Factors determining Video Game Sales

John Gilmore
Professor Hoderlein/Joseph Cooprider
Machine Learning for Economics
12/19/18

Introduction and Research Question

In my investigation, I sought to analyze the importance of a number of factors for the global sales of video games. Specifically, I sought to investigate factors I determined to be external to the game itself which I initially thought of as the system the game is played on, the ESRB rating, among others. I ultimately settled on a data set that I found on Kaggle called "Video Game Sales with Rating" which has 15 variables and 16,719 observations. The dataset included the sales numbers from Vgchartz for 16,719 videogames in North America, Europe, Japan, and other territories, as well as a global sales value. Also there are accompanying metacritic scores from both critics and users, the count of both critic and user scores, the game's publisher, developer, the system its played on, and ESRB rating among others. In order to understand the importance of each of these factors I intend to implement a linear regression and then use a lasso in order to shrink the number of regressors down to isolate the potentially more important regressors. I will finish by comparing the models in order to determine which is superior.

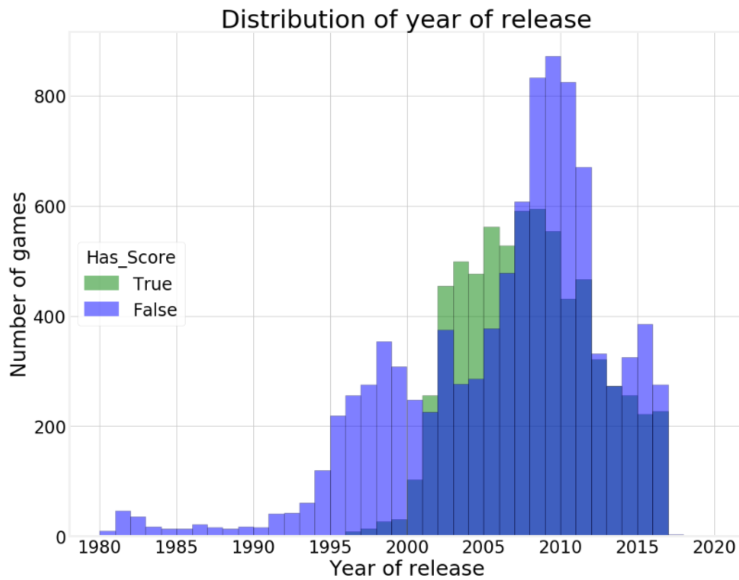
My left hand side variable is Global sales, and the other mentioned variables will make up the base of my variables of interest. Since video games have become such a global, and interconnected product, I chose to focus on Global sales and not include the regional sales figures in any of my regressions for fear of jolting up collinearity. The mean of global sales is 536,170 units, or .53617 million units as it appears in the dataset. The mean critic score is 68.99463 out of 100 and the mean user score is 7.126330 out of 10. Data summary and the breakdown of game releases by year can be seen below.

	Name	System	Year	Genre	Publisher	NA	EU	JP	Other	Global	Critic_Score	Critic_Count	User_Score	User_Count	Developer	Rating	Age
count	16448	16448	16448.000000	16448	16416	16448.000000	16448.000000	16448.000000	16448.000000	16448.00000	7983.000000	7983.000000	7463.000000	7463.000000	9907	9769	16448.000000
unique	11429	31	NaN	12	579	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1680	8	NaN
top	Need for Speed: Most Wanted	PS2	NaN	Action	Electronic Arts	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Ubisoft	E	NaN
freq	12	2127	NaN	3308	1344	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	201	3922	NaN
mean	NaN	NaN	2006.488996	NaN	NaN	0.263965	0.145895	0.078472	0.047583	0.53617	68.994363	26.441313	7.126330	163.015141	NaN	NaN	11.511004
std	NaN	NaN	5.877470	NaN	NaN	0.818286	0.506660	0.311064	0.187984	1.55846	13.920060	19.008136	1.499447	563.863327	NaN	NaN	5.877470
min	NaN	NaN	1980.000000	NaN	NaN	0.000000	0.000000	0.000000	0.000000	0.01000	13.000000	3.000000	0.000000	4.000000	NaN	NaN	-2.000000
25%	NaN	NaN	2003.000000	NaN	NaN	0.000000	0.000000	0.000000	0.000000	0.06000	60.000000	12.000000	6.400000	10.000000	NaN	NaN	8.000000
50%	NaN	NaN	2007.000000	NaN	NaN	0.080000	0.020000	0.000000	0.010000	0.17000	71.000000	22.000000	7.500000	24.000000	NaN	NaN	11.000000
75%	NaN	NaN	2010.000000	NaN	NaN	0.240000	0.110000	0.040000	0.030000	0.47000	79.000000	36.000000	8.200000	81.000000	NaN	NaN	15.000000
max	NaN	NaN	2020.000000	NaN	NaN	41.360000	28.960000	10.220000	10.570000	82.53000	98.000000	113.000000	9.700000	10665.000000	NaN	NaN	38.000000

Data Manipulation

The most apparent problem that needed to be dealt with in the data is the large number of missing variables. Just upon visual inspection it is clear that many of the games do not have listed Metacritic reviews, publisher, developer, or ESRB rating. Fortunately, almost all of the games have their global sales information but in order to use the data set effectively the missing values will have to be dealt with.

The primary source of missing information come from the Metacritic scores. An unfortunately large proportion of the games in the dataset do not have this information, as shown in the graph below.



It is likely that the general absence of scored games through the 80s up to about 2000 can be blamed on video games being a still niche medium and a lack of the websites and magazines dedicated to reviewing videogames that we have now like IGN and GameInformer. Further inspection into the missing values shows that a large 54.6 percent of games are missing the user score. The large proportion of missing values is discouraging but an inability to find another dataset with the information I want prompted me to stick with this one. Thanks to the otherwise large size of the dataset for my purposes I felt comfortable dropping observations with missing variables and as such, all observations with a null were dropped from the set.

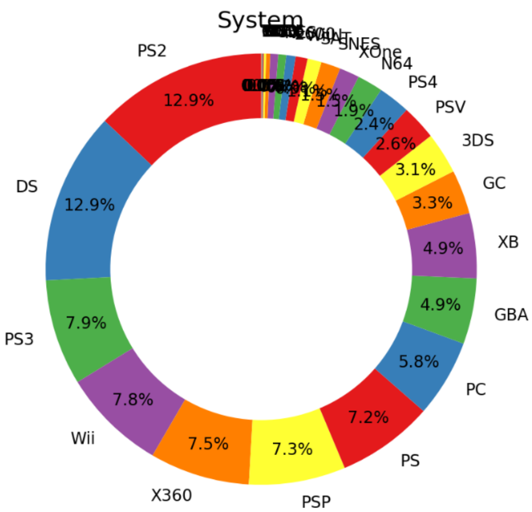
	Missing Values	% of Total Values
Name	0	0.000000
System	0	0.000000
Year	0	0.000000
Genre	0	0.000000
Publisher	32	0.194553
NA	0	0.000000
EU	0	0.000000
JP	0	0.000000
Other	0	0.000000
Global	0	0.000000
Critic_Score	8465	51.465224
Critic_Count	8465	51.465224
User_Score	8985	54.626702
User_Count	8985	54.626702
Developer	6541	39.767753
Rating	6679	40.606761
Age	0	0.000000

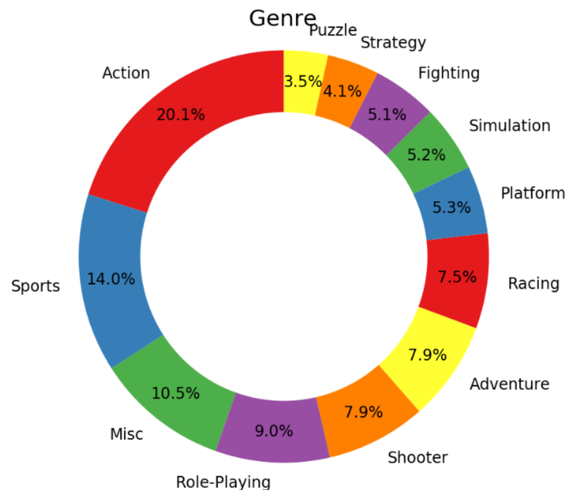
After dropping the nulls from the dataset, I began creating dummy variables to better implement the Genre, System, and Rating variables. These variables are represented by strings in the initial dataset and thus could not be used in the regressions. This change yielded 17 dummies for system, 12 for genre, and 5 for rating.

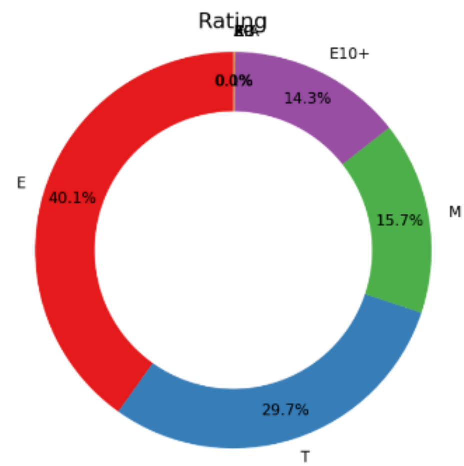
Platform Dummies: 3DS DC DS GBA GC PC PS PS2 PS3 PS4 PSP PSV Wii WiiU X360 XB XOne

Genre Dummies: Action Adventure Fighting Misc Platform Puzzle Racing Role-Playing Shooter Simulation Sports Strategy

Rating Dummies: AO E E10+ K-A M RP T







I also created a variable for game age which was made my subtracting the game's year release from 2018. Percentage of games that fall into each dummy can be seen below and full summary statistics for the final dataset can be found on the last page.

First Linear Regression

Following the completion of my data manipulation, I ran a linear regression, seen on next page, on the data. The left hand side variable was global sales and the right hand side variables were Year, Critic Score, Critic Count, User Score, User Age, Age, the system dummies, the genre dummies, and the rating dummies. X360 is the base for the system dummies, Sports is the base for the genre dummies, and T (teen) is the base for the rating dummies. The results of the regressions were mostly in line with my expectations with a few exceptions. Critic_Score had a relatively strong, positive, significant effect with a coefficient of .0244 and a t value of 10.928. User_Count was another strong positive effect which is unsurprising as a game with more user reviews more than likely has more users. I was pleased to see my inclusion of an age variable was worthwhile as it nicely shows that older games have more units sold, an unsurprising statement but given the negative coefficient attached to year, I believe it is good to have so that snap conclusions that the video game industry is declining can be avoided. The dummy variables were also consistent with my expectations with the more popular consoles, like the Wii and PlayStation 2, and genres commanding much higher significance.

OLS Regression Results

Dep. Variable:	y	R-squared:	0.192
Model:	OLS	Adj. R-squared:	0.187
Method:	Least Squares	F-statistic:	42.43
Date:	Wed, 19 Dec 2018	Prob (F-statistic):	5.78e-280
Time:	02:03:09	Log-Likelihood:	-13561.
No. Observations:	6825	AIC:	2.720e+04
Df Residuals:	6786	BIC:	2.747e+04
Df Model:	38		
Covariance Type:	nonrobust		

		coef	std err	t	P>|t|	[0.025	0.975]
	const	1.829e-05	5.07e-06	3.608	0.000	8.35e-06	2.82e-05
Year	x1	-0.0008	8.96e-05	-9.119	0.000	-0.001	-0.001
Critic_Score	x2	0.0244	0.002	10.928	0.000	0.020	0.029
Critic_Count	x3	0.0213	0.001	14.312	0.000	0.018	0.024
User_Score	x4	-0.0841	0.020	-4.182	0.000	-0.124	-0.045
User_Count	x5	0.0007	4.33e-05	16.736	0.000	0.001	0.001
Age	x6	0.0377	0.010	3.673	0.000	0.018	0.058
3DS	x7	0.2766	0.161	1.717	0.086	-0.039	0.592
DC	x8	-0.5935	0.491	-1.208	0.227	-1.556	0.369
DS	x9	0.3740	0.113	3.303	0.001	0.152	0.596
GBA	x10	-0.0667	0.155	-0.431	0.667	-0.370	0.237
GC	x11	-0.2471	0.134	-1.848	0.065	-0.509	0.015
PC	x12	-0.8282	0.104	-7.980	0.000	-1.032	-0.625
PS	x13	0.7185	0.196	3.660	0.000	0.334	1.103
PS2	x14	0.1776	0.103	1.718	0.086	-0.025	0.380
PS3	x15	0.1837	0.089	2.053	0.040	0.008	0.359
PS4	x16	-0.0042	0.143	-0.030	0.976	-0.284	0.275
PSP	x17	-0.0130	0.114	-0.114	0.909	-0.237	0.211
PSV	x18	-0.0548	0.181	-0.302	0.763	-0.410	0.301
Wii	x19	0.9258	0.106	8.726	0.000	0.718	1.134
WiiU	x20	0.0201	0.204	0.098	0.922	-0.380	0.420
XB	x21	-0.4293	0.115	-3.722	0.000	-0.655	-0.203
XOne	x22	0.1816	0.166	1.097	0.273	-0.143	0.506
Action	x23	0.0514	0.088	0.582	0.561	-0.122	0.224
Adventure	x24	-0.2036	0.136	-1.495	0.135	-0.471	0.063
Fighting	x25	0.0160	0.122	0.131	0.895	-0.223	0.255
Misc	x26	0.3223	0.112	2.883	0.004	0.103	0.541
Platform	x27	0.0509	0.110	0.463	0.643	-0.165	0.267
Puzzle	x28	-0.4110	0.179	-2.296	0.022	-0.762	-0.060
Racing	x29	0.0757	0.095	0.794	0.427	-0.111	0.262
Role-Playing	x30	-0.2177	0.104	-2.099	0.036	-0.421	-0.014
Shooter	x31	0.0289	0.103	0.280	0.779	-0.174	0.231
Simulation	x32	0.1683	0.125	1.347	0.178	-0.077	0.413
Strategy	x33	-0.2686	0.135	-1.995	0.046	-0.533	-0.005
AO	x34	0.6042	1.773	0.341	0.733	-2.871	4.080
E	x35	0.3685	0.068	5.383	0.000	0.234	0.503
E10	x36	0.0329	0.075	0.442	0.659	-0.113	0.179
K-A	x37	-0.3012	1.780	-0.169	0.866	-3.790	3.187
M	x38	0.0323	0.067	0.484	0.628	-0.099	0.163
RP	x39	1.2624	1.774	0.711	0.477	-2.216	4.741

Omnibus:	14701.741	Durbin-Watson:	0.386
Prob(Omnibus):	0.000	Jarque-Bera (JB):	132435609.946
Skew:	18.804	Prob(JB):	0.00
Kurtosis:	684.391	Cond. No.	4.15e+18

Lasso Regression

Given that I was aiming to identify the most important factors for video game sales, using a lasso to reduce the number of repressors was a natural second step after my first regression, moreover at 39 regressors not counting baselines, my model seemed like an ideal candidate to be shrunken down and made more efficient. A number of my regressors are also insignificant in my first regression, particularly among the dummies, so cutting some variables should lead to a better model overall. The large number of dummies also was a consideration when picking lasso over a ridge regression. Working with binaries, it seemed more prudent to eliminate the variables entirely if they were determined to not be useful.

My lasso eliminated a sizable number of my variables, leaving Year, Critic_Score, Critic_Count, User_Score, User_Count, DC, GBA, GC , PC, PSV, X360, Fighting, Platform, Racing, Sports, and Simulation. This means that the only non-dummy removed was Age and that all of the rating dummies were removed. These results surprised me initially, largely because of the smaller popularity, or niche nature of some of the leftover dummy variables. Most of the systems left had lower sales numbers than the ones that were removed aside from the Xbox 360 and possibly PCs but given the multiple functions of a PC this is more difficult to nail down. Additionally, many of the genres left would likely be considered niche. Fighting games generally circulate around a smaller community and strategy, racing, and simulation games are all fairly far between. This all being said, I don't see the changes made by lasso as a problem, while the systems and genres left are likely less popular, they are also less interchangeable, the PS3 and Xbox 360 are practically the same for example and thus the system effect would be smaller than from the smaller systems left. Additionally, systems with smaller numbers are possibly more influential due to being able to decrease sales more effectively. The regression below includes only the variables left over after the lasso (still maintaining the use of X360 and Sports as baselines for their respective dummy collections).

OLS Regression Results

Dep. Variable:	y	R-squared:	0.159
Model:	OLS	Adj. R-squared:	0.157
Method:	Least Squares	F-statistic:	91.90
Date:	Wed, 19 Dec 2018	Prob (F-statistic):	4.41e-243
Time:	02:03:45	Log-Likelihood:	-13698.
No. Observations:	6825	AIC:	2.743e+04
Df Residuals:	6810	BIC:	2.753e+04
Df Model:	14		
Covariance Type:	nonrobust		

		coef	std err	t	P>|t|	[0.025	0.975]
	const	65.5982	12.075	5.433	0.000	41.928	89.269
Year	x1	-0.0331	0.006	-5.513	0.000	-0.045	-0.021
Critic_Score	x2	0.0256	0.002	11.927	0.000	0.021	0.030
Critic_Count	x3	0.0166	0.001	12.303	0.000	0.014	0.019
User_Score	x4	-0.0955	0.020	-4.807	0.000	-0.134	-0.057
User_Count	x5	0.0007	4.29e-05	17.142	0.000	0.001	0.001
DC	x6	-0.9041	0.486	-1.862	0.063	-1.856	0.048
GBA	x7	-0.1639	0.124	-1.324	0.186	-0.407	0.079
GC	x8	-0.3712	0.103	-3.604	0.000	-0.573	-0.169
PC	x9	-1.1352	0.081	-14.075	0.000	-1.293	-0.977
PSV	x10	-0.3514	0.172	-2.045	0.041	-0.688	-0.015
Fighting	x11	-0.0789	0.096	-0.818	0.413	-0.268	0.110
Platform	x12	0.2156	0.094	2.286	0.022	0.031	0.400
Racing	x13	0.1502	0.079	1.893	0.058	-0.005	0.306
Simulation	x14	0.2497	0.109	2.295	0.022	0.036	0.463

Omnibus:	14706.001	Durbin-Watson:	0.349
Prob(Omnibus):	0.000	Jarque-Bera (JB):	128556956.740
Skew:	18.834	Prob(JB):	0.00
Kurtosis:	674.304	Cond. No.	1.12e+06

Cross Validation

In order to determine which of my regressions was superior, before or after the lasso, I conducted a cross validation in order to compare the MSE (Mean Squared Error) of the two models. My analysis returned a MSE of 6.715 for the pre lasso regression and a MSE of 3.166 for the post lasso regression. This is a notable drop in MSE due to the lasso regression and it leads me to conclude that the post lasso linear regression is the superior model.

Acknowledging the bias-variance tradeoff, we can understand that the original regression likely was a truer fit of the actual data. Increasing the number of regressors can be an effective way to reduce bias as confounding variables are a potential cause of increasing bias. However, although the bias of the model was low, the high number of regressors contributed to high levels of variance which lead to overfitting the data and as such, reducing the number of regressors, where appropriate, led to a decrease in the MSE. Cutting out the variables and moving to the post lasso regression will be accompanied by an increase in bias as fewer regressors are responsible for the results of the model. However, since we see that the MSE decreases with the switch, the variance of the model is decreasing to a suitable degree such that the increase in bias is outweighed. This means that the model is approaching an optimal tradeoff between bias and variance. The decreased MSE leads me to conclude that the post lasso regression, while not being as true a match to the dataset, provides results that can be better generalized and less affected by the noise in the data.

Improvements Going Forward

There is still work that can be done with this dataset, and as such I have identified some avenues for further analysis provided increased time and expertise in machine learning. I would have liked to make use of the developer and publisher data provided in the dataset. One possible idea would be to attach a prestige rating to developers and publishers based off of their reputation and see the effect of more prestigious studios on units sold. Studios like Nintendo or Rockstar could be given a high prestige rating and smaller indie studios could be given lower ones in order to quantify the information that we currently have in strings.

Also with the conclusion of my analysis the next logical step is to look into predicting sales numbers. I do not have a full prediction but my work on this project led me to produce the beginning of a random forest model which could be used as a way to predict sales or possibly classify games as hits or duds based on passing some sales mark. In fact, my random forest yielded a MSE of 2.296 after finding the number of estimators between 1 and 250 that led to the lowest MSE. 2.296 is lower than both the pre-lasso and post-lasso regressions so this would likely be a fruitful avenue to pursue.

```
rf_mse_low = 10000000

for i in range(1,251,50):
    regressor = RandomForestRegressor(n_estimators= i, random_state=0)
    regressor.fit(x_train, y_train)
    y_pred = regressor.predict(x_test)
    if  metrics.mean_squared_error(y_test, y_pred) < rf_mse_low:
        rf_mse_low = metrics.mean_squared_error(y_test, y_pred)
        ind = i
print(ind)

regressor = RandomForestRegressor(n_estimators= ind, random_state=0)
regressor.fit(x_train, y_train)
y_pred = regressor.predict(x_test)
random_forest_mse = metrics.mean_squared_error(y_test, y_pred)
print('Mean Squared Error:', random_forest_mse)
```

```
201
Mean Squared Error: 2.295686179180636
```

	Year	NA	EU	JP	Other	Global	Critic_Score	Critic_Count	User_Score	User_Count	Age
count	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000
mean	2007.436777	0.394484	0.236089	0.064158	0.082677	0.777590	70.272088	28.931136	7.185626	174.722344	10.563223
std	4.211248	0.967385	0.687330	0.287570	0.269871	1.963443	13.868572	19.224165	1.439942	587.428538	4.211248
min	1985.000000	0.000000	0.000000	0.000000	0.000000	0.010000	13.000000	3.000000	0.500000	4.000000	2.000000
25%	2004.000000	0.060000	0.020000	0.000000	0.010000	0.110000	62.000000	14.000000	6.500000	11.000000	7.000000
50%	2007.000000	0.150000	0.060000	0.000000	0.020000	0.290000	72.000000	25.000000	7.500000	27.000000	11.000000
75%	2011.000000	0.390000	0.210000	0.010000	0.070000	0.750000	80.000000	39.000000	8.200000	89.000000	14.000000
max	2016.000000	41.360000	28.960000	6.500000	10.570000	82.530000	98.000000	113.000000	9.600000	10665.000000	33.000000

	3DS	DC	DS	GBA	GC	PC	PS	PS2	PS3	PS4	PSP	PSV
count	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000
mean	0.022711	0.002051	0.067985	0.034725	0.050989	0.095385	0.021978	0.167033	0.112674	0.035018	0.057143	0.017289
std	0.148990	0.045248	0.251739	0.183097	0.219991	0.293767	0.146622	0.373033	0.316217	0.183840	0.232132	0.130357
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

	Wii	WiiU	X360	XB	XOne	Action	Adventure	Fighting	Misc	Platform	Puzzle	Racing
count	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000
mean	0.070183	0.013040	0.125714	0.082784	0.023297	0.238828	0.036337	0.055385	0.056264	0.059048	0.017289	0.085128
std	0.255474	0.113455	0.331551	0.275575	0.150855	0.426399	0.187141	0.228746	0.230447	0.235731	0.130357	0.279093
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

	Role-Playing	Shooter	Simulation	Sports	Strategy	AO	E	E10+	K-A	M	RP	T
count	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000	6825.000000
mean	0.104322	0.126593	0.043516	0.138168	0.039121	0.000147	0.305055	0.136264	0.000147	0.209963	0.000147	0.348278
std	0.305701	0.332541	0.204032	0.345102	0.193897	0.012105	0.460464	0.343094	0.012105	0.407312	0.012105	0.476460
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
75%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	0.000000	0.000000	1.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000