0u
s

bestTest = 0.2145364513
bestIteration = 6667

Shrink model to first 6668 iterations.
```

Out[166]: <catboost.core.CatBoostRegressor at 0x7fdb1ae60990>

In [167]:
```
train_df.columns
```
Out[167]:
```
Index(['ID', 'CONSOLE', 'YEAR', 'CATEGORY', 'PUBLISHER', 'RATING',
       'CRITICS_POINTS', 'USER_POINTS', 'SalesInMillions'],
      dtype='object')
```

In [168]:
```
model.feature_importances_
```
Out[168]:
```
array([10.23530811, 15.34547601,  7.59555555, 10.37088478, 11.25065248,
       10.8061883 ,  8.98177866, 25.41415611])
```

In [169]:
```
sns.barplot(x=train_df.columns[:-1], y = model.feature_importances_)
```
Out[169]: <matplotlib.axes._subplots.AxesSubplot at 0x7fdb1b16cb50>



In [170]:
```
model.get_best_score()
```
Out[170]:
```
{'learn': {'RMSE': 0.08846563372293526},
 'validation': {'RMSE': 0.21453645127671214}}
```

In [171]:
```
model.best_iteration_
```
Out[171]: 6667

In [177]:
```
pred = model.predict(X_test)

print("RMSE ERROR == ",np.sqrt(mean_squared_error(inv_boxcox(y_test,0.27),inv_boxcox(pred,0.27))))
```
```
RMSE ERROR ==  1.6338228221642215
```

In [173]:
```
test_df.dtypes
```
Out[173]:
```
ID                  int64
CONSOLE            object
YEAR               object
CATEGORY           object
PUBLISHER          object
RATING             object
CRITICS_POINTS    float64
USER_POINTS       float64
dtype: object
```

In [176]:
```
submission = pd.DataFrame()

submission["SalesInMillions"] = inv_boxcox(model.predict(test_df),0.27)

submission.head()

submission.to_csv("sub.csv",index = False)
```

In [175]:
```
submission.head()
```
Out[175]:

	SalesInMillions
0	1.672469
1	1.991664
2	4.589158
3	2.328596
4	1.527706