Finding the Best Set of Predictors for Global Sales and Comparing Predictive Power of a Statistical Model Versus a Machine Learning Model on Video Game Sales Data

Introduction

When you are making a new game, it is important to keep potential sales in mind, especially if the goal is to make money. This issue applies to any game developer looking to make a profit out of a game. This analysis is meant to show developers what kind of game they should make. It would be in their best interest to create a game with a specific genre or on a specific platform if it is shown that a certain one is shown to have historically impacted global sales. For an example, if Platform and Genre were found to be good predictors, a game developer would know exactly which platform contributes the most to global sales and what kind of game generates the most sales. If Critic score was found to be a good predictor, a game developer could research the games whose critic score is high and get a general sense of how critics rate games. They would be able to use those features and implement them into their own games to maximize their sales. There would be similar reasoning for the remaining possible predictors.

First, I aim to find the best set of variables to predict Global Sales other than Sales from the other regions. Second, I plan to compare the accuracy of predictions using statistical methods versus machine learning methods. I will use the mean squared error as the metric.

Dataset

The dataset that I will use is located on Kaggle at: https://www.kaggle.com/rush4ratio/video-game-sales-with-ratings

It combines a web scrape of VGChartz data with a web scrape from Metacritic. It acknowledges that there are null values because Metacritic only includes a subset of the platforms.

It includes the following columns:

Name – The title of the game

Platform – What platform the game is on. I.e. PC, PS2

Year_of_Release – The year the game was released

Genre – Genre of the game (Only one is listed)

Publisher – Publisher of the game

NA_Sales – Sales in North America (in millions)

EU_Sales – Sales in Europe (in millions)

JP_Sales – Sales in Japan (in millions)

Other_Sales – Sales in the rest of the world (in millions)

Global_Sales – Total worldwide sales (in millions)

Critic_score – Aggregate score compiled by Metacritic staff

Critic_count – Number of critics used in Critic_Score

User_score – Score by Metacritic's subscribers

User_count – Number of users in the User_score

Developer – Party responsible for developing the game

Rating – The ESRB ratings

I do not know of any other datasets I can use. If I were to find one, it would need to be games listed in the same range of data from the same year. If it is data used in any other year, it would not accurately be matched with the global sales found in the data set. For an example, the global sales could be 20 million higher in the 2018 data for 'Overwatch' compared to the data in 2016 (if this data point exists in the dataset). So, this makes finding a suitable dataset to combine with difficult. If there are games found outside what is available in the dataset, there would be null values in the other columns.

If another dataset is used, it would most likely be better to webscrape it from scratch.

Data Cleaning

The first cleaning step I performed was removing all null values. I checked beforehand that none of the null values came from irrelevant columns (regional sales). My question requires complete cases and for the sake of simplicity, imputation is not used. The data set is large enough for imputation to not be impactful.

I noticed some of the columns were of the wrong data type. User_Score was an object instead of a float64 value. I changed it to a float64 data type. I also changed Platform, Genre, Rating, and Publisher as a category instead of an object.

Under Platform, DC (Dreamcast) has a fairly low sample size. I combine it with the WiiU category and renamed it 'Other'.

There are many categories under the Publisher column. I found the first Publisher with less than 30 observations and then took everything below that and combined it into the 'Other' category.

The Developer column was done the same since there many distinct categories. I took the top 50 categories with the remaining ones collapsed in the 'Other' category.

Next, I created a new column which shows if the Publisher is the same value as the Developer.

I notice the Rating column has only 1 count for RP, K-A, and AO. The game tagged as AO was GTA: San Andreas. There was a controversy about a scene in that game which made it AO several years after release. Originally, it was tagged as M, so I collapse it with the M category. K-A is kids to adults, so that belongs in the E category. RP means Rating Pending. I collapse it with the T category because it is the most populated one.

I examine how many rows have Critic_Count or User_Count less than 10. These are the number of votes that contribute to Critic_Score and User_Score respectively. There are 1954 rows. I do not remove any because there are so many.

I change the year_of_release column to years_since_release. The data was collected at the end of 2016, so I assume that any game released in 2016 has been out for a year, 2015 for 2 years, etc.

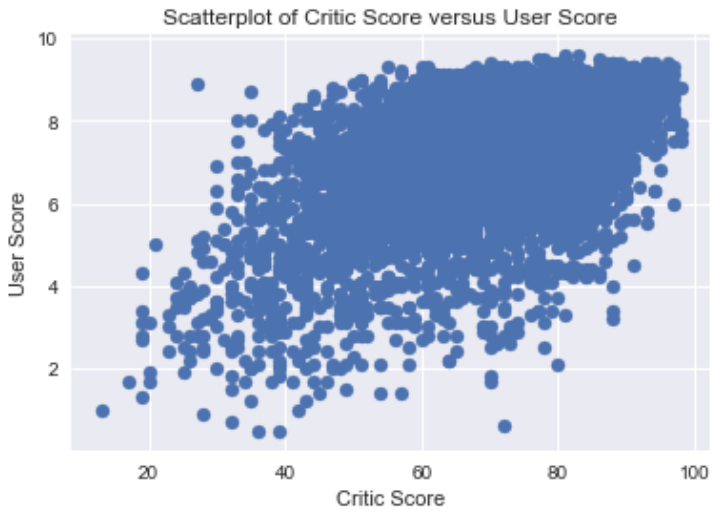
There are outliers, however, they are valid, so I do not remove them. For an example, one of the outliers is the highest Global_Sales contributor, Wii Sports. It has over 2 times the value of the 2nd highest Global_Sales. The outliers can be examined further than the analysis to see what effect they have if neccesary.

Exploratory Data Analysis

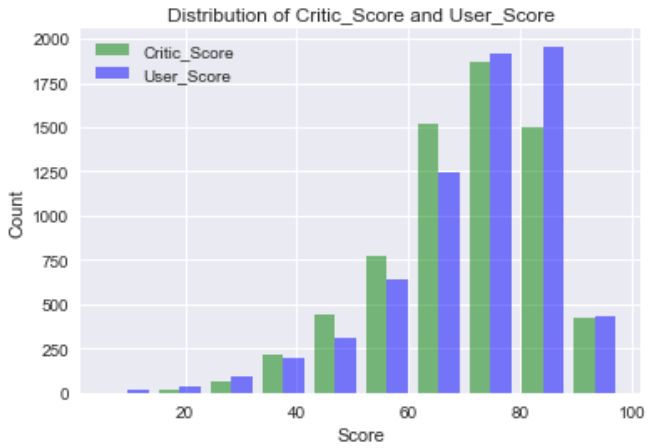
There are 16719 rows of data in the original dataset. In the cleaned data, there are 6825 complete cases. The game with the highest global Sales is Wii Sports at 82.53 million.

In the EDA, I will go through a few visualizations from comparing a variable with global sales and what it tells us. I will also use the same visualizations to look at the full dataset to see if anything changes if we had more observations in our cleaned dataset.

How does Critic Score compare with User Score?

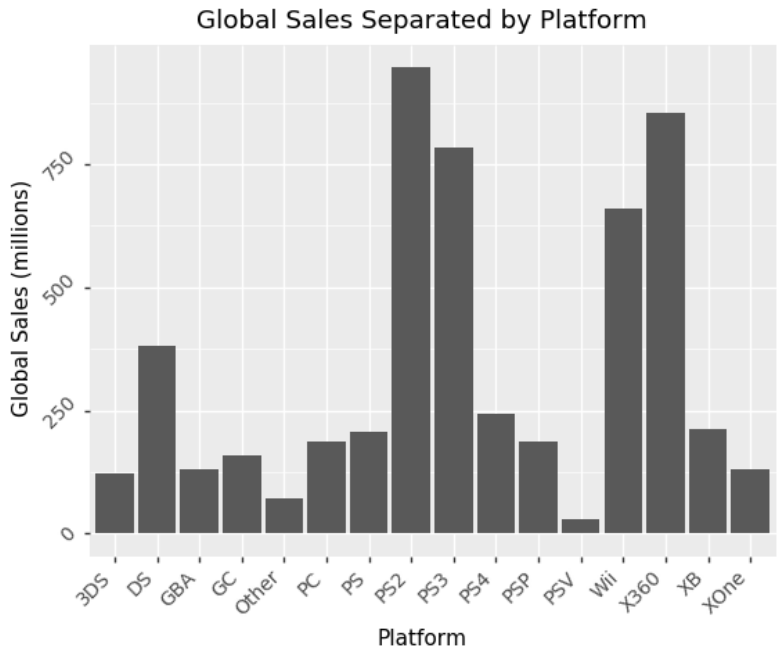


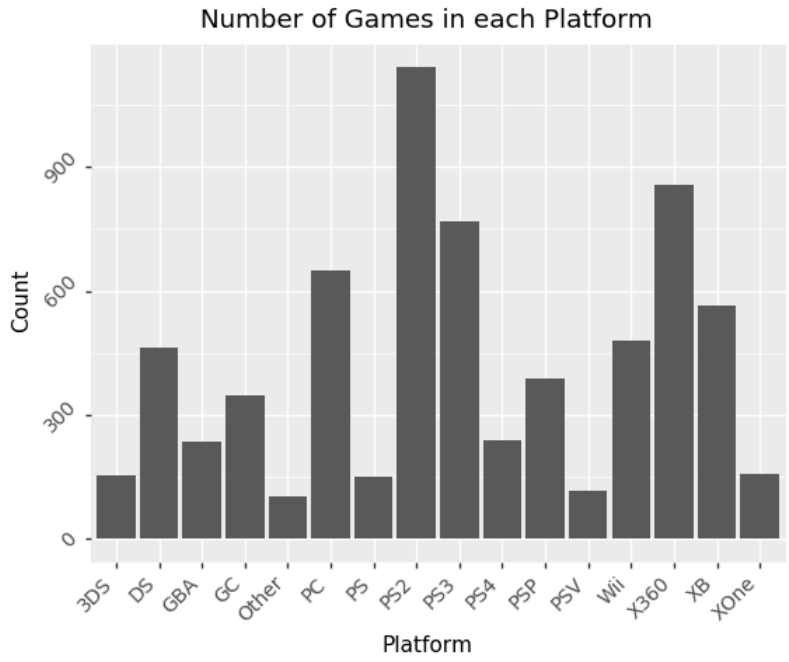
We see that there is a positive correlation between these two variables. The pearson correlation coefficient is 0.58 showing a moderate-strong positive correlation. This means that when critics rate games low, users tend to also rate games low. When critics rate games high, users tend to also rate games high.

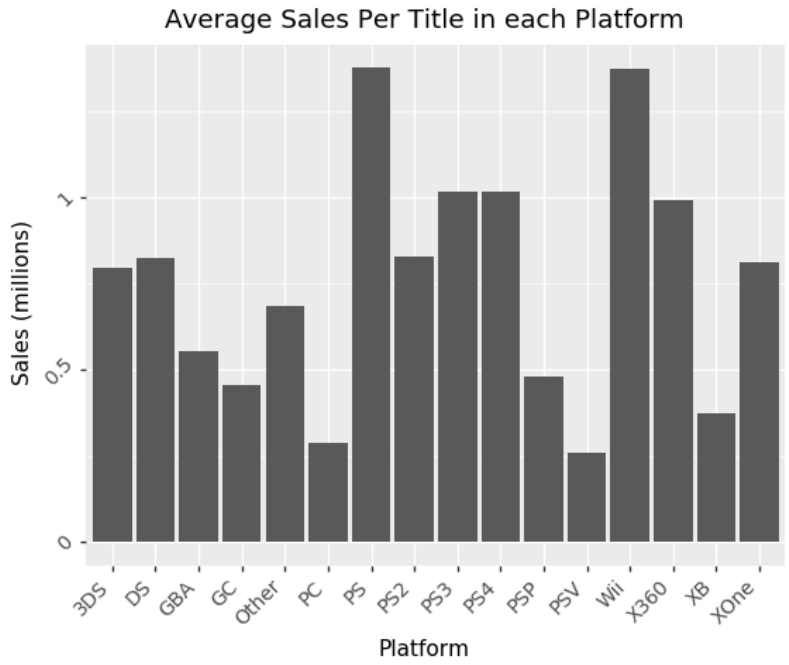


From this we see that the distribution of scores are fairly similar between the Critic group and the User group. We see a left skewed distribution for both groups. This shows that both users and critics give higher ratings more often than lower ratings.

How does each platform contribute to global sales?







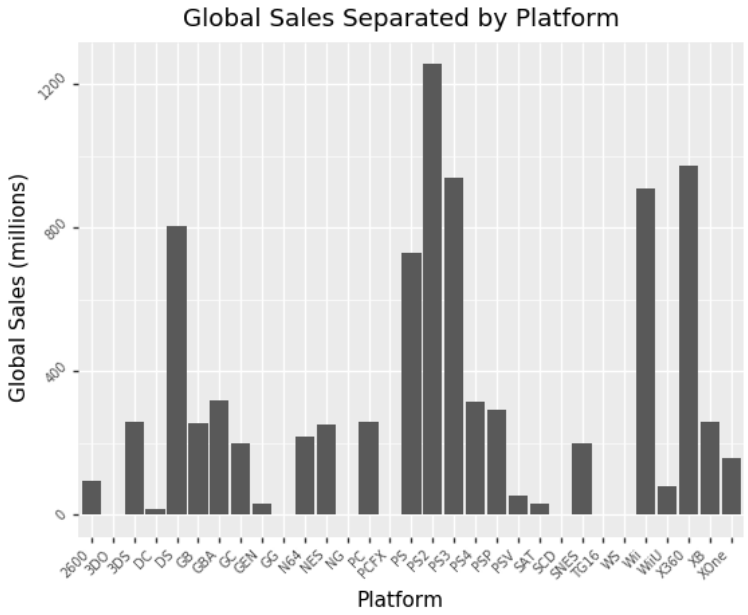
We see the lowest contributor to Global Sales is the PSV (PS Vita) and the highest platform contributor is the PS2 (Playstation 2). So despite that the Wii Sports game is the highest individual game contributor to Global Sales, PS2 games made more money than the Wii overall. This is most likely attributed to the fact that the PS2 has more titles included in the sum. The average Wii title has made more on average compared to the PS2. Part of this reason may be due to the Wii Sports game and that Wii does not have many other games in the data.

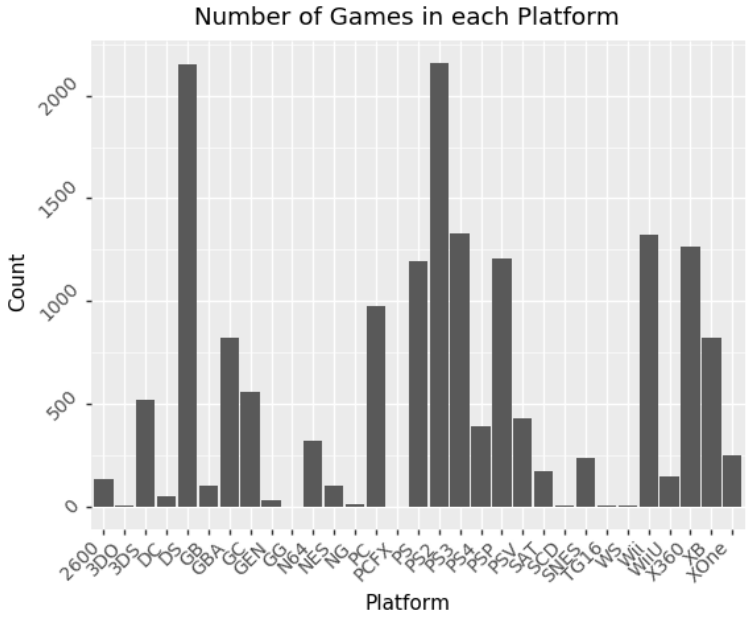
It is also interesting to see that the PS3 and PS4 has not surpassed the PS2 in sales yet, which may be due to that not enough years has passed since the release of those consoles. We see something similar happen with the Xbox 360 to the Xbox One. The Xbox 360 came out in 2005 while the Xbox One came out in 2013. The PS2 came out in 2000 and the PS3 came out in 2006. We see a six year difference for PS2 and PS3 and a 8 year difference for Xbox 360 and Xbox One. We also note that the average PS2 title has been less successful in terms of sales when compared to the average PS3 or PS4 title. Since this is the case, when there is a larger sample of PS3 and PS4 games in the data, we would see to it that PS3 and PS4 will eventually surpass the PS2 in global sales. Since PS4's count is still not as high as PS3's count, new titles added to the PS4 group will influence the average title's worth a lot more. A really successful title or several unsuccessful titles can easily influence it. Since the PS3 titles have over 3 times as many games and the average sales is about the same, we would most likely see greater growth in the global sales for PS3 games. On the other hand, the average Xbox 360 title has been more successful in terms of sales when compared to the average Xbox One title. Assuming that the sample in my data is representative of the population, the Xbox One has not been as successful in terms of sales compared to its predecessor in both total sales and average sales per title. With this trend, the Xbox One would need to have a much higher count compared to the Xbox 360 in order to surpass it in global sales.

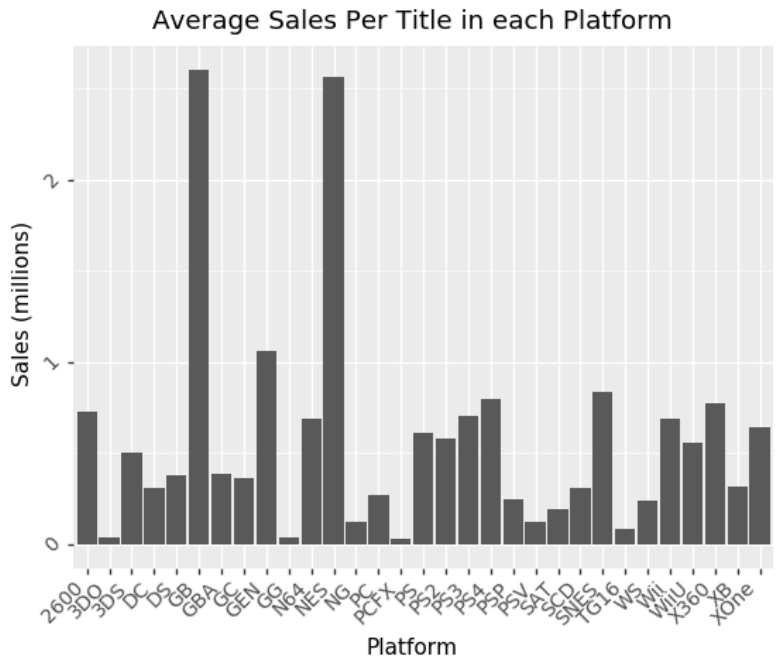
With this current dataset we cannot really investigate why there is a large gap in Global Sales between these platforms other than the lack of counts in some platforms, but we can speculate.

One possible reason for lower Global Sales in the successor systems is technology advancement. This would also apply to average sales per title in each platform. What I mean by that is that it is possible to play games without buying them. Also known as pirating. There are ways around systems to jailbreak them and be able to download the game online and burn them to a disc and then run them on the system. This would lower the sales for games and overall lower global sales. We notice that Xbox 360 and PS3 came out roughly around the same time. Xbox 360 came out about a year later. These sales are greater in PS3 than Xbox 360. They also see similar decreases in Global Sales in their successor systems. Although this is more than likely attributed to a low count especially as seen in the PS3-PS4 because they have similar average sales per title. Xbox One however, may just be not as successful. Pirating may also be a reason why the PC average sales and global sales are both fairly low. It is much easier to pirate on the PC than other platforms because you do not need to do any extra steps other than download a file. That may be why we see PC being 4th highest count in the data but have fairly low global sales. This is attributed to a low average sales per game.

Another reason is that there is missing data on some of the newer titles, hence not giving these other platforms enough observations to include more sales as seen with the PS3-PS4 case discussed earlier. In the cleaning stage, roughly 10000 observations were removed. So it could be that these are all observations that belong in the newer generation systems which we can verify.


Global Sales Separated by Platform
Global Sales (millions)
0
400
800
1200
2600
3DO
3DS
DC
DS
GB
GBA
GC
GEN
GG
N64
NES
NG
PC
PCFX
PS
PS2
PS3
PS4
PSP
PSV
SAT
SCD
SNES
TG16
WS
Wii
WiiU
X360
XB
XOne
Platform


Number of Games in each Platform
Count
0
500
1000
1500
2000
2600
3DO
3DS
DC
DS
GB
GBA
GC
GEN
GG
N64
NES
NG
PC
PCFX
PS
PS2
PS3
PS4
PSP
PSV
SAT
SCD
SNES
TG16
WS
Wii
WiiU
X360
XB
XOne
Platform



From this we see that the relative distribution of the PS2, Xbox 360, and the newer gen systems are the same for global sales. We see increases in Global Sales for systems such as the PS1 and DS which are all much older. It makes sense because users and critics are less likely to have played and rated much older games. So despite that the data of global sales were available, users and critics have not rated those games. If that is the case, those null values would be removed, which is our current dataframe. But it is also important to note that Metacritic data does not contain certain platforms, such as the original Nintendo. So this line of reasoning does not apply to those platforms.

In the average sales plot, see that with a higher count for PS4 games, the PS4 on average has been more successful in terms of sales per title compared to PS3 and the previous generations. Since we would unlikely continue to see an increase in older generation sales, the PS4 would eventually overtake the previous generations for global sales since it has the highest average per game. This can change if the newer counts of PS4 games are not as successful, so it may be too early to tell, especially because it is not much higher than the previous generations. For Xbox 360 comparing to Xbox One, we still see the same trend. Xbox 360 is more successful compared to the Xbox One in both global sales and average sales per title.

The PC average game value actually decreased in the full data set which is not too surprising. Especially because I think smaller, not as popular games are more likely to be pirated due to not wanting to spend money on a game someone may not like. It also may be more convenient. A large number of these games are single player games. There are a limited amount of multiplayer games which require players to buy the game. These games may have already been included in the cleaned data set so that including more PC games lowered the average game value due to a greater ratio of single player games compared to popular multiplayer games.

In the very old generation platforms, we see two interesting points. We note that the gameboy and the original nintendo have extremely high average sales per game values. These are the one of the few earliest systems and are regarded as classics. There was also little technology available at the time so pirating games was virtually impossible. We see the effect here on the older games have much higher sales per title than newer systems. The Super nintendo may have also been fairly successful as well, but our data just may not include it due to the nature of the website the data is collected on.

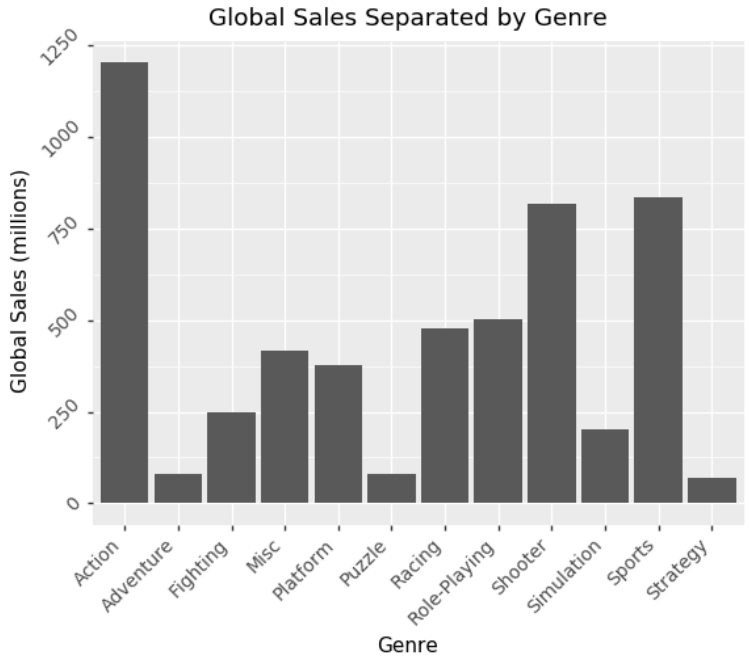
One other point sticks out from the full data set which is the number of DS games. It has fairly high global sales, but low average game sales. Each game individually did not perform well in sales most likely due to pirating and jailbreaking, but it made it up with the number of DS games.

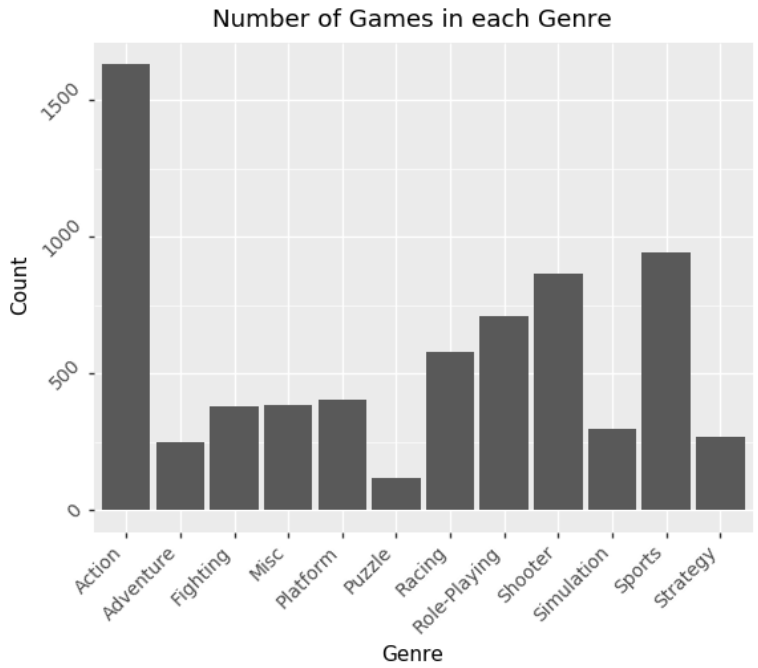
For the platforms with low count that were pooled into the 'Other' category in the cleaned data, the included platforms may have been the more popular titles on that particular platform. This may lead to a high average sales value for the 'Other' category. A closer inspection on what games were included could be insightful. The two platforms that are under 'Other' are the DC and Wii U.

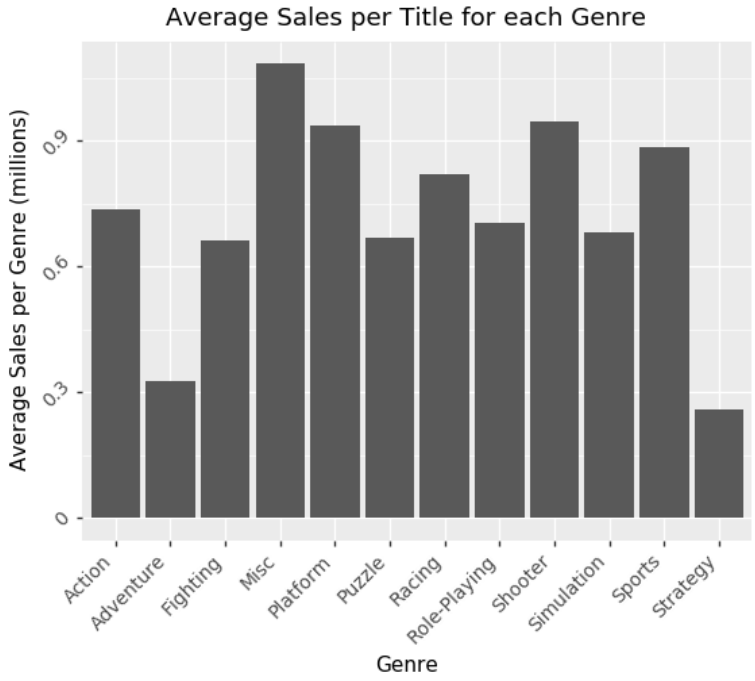
After examination, I see that from the Wii U games where we have non-null values for critic-score we have several titles >4 million in global sales. These games are contributing to an higher average global sales value for the 'Other' category. The top two contributing games are Mario Kart 8 and New Super Mario Bros U. The other high contributors are also Mario games. So it is the case that more popular titles of the Wii U games were included in the data. When the sample size of Wii U games increases, we are likely to see the average global sales value for the 'Other' category to go down. This could be what is going on in other platforms as seen in the case with Wii Sports.

Under the DC games, there are not any non-null values that are pulling up the average global sales value for the 'Other' category. The opposite could be true for the DC since the count is low. It could be that the average global sales would increase if it had more games included because it would eventually include a game whose global sales value is high.

How does each genre contribute to global sales?


Global Sales Separated by Genre
Global Sales (millions)
1250
1000
750
500
250
0
Action
Adventure
Fighting
Misc
Platform
Puzzle
Racing
Role-Playing
Shooter
Simulation
Sports
Strategy
Genre


Number of Games in each Genre
Count
1500
1000
500
0
Action
Adventure
Fighting
Misc
Platform
Puzzle
Racing
Role-Playing
Shooter
Simulation
Sports
Strategy
Genre



From this we see that the Action genre is the highest contributor to Global Sales. But this does not neccesarily mean that Action by itself is the most popular. Games are not neccesarily only one genre. It could be that Action paired with another genre contributes the most sales, but in this data set only one genre is listed for each game. For an example, Action may be frequently paired with Shooter. It is hard to imagine a shooter game that is not "action"-based.

It looks like the Action genre may just be an average of all the other categories assuming that most games are "action"-based. We can check this by averaging all the other genres.

I pooled together all the other genres except Action and pooled them together. I found the average to be 0.7284. The value for Action is 0.738135.

From this we see that is pretty much the case. The value for average sales for Action games is 0.738135 while our computed average sales for every genre except Action combined is 0.7248. This shows that it is very likely that the Action genre is including at least one of another genre.

It is the same for other genres. To analyze the data better we would need to have better data, such as all of the genres associated for each game rather than a single one.

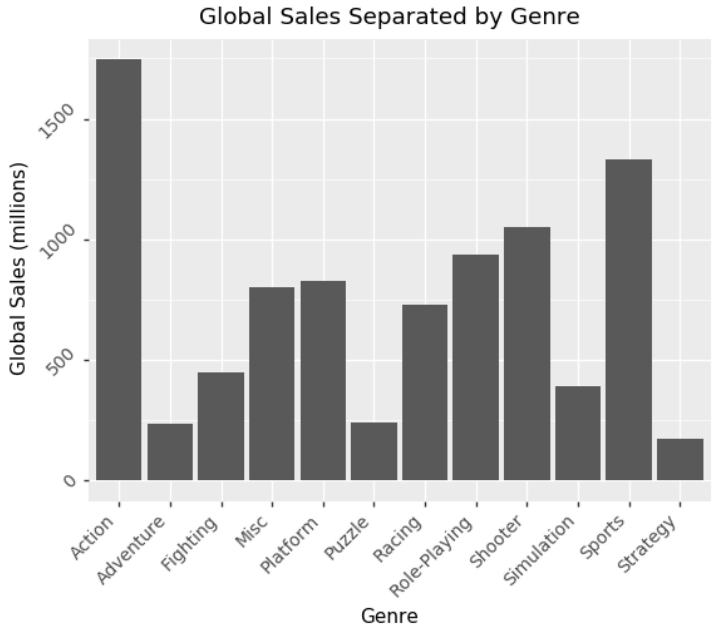
We see that the Misc genre on average has more successful games in terms of sales. The Misc genre is somewhat vague. We can examine what kind of games are under the Misc genre and its highest contributor by examining the data.

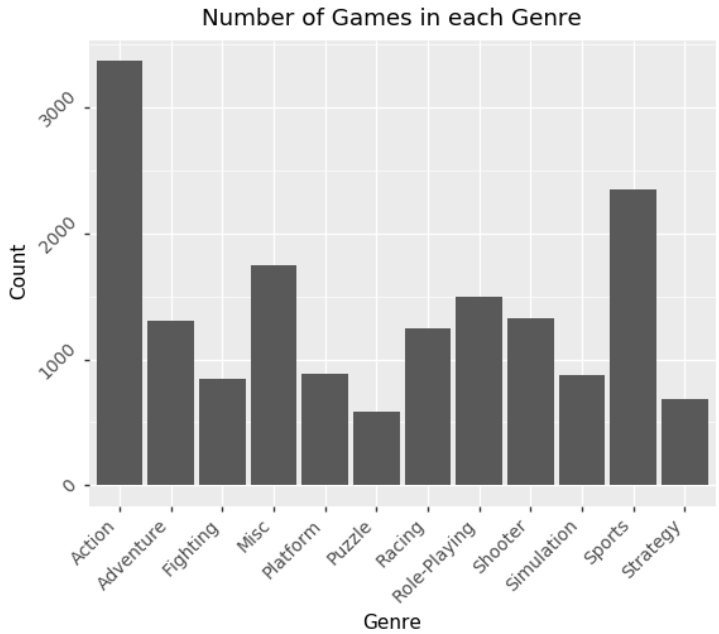
I discover that most of these titles are music games, 'Party' games which comprise of mini games, and brain academy. One of the reasons why this genre in particular has a high average sales value is because there are not as many of these types of games around so more people want to buy them. Mario party is

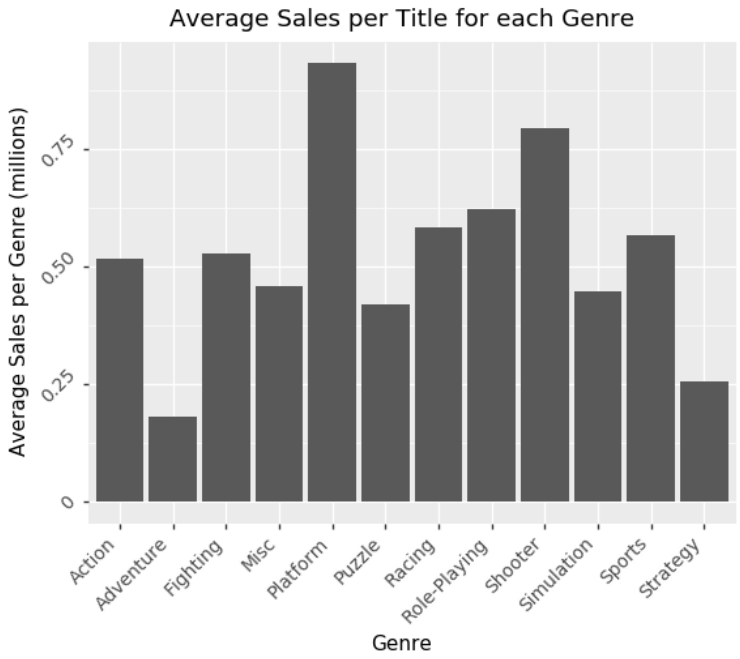
the first of its type of game and there has not been many other similar games which may cause the sales of these games to increase. Music or rhythm based games are not as saturated as other times of games either.

I also notice that the majority of these titles come from the Wii, which we saw earlier have very high average game sales attributed to mostly a few titles with high global sales. This also comes to the fact that the Wii caters to a wider audience in general compared to other systems, so it ends in more sales.

We can look at the full data set to see if anything changes with more data.







The global sales looks relatively the same with an increase in Sports and Role-Playing games. Since many null values were in the Critics Count and User Count category, it could be that these Genres are not popular on the website in the population of users and critics that use that website.

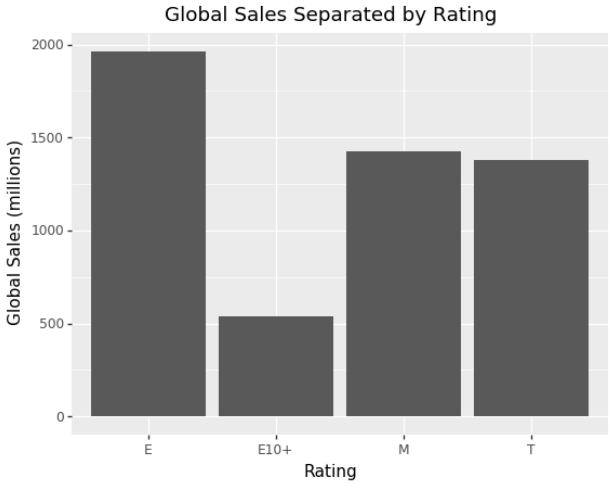
The distribution of average sales remains mostly the same but there are two notable changes. The misc genre went down and the Platform genre went up for average sales. The number of misc games increased dramatically about 5-6 times more games were included in the full data. So my earlier discussion about the presence of party games being more popular and the misc genre itself unsaturated with games was incorrect. It still could be true that these games are the more popular games and just have a large number of unsuccessful games. We would need to look at a large sample of the unsuccessful games to know for sure which is not the focus of this analysis.

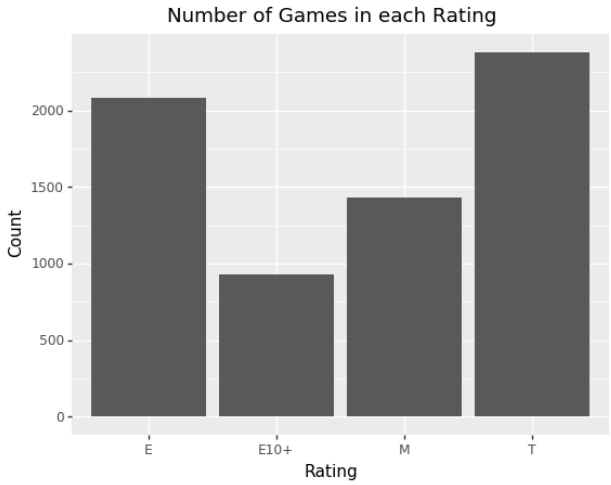
We can examine further what titles exactly are pulling the Platform genre upward. It is more than likely that these will belong to the platforms that are not represented on Metacritic.

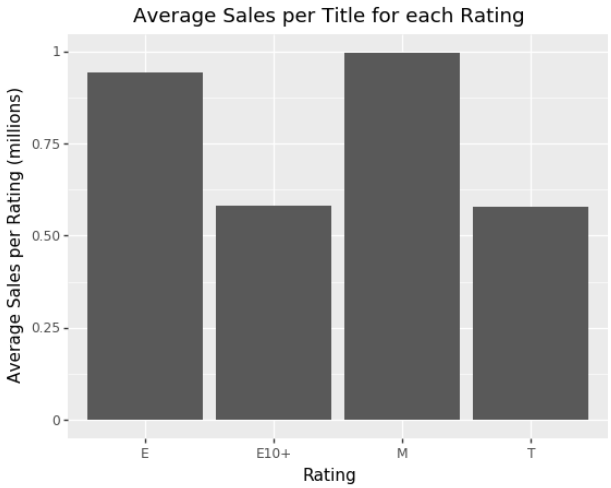
After examining the data, I discover that NES, SNES, and N64 are some of the platforms for the null values. Indeed, the Average Sales value for Platform went up because of the inclusion of the platforms not on Metacritic.

I also see that most of these games are the Mario games. Since my analysis is not focused on subsetting the data further, I will not examine this data further. It could be insightful to subset the Platforming data more so that it is separated into groups such as "Mario", "Donkey Kong", etc to understand if its the Platform genre itself having an effect or a specific franchise.

How does each rating contribute to global sales?







From this we see that the E rating contributes the most to Global Sales. This is attributed to a fairly high number of E rated games and having a high average sales value. The most saturated Rating is the "T" for

Teens rating, but since they have a low average sales value it resulted in a lower global sales compared to the E rating. Since games rated for everyone are accessible to everyone, it has a larger population buying this type of game which results in higher global sales. T and M rated games have about the same contribution to global sales despite having very different average sales values.

The highest average sales value belongs to the M rating. This could be due to that the high contributing Shooter games are rated M. Since we know that Shooter games have a fairly high average sales value it comes to no surprise that this would have an effect on the M games. These games involve more violence and gore which would make them inappropriate for some audiences. There are also less M rated games compared to E rated games. This is either due to a lack of data for M rated games or that there are just more E rated games in general which would require additional data or research.

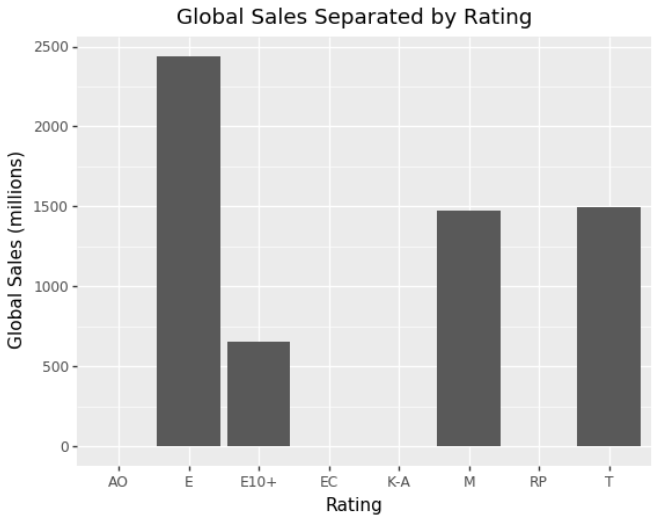
We examine some of the top contributors of the M rating to what titles are pulling it upward.

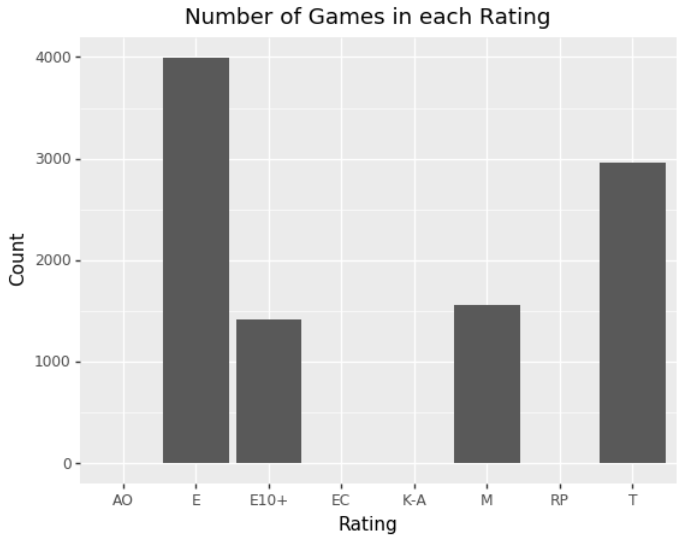
After examination, I discover that my assumptions were correct. The top 20 contributors to the M rating are all of the Shooting genre. Since we found that Action was sharing its genre with Shooting, it is no surprise to find a few games tagged as Action here such as the Grand Theft Auto series.

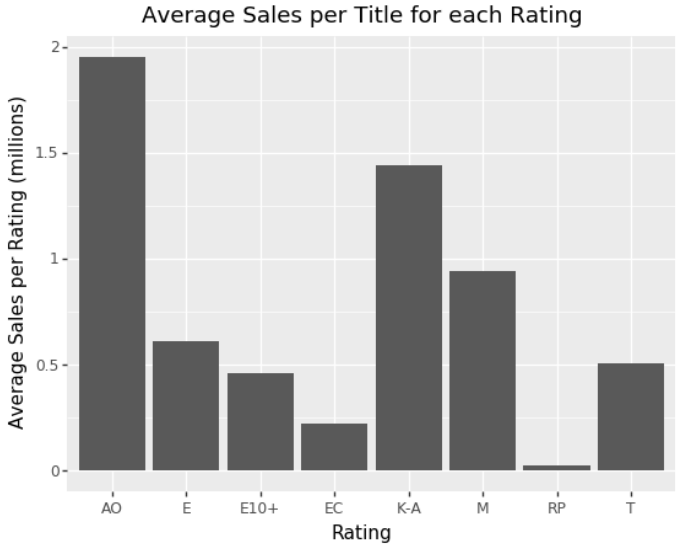
The E rated games are most likely being pulled up by Misc, Platforming, and Sports games including Wii sports since those do not involve as much violence which we can check.

I examine the data and do find that the top 20 includs those genres. The top contributors are also Wii so it lines up what we found earlier about platforms.

We can check the full data next.



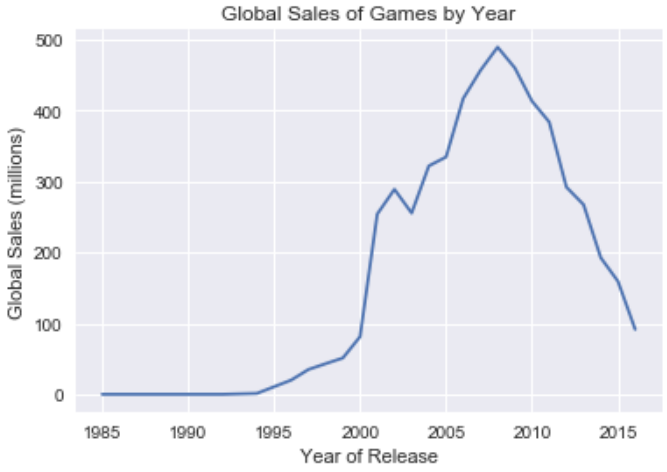


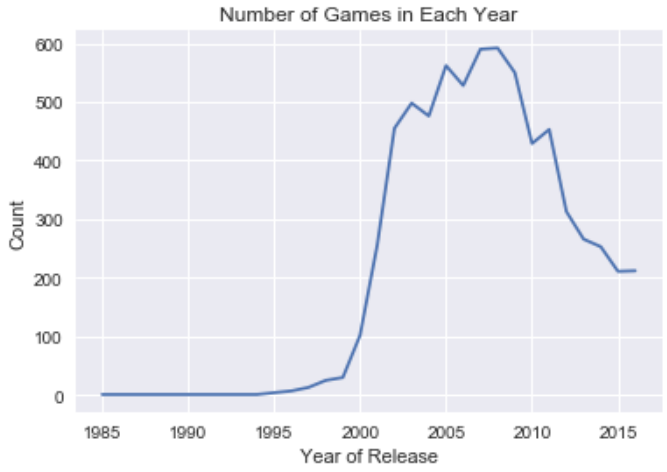


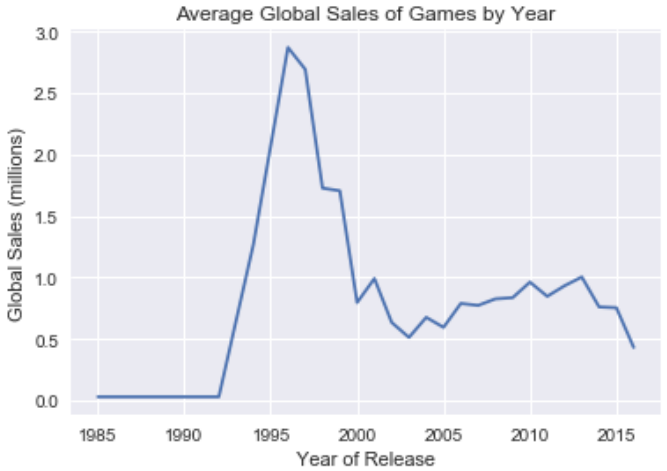
We see the same distribution for global sales and average sales in the full data. Despite having 10000 less observations, we have the same shape. This could mean that this is the "true" distribution of global sales by rating, meaning that there is a sufficient sample size in each rating we have in our clean data. So our earlier possibility of either having less M rated games compared to E rated games or just not having enough data leans more on the side of just being less M rated games in general. This means that E rated games in general will have a higher overall global sales, but M rated games tend to have more popular games because of. Assuming that high individual contributors to global sales means that it is popular.

Since there is only 1 title in AO, EC, and RP, these values are inflated, especially AO. As we know this value is GTA San Andreas which we re-categorized as M. The EC category has null values so it was dropped and was not in the clean data.

Does years since release correlate to higher global sales?



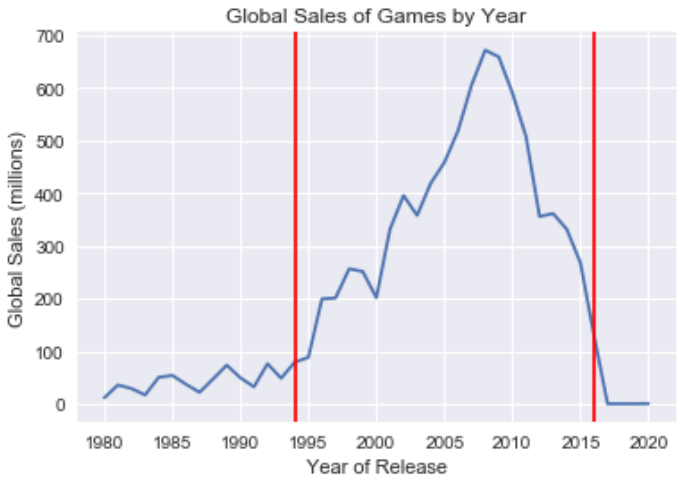


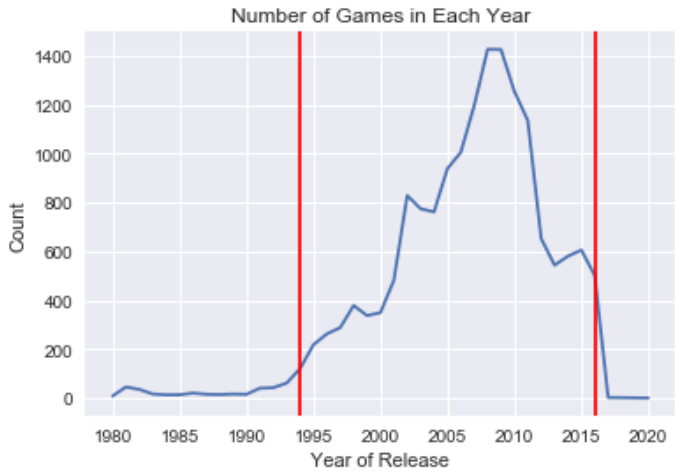


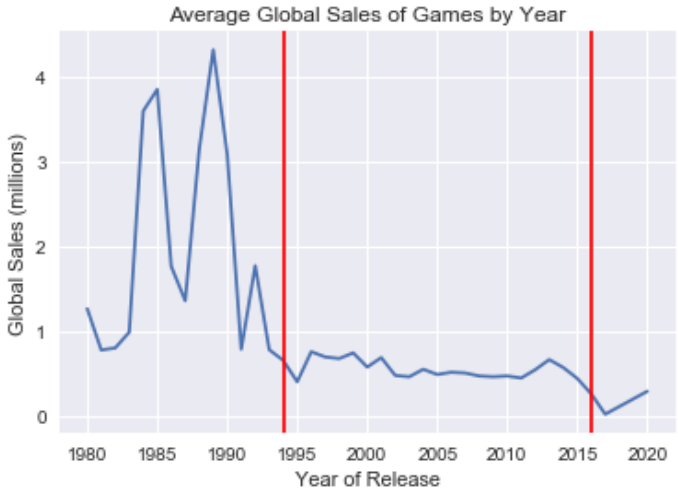
Based on total global sales alone, we see that the games released around 2007 and 2008 has seen the highest overall sales. This is also about the time where the count of games were the highest. So this peak could just be attributed to a larger abundance of data for these years. We see the average global sales of games peak around 1996. It is likely that this happened by chance due to low sample size as we can see in the number of games in each year. It was less than 100 until year 2000.

There does not seem to be any linear relationship between years since release and global sales overall or the average. If we compare the data that have enough counts from year 2000 and onward, we see a non linear relationship in overall global sales and average global sales of games seem to go down as the years progress. This means that our years_since_release variable would have a slight positive correlation. This could be attributed to the technology/pirating explanation from earlier because it is easier to pirate games and play them illegally nowadays.

It may be more insightful to look at the full data.







The vertical red lines indicate the years of data that we have that have enough data points. Looking at the full range of data, we see very inconsistent values for the average global sales before 1994.These are definitely due to those NES and GB games. For global sales, we see a steady positive increase in global sales up until 2007-2008 which is the same in our clean data set. This may indicate that games released around 8-9 years ago see the higest peak in global sales. This is also when we had the highest count of data so that may not be accurate. But it could be that a year after this data was collected, more data on games released in 2009 were made available which make it have as high as count or higher than 2007 or 2008. Without more data. If we compare our past data, it has always been the case that global sales for games were greater than the previous year up until 2007 or 2008 in general.

In the average global sales plot, we see a similar decrease in global sales as we did in our clean data that I speculate is due to the technology/pirating effect.

In general, we can't be sure that the low count or average game sales observed above are caused by unpopularity or just a lack of titles in those genres/platforms etc on the website in which the data was collected from except in the case of the Ratings because we found that the distributions of the full data set versus the clean dataset were relatively the same.

There are also a few outliers that are mistakes in the data. We have 3 2017 games and 1 2020 game in 2016 data which has to be an error, unless it counts preorders. But I find it hard to believe that preordering a game 4 years in advance is possible. We can check exactly what 4 games these are to see if there is an error in the data. Since our clean data doesn't have these points it doesn't matter for the analysis but will check nonetheless.

After comparing the names and cross checking with quick google searches, I found that they were incorrect values. Imagine:Makeup Artist was tagged as 2020 but it was released in 2009. While these data points don't matter in the analysis, the presence of these errors could hurt the analysis or our previously found results if these errors are prevalent in the data set. There's no way to know if there are more errors without cross checking every single title manually. But if there are more errors like this and

a lot of them for other information such as year of release, global sales, critic score, etc then we have more problems other than underpresented categories.

Are there correlations between our possible quantitative predictors?

I examine a correlation matrix between all the quantitative variables. Other than our previously found moderate-strong correlation between User score and Critic score, there are no other correlations between critic score, user score, critic count, user count, and years since release.

Prediction Using Statistical Methods

I split the cleaned data into 80% train, 20% test. I only use the training data while building the statistical model so that when we compare our prediction accuracy with the machine learning model, we are are using the same data.

Univariate Analysis:

I built simple linear regression models in the form:

$y = \beta_0 + \beta_1 x_1 + \varepsilon$

I used Global Sales as the response variable and the other variables one at a time. For categorical variables, I made indicator variables for them. The group with the largest sample size was made into the reference group.

I used the following hypotheses to determine whether the variable was significant:

$H_0: \beta_1 = 0$
$H_1: \beta_1 \neq 0$

I used the p value associated with the F-statistic since it is equivalent to the hypotheses above because it is the simple linear regression case. If the p value was less than 0.05, it is statistically significant and we reject that $\beta_1 = 0$. This means that Global Sales = β_0 + Variable is better than Global Sales = β_0

Variable	**P-value**	**Significant?**
Platform	~0	Yes
Genre	~0	Yes
Publisher	~0	Yes
Critic_Score	~0	Yes
Critic_Count	~0	Yes
User_Score	~0	Yes
User_Count	~0	Yes
Developer	~0	Yes
Rating	~0	Yes

Dev_same_publisher	0.245	No
Years_Since_Release	0.702	No

Building the model:

I built the multiple linear regression model in the form:

$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_i x_i + \varepsilon$, where i is the number of statistically significant predictors in the model.

Before building the model, I recoded the Publisher and Developer columns. The Publisher column has a P_ prefix and the Developer has a D_ prefix so that the 'Other' category is differentiated between these two predictors. I also prepared a new dataframe with all previously found significant variables at the univariate level.

The starting model is with all significant variables found at the univariate level and with Global_Sales as the response. I tried to remove one variable at a time to see if I can keep the reduced model. I did this through a partial-F test with the following hypotheses:

H_0: Use the reduced model
H_1: Use the full model

I compared the partial-F statistic and compared it to a critical F value with appropriate degrees of freedom for the 0.05 significance level. If the partial-F statistic was higher than the critical F value, we reject the null hypothesis that the reduced model is better. I would keep the full model.

Variable to remove	**Conclusion**
Platform	Full model
Genre	Full model
Publisher	Full model
Critic_Score	Full model
Critic_Count	Full model
User_Score	Full model
User_Count	Full model
Developer	Full model
Rating	Full model

In all the cases above, we found that the more complex model was better.

Next, I try adding Dev_same_publisher and Years_Since_Release to the model using the same test to see if the reduced model is better. It could be that the effects of these variables are significant only when these other variables are now in the model.

Variable to add	Conclusion
Dev_same_publisher	Reduced model
Years_Since_Release	Full model

I found that Dev_same_publisher was not statistically significant but Years_Since_Release was. This means that the effects of Years_Since_Release are only seen when Platform, Genre, Publisher, Critic_Score, Critic_Count, User_Score, User_Count, Developer, and Rating are in the model as well.

Before checking for interactions, I examined the VIFs(variation inflation factors). I do this before checking for interactions because after interactions are thrown in, the VIFs will naturally be high because of the interaction terms are related to each other. Another reason is that I noticed something strange in the beta coefficients. Critic_Score had a positive coefficient while User_Score had a negative coefficient. Normally you would expect that as both increase, Global_Sales would increase.

Upon examination, I discovered that the highest VIF was for Critic_Score (58.5) and the second highest was User_Score (47.1). Earlier, we found that there was a possible issue with collinearity between Critic_Score and User_Score and with the observation that User_Score unexpectedly had a negative coefficient. We remove Critic_Score from the model since it has a higher VIF.

After removal, the VIF for User_Score decreased down to 23.7, still fairly high. The only other VIF > 10 was Years_Since_Release at 15.6. But this is most likely due to the nature in the data as seen earlier in EDA. Users were most likely not rating older games, so it will be slightly correlated with Platform since older systems may be less popular. The same can be said about Years_Since_Release. Users are more likely to rate newer games, or games that have low Years_Since_Release. We saw in the correlation matrix that User_Count and Years_Since_Release had a correlation of 0.25, which is not an issue. Years_Since_Release may also be slightly correlated with Platform since older systems will all have games associated with higher Years_Since_Release.

Because there was no other highly correlated variable found between continuous variables, I will not try to remove more predictors to lower the VIFs of User_Score and Years_Since_Release. There is no issue with high VIFs caused by categorical variables in this dataset.

After the removal of Critic_Score from the model, I found that the User_Score coefficient is now positive as expected.

Checking For Interactions:

Since checking every interaction would take too long, I only checked a few that made sense. It made sense to think that the User_Count can interact with User_Score. It also made sense for Years_Since_Release to interact with User_Count and Critic_Count because the longer a game has been out, the more likely a user or critic has given a game a score. It also made sense that the Critic_Count can interact with Rating. Critics may be more likely to focus on a certain Rating. I tried adding each of

these interactions one at a time and used the P-value of the interaction or the partial F statistic as before to determine to keep the reduced model or full model. For single interaction terms, I used the p-value. For Critic_Count * Rating, I used the partial F statistic.

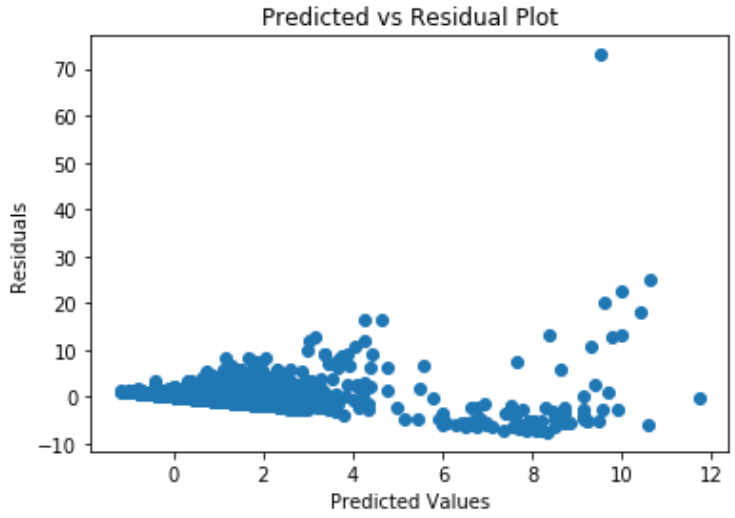
Interaction	**Conclusion**
User_Score * User_Count	Reduced Model
Critic_Count * Years_Since_Release	Full model
User_Count * Years_Since_Release	Reduced model
Critic_Count * Rating	Full model

Our current model is:

Global Sales = β_0 + Platform + Genre + Publisher + Critic_Count + User_Score + User_Count + Developer + Rating + Critic_Count * Years_Since_Release + Critic_Count * Years_Since_Release + Critic_Count * Rating (Betas are not written out due to the sheer length)

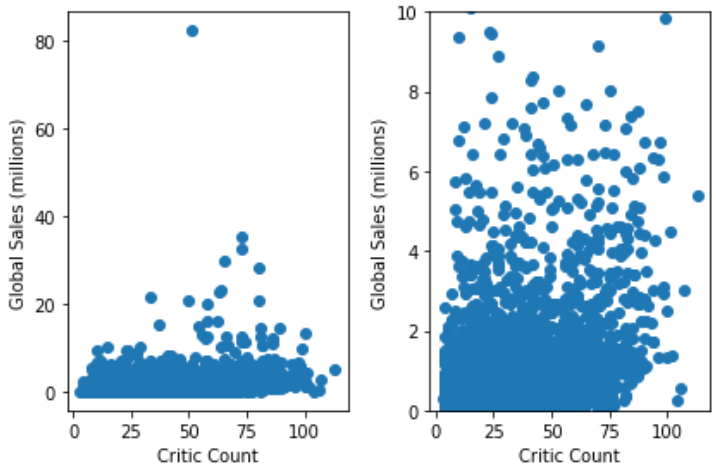
Model Diagnostics:

Predicted vs Residual Plot



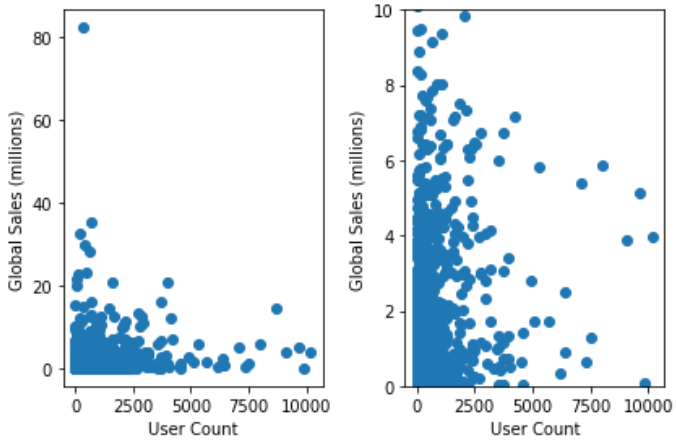
Above we see that homoscedasticity is clearly violated. Note that 43% of the points are above 0.

Critic Count vs Global Sales



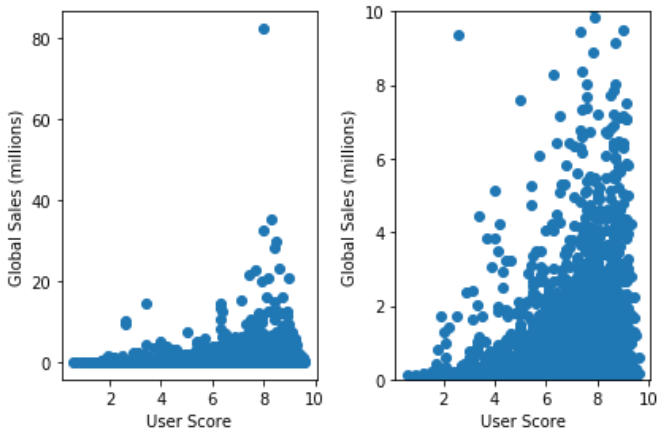
Above we do not see anything unusual here. Critic Count and Global Sales appear to have a linear relationship.

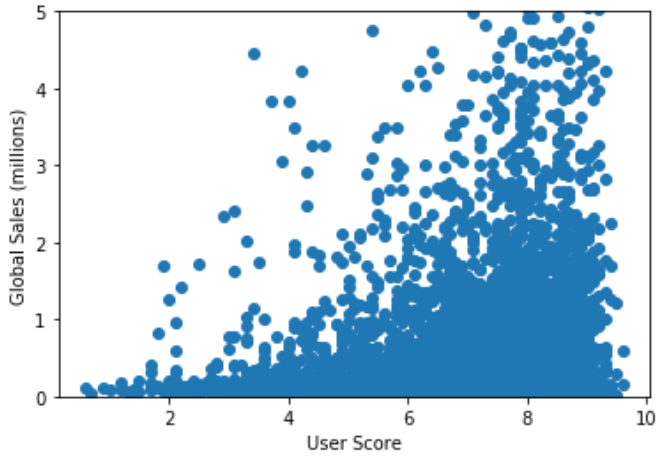
User Count vs Global Sales



User Count appears to have a linear relationship with Global Sales.

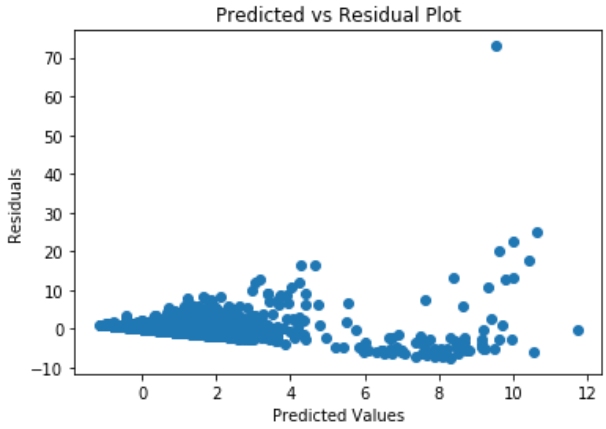
User Score vs Global Sales





Above we see that User Score does not have a linear relationship with Global Sales. I add a quadratic term to the model to compensate for this.

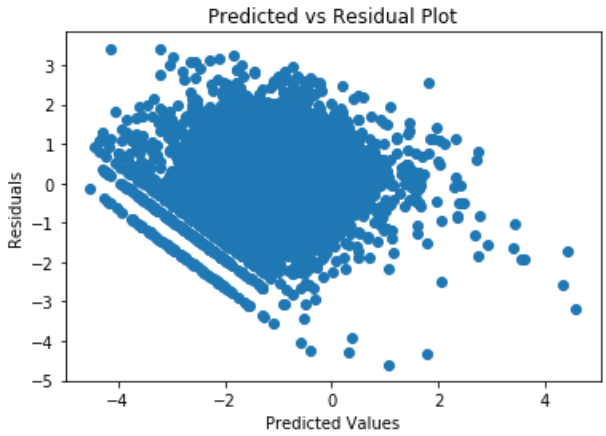
<u>Predicted vs Residual Plot With Quadratic Term</u>



Adding the quadratic term did not remedy the non-constant variance.

Next, I tried a box-cox transformation.

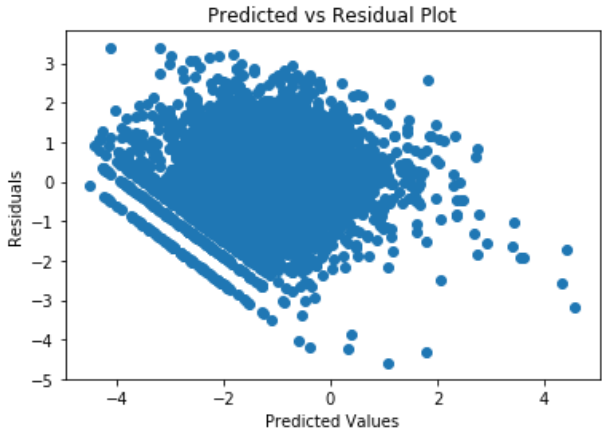
Predicted vs Residual Plot With Box-cox



The residuals here look a bit better. The R^2 value also increased to 0.524 from 0.317. A R^2 value of 0.524 means that 52.4% of the variation observed in Global_Sales is explained by the predictors in the model. But we still do not have the constant variance assumption met.

Next, we try transforming Global Sales with a natural log.

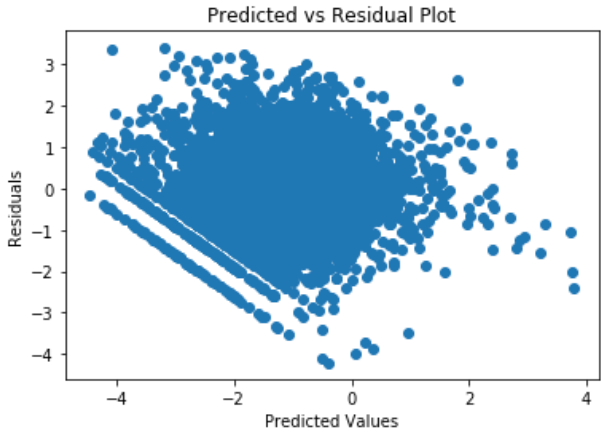
Predicted vs Residual Plot With Natural Log



The R^2 value here is 0.525, very slightly better than our box-cox transformation. Our residuals look relatively the same. Our overall F statistic is a bit higher in our natural log transformation as well, so the natural log transformation is very slightly better than the box-cox overall.

Next, we try weighted regression with the natural log transformed response variable. I used the method described at https://stats.stackexchange.com/questions/97832/how-do-you-find-weights-for-weighted-least-squares-regression to determine the weights.

Predicted vs Residual Plot With Weighted Regression



Our residuals did not improve and our R^2 value is 0.512. I scrap this model and go back to the log transformation model.

I tried many other different methods to remedy the non-constant variance but nothing worked. This included things such as transforming the predictor variables, removing a combination of different predictor variables, and recoding Developer/Publisher/Platform to have less categories.

Since I was not able to remedy the non-constant variance assumption, I do not check for the normality of the residuals or for influential points or outliers. It makes no sense to do those things when the model is invalid regardless.

I call the current natural log transformation model, Model 1.

Model 1:

ln(Global Sales) = β_0 + Critic_Count + User_Score + User_Count + Platform + Genre + Publisher + Rating + Developer + * Years_Since_Release + Critic_Count * Years_Since_Release + Critic_Count * Rating + User_Score^2

Parameters for Model 1:

	coeff	std err	t	P>|t|	[0.025	0.975]
const	-2.4574	0.213	-11.545	0.000	-2.875	-2.040
Critic_Count	0.0200	0.002	9.472	0.000	0.016	0.024
User_Score	0.0227	0.059	0.384	0.701	-0.093	0.138
User_Count	0.0005	2.87e-05	18.579	0.000	0.000	0.001
3DS	-0.5010	0.110	-4.546	0.000	-0.717	-0.285

DS	-0.3019	0.069	-4.354	0.000	-0.438	-0.166
GBA	-0.4238	0.087	-4.857	0.000	-0.595	-0.253
GC	-0.7299	0.070	-10.483	0.000	-0.866	-0.593
Other	-0.9559	0.125	-7.628	0.000	-1.202	-0.710
PC	-1.9914	0.068	-29.235	0.000	-2.125	-1.858
PS	0.6634	0.103	6.431	0.000	0.461	0.866
PS3	-0.0248	0.066	-0.378	0.705	-0.153	0.104
PS4	-0.7349	0.106	-6.916	0.000	-0.943	-0.527
PSP	-0.3933	0.070	-5.658	0.000	-0.530	-0.257
PSV	-0.6768	0.119	-5.672	0.000	-0.911	-0.443
Wii	0.0131	0.068	0.192	0.848	-0.121	0.147
X360	-0.4737	0.065	-7.288	0.000	-0.601	-0.346
XB	-0.8706	0.059	-14.846	0.000	-0.986	-0.756
XOne	-0.4338	0.118	-3.667	0.000	-0.666	-0.202
Adventure	-0.3949	0.076	-5.193	0.000	-0.544	-0.246
Fighting	0.1953	0.072	2.726	0.006	0.055	0.336
Misc	0.3193	0.067	4.744	0.000	0.187	0.451
Puzzle	-0.3447	0.107	-3.215	0.001	-0.555	-0.134
Racing	0.1348	0.060	2.232	0.026	0.016	0.253
Role-Playing	0.0513	0.054	0.949	0.343	-0.055	0.157

Shooter	-0.0117	0.049	-0.240	0.811	-0.108	0.084
Simulation	0.3640	0.077	4.756	0.000	0.214	0.514
Sports	0.0327	0.059	0.552	0.581	-0.083	0.149
Strategy	-0.2625	0.076	-3.474	0.001	-0.411	-0.114
P_505 Games	0.0885	0.128	0.689	0.491	-0.163	0.340
P_Acclaim Entertainment	0.3553	0.176	2.017	0.044	0.010	0.701
P_Activision	0.8033	0.070	11.462	0.000	0.666	0.941
P_Atari	0.4241	0.093	4.577	0.000	0.242	0.606
P_Bethesda Softworks	1.0735	0.169	6.344	0.000	0.742	1.405
P_Capcom	-0.0192	0.109	-0.177	0.860	-0.233	0.194
P_Codemasters	-0.0071	0.159	-0.045	0.964	-0.318	0.304
P_D3Publisher	0.0345	0.184	0.187	0.852	-0.327	0.396
P_Deep Silver	-0.0323	0.146	-0.222	0.824	-0.318	0.253
P_Disney Interactive Studios	0.8153	0.138	5.919	0.000	0.545	1.085
P_Eidos Interactive	0.0912	0.104	0.881	0.378	-0.112	0.294
P_Electronic Arts	0.9890	0.069	14.399	0.000	0.854	1.124
P_Focus Home Interactive	0.2749	0.198	1.387	0.165	-0.114	0.663
P_Ignition Entertainment	-0.6545	0.224	-2.928	0.003	-1.093	-0.216
P_Konami Digital Entertainment	-0.0422	0.104	-0.406	0.685	-0.246	0.161
P_LucasArts	1.3131	0.145	9.071	0.000	1.029	1.597

P_Microsoft Game Studios	0.7183	0.107	6.683	0.000	0.508	0.929
P_Midway Games	0.0058	0.145	0.040	0.968	-0.279	0.291
P_Namco Bandai Games	0.1389	0.085	1.635	0.102	-0.028	0.306
P_Nintendo	1.0852	0.091	11.954	0.000	0.907	1.263
P_Nippon Ichi Software	-0.2827	0.149	-1.900	0.057	-0.574	0.009
P_Rising Star Games	-0.2414	0.156	-1.548	0.122	-0.547	0.064
P_Sega	0.3320	0.085	3.919	0.000	0.166	0.498
P_Sony Computer Entertainment	0.3626	0.078	4.663	0.000	0.210	0.515
P_Square Enix	0.6374	0.113	5.618	0.000	0.415	0.860
P_THQ	0.6824	0.080	8.482	0.000	0.525	0.840
P_Take-Two Interactive	0.7154	0.082	8.730	0.000	0.555	0.876
P_Tecmo Koei	-0.2931	0.130	-2.254	0.024	-0.548	-0.038
P_Ubisoft	0.4241	0.075	5.662	0.000	0.277	0.571
P_Vivendi Games	0.3004	0.120	2.499	0.012	0.065	0.536
P_Warner Bros. Interactive Entertainment	1.1260	0.124	9.067	0.000	0.883	1.369
E	0.1190	0.066	1.801	0.072	-0.011	0.248
E10+	0.0093	0.081	0.115	0.909	-0.150	0.168
M	0.1793	0.079	2.264	0.024	0.024	0.335
D_Acclaim	0.1072	0.287	0.373	0.709	-0.456	0.670
D_Arc System Works	-0.2279	0.203	-1.123	0.261	-0.626	0.170

D_Artificial Mind and Movement	0.2228	0.215	1.037	0.300	-0.198	0.644
D_BioWare	-0.3381	0.241	-1.405	0.160	-0.810	0.133
D_Capcom	0.5799	0.126	4.592	0.000	0.332	0.827
D_Climax Group	-0.3222	0.224	-1.439	0.150	-0.761	0.117
D_Codemasters	0.2879	0.213	1.355	0.176	-0.129	0.705
D_Criterion Games	-0.1911	0.243	-0.787	0.432	-0.667	0.285
D_CyberConnect2	0.5830	0.246	2.370	0.018	0.101	1.065
D_EA Canada	-0.0766	0.111	-0.689	0.491	-0.295	0.141
D_EA DICE	0.3973	0.239	1.662	0.097	-0.071	0.866
D_EA Games	-0.0064	0.227	-0.028	0.978	-0.451	0.439
D_EA Sports	0.2777	0.117	2.374	0.018	0.048	0.507
D_EA Tiburon	0.1516	0.139	1.088	0.276	-0.121	0.425
D_Electronic Arts	0.1678	0.147	1.139	0.255	-0.121	0.457
D_Eurocom Entertainment Software	0.1388	0.192	0.721	0.471	-0.238	0.516
D_Exient Entertainment	-0.4101	0.228	-1.799	0.072	-0.857	0.037
D_From Software	-0.3309	0.179	-1.853	0.064	-0.681	0.019
D_Gearbox Software	0.3375	0.231	1.461	0.144	-0.115	0.790
D_Griptonite Games	0.1662	0.249	0.667	0.505	-0.323	0.655
D_Harmonix Music Systems	0.4720	0.215	2.198	0.028	0.051	0.893
D_High Voltage Software	0.2142	0.214	1.002	0.316	-0.205	0.633

D_KCET	0.6399	0.251	2.549	0.011	0.148	1.132
D_Koei	0.0869	0.225	0.386	0.700	-0.354	0.528
D_Konami	0.6259	0.149	4.201	0.000	0.334	0.918
D_Krome Studios	-0.2214	0.237	-0.934	0.350	-0.686	0.243
D_Maxis	0.6620	0.181	3.665	0.000	0.308	1.016
D_Midway	0.6236	0.214	2.913	0.004	0.204	1.043
D_Namco	0.2677	0.149	1.798	0.072	-0.024	0.560
D_Neversoft Entertainment	0.6216	0.196	3.168	0.002	0.237	1.006
D_Nintendo	1.1405	0.150	7.609	0.000	0.847	1.434
D_Omega Force	0.6866	0.165	4.173	0.000	0.364	1.009
D_Radical Entertainment	0.2147	0.205	1.047	0.295	-0.187	0.617
D_Rainbow Studios	0.3203	0.220	1.458	0.145	-0.110	0.751
D_Rebellion	-0.4742	0.194	-2.444	0.015	-0.855	-0.094
D_SCEA San Diego Studios	0.3606	0.230	1.565	0.118	-0.091	0.812
D_Sega	0.1230	0.213	0.579	0.563	-0.294	0.540
D_Sonic Team	0.7215	0.192	3.755	0.000	0.345	1.098
D_Square Enix	0.2225	0.195	1.139	0.255	-0.161	0.606
D_TOSE	0.6100	0.202	3.024	0.003	0.214	1.006
D_TT Games	0.4355	0.215	2.023	0.043	0.013	0.858
D_Traveller's Tales	0.4972	0.162	3.063	0.002	0.179	0.815

D_Treyarch	0.5529	0.180	3.079	0.002	0.201	0.905
D_Ubisoft	0.6685	0.127	5.284	0.000	0.420	0.916
D_Ubisoft Montreal	0.6885	0.134	5.146	0.000	0.426	0.951
D_Ubisoft Shanghai	0.1291	0.247	0.523	0.601	-0.355	0.613
D_Vicarious Visions	0.6152	0.160	3.845	0.000	0.302	0.929
D_Visual Concepts	0.7141	0.157	4.540	0.000	0.406	1.022
D_Volition Inc.	0.1526	0.255	0.600	0.549	-0.346	0.652
D_Yuke's	0.6407	0.180	3.559	0.000	0.288	0.994
Years_Since_Release	-0.0117	0.008	-1.425	0.154	-0.028	0.004
Critic_Count_Years	0.0004	0.000	2.025	0.043	1.26e-05	0.001
Critic_Count_E	0.0042	0.002	2.075	0.038	0.000	0.008
Critic_Count_E10+	-0.0005	0.002	-0.189	0.850	-0.005	0.004
Critic_Count_M	-0.0017	0.002	-0.915	0.360	-0.005	0.002
User_Score_Squared	0.0032	0.005	0.692	0.489	-0.006	0.012

Coefficient Interpretation:

In our first model, our statistically significant predictors are Critic_Count, User_Score, User_Score^2, User_Count, Platform, Genre, Publisher, Rating, Developer, Years_Since_Release, Critic_Count * Years_Since_Release, and Critic_Count * Rating.

For coefficient interpretation of these variables, we have to take into account any interaction or squared term simultaneously.

Critic_Count:

With everything else held constant and for a T Rated game, for every one unit increase in Critic_Count, we expect the Global_Sales to increase by $e^{0.02+0.0004} - 1 = 2.06\%$.

User_Score and User_Score^2:

Interpreting the meaning of the User_Score coefficient is difficult due to the presence of the User_Score_Squared term. If the User_Score of a game is 5, the contribution to Global_Sales is $e^{0.0227*5 + 0.0032*25} - 1 = e^{0.1935} - 1 = 21.35\%$ more than if User_Score of a game is 0. With everything else held constant, for a one unit increase to a User_Score of 6 from a User_Score of 5, we expect an increase of $e^{(0.0227*6 + 0.0032*36)} - 1.21 = 7.58\%$ for Global_Sales.

User_Count:

With everything else held constant, for every one unit increase in User_Count, we expect Global_Sales to increase by $e^{0.0005} - 1 = 0.05\%$.

Platform:

With everything else held constant, if the platform of the game is 3DS, the Global_Sales of that game is expected to be $1 - e^{-0.5010} = 39.41\%$ less compared to a PS2 game.

Genre:

With everything else held constant, if the genre of the game is Adventure, the Global_Sales of that game is expected to be $1 - e^{-0.3949} = 32.63\%$ less compared to an Action game.

Publisher:

With everything else held constant, if the Publisher of the game is 505 Games, the Global_Sales of that game is expected to be $e^{0.0885} - 1 = 9.25\%$ more compared to the P_Other group. More details on what is in the Other group is in the data cleaning notebook. A game published by Lucas Arts increases the Global_Sales value the most ($e^{1.3131} - 1 = 171.77\%$ more than Other). A game published by Ignition Entertainment decreases the Global_Sales value the most ($1 - e^{-0.6545} = 48.03\%$ less than Other).

Rating:

With everything else held constant, if the Rating of the game is E, the Global_Sales of that game is expected to be $e^{0.1190 + 0.0042} - 1 = 13.06\%$ more compared to a T Rated game.

Developer:

With everything else held constant, if the Developer of the game is Acclaim, the Global_Sales of that game is expected to be $e^{0.1072}$ - 1 = 11.32% more compared to the D_Other group. More details on what is in the Other group is in the data cleaning notebook. A game developed by Nintendo increases Global_Sales value the most ($e^{1.1405}$ - 1 = 212.83% more than Other). This is not surprising because Wii Sports was made by Nintendo which has the highest Global_Sales value. A game developed by Rebellion decreases Global_Sales value the most (1 - $e^{-0.4742}$ = 37.76% less than Other).

Years_Since_Release:

With everything else held constant, for one unit increase in Years_Since_Release, the Global_Sales of that game is expected to decrease by 1 - $e^{-0.0117 + 0.0004}$ = 1.12%. We notice that this is opposite of the sign expected. It makes more sense that the longer a game has been out, the more Global_Sales it would have. Our first thought is that this might be due to a collinearity problem. We probably should have removed Years_Since_Release from the model due to the nature of the data. We will do this in Model 2 and 3.

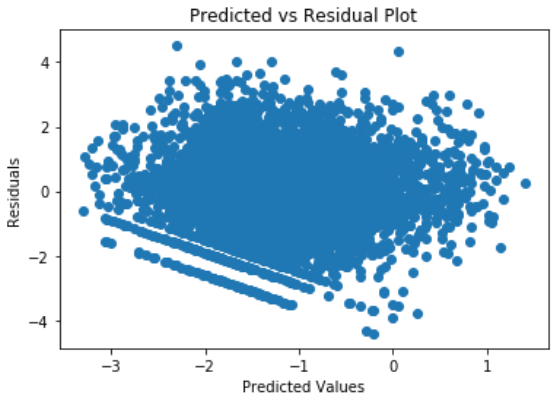
Model 2:

$\ln(\text{Global\_Sales}) = \beta_0 + \text{Critic\_Count} + \text{User\_Score} + \text{Genre} + \text{Rating} + \text{User\_Score}^2$

Parameters for Model 2:

	coeff	std err	t	P>|t|	[0.025	0.975]
const	-2.4990	0.224	-11.158	0.000	-2.938	-2.060
Critic_Count	0.0332	0.001	36.351	0.000	0.031	0.035
User_Score	-0.0708	0.070	-1.010	0.313	-0.208	0.067
Adventure	-0.5857	0.091	-6.437	0.000	-0.764	-0.407
Fighting	0.2090	0.079	2.656	0.008	0.055	0.363
Misc	0.3538	0.077	4.607	0.000	0.203	0.504
Puzzle	-0.6185	0.128	-4.848	0.000	-0.869	-0.368
Racing	-0.0567	0.067	-0.846	0.398	-0.188	0.075
Role-Playing	-0.1745	0.059	-2.941	0.003	-0.291	-0.058

Shooter	-0.0684	0.057	-1.192	0.233	-0.181	0.044
Simulation	0.0679	0.084	0.805	0.421	-0.097	0.233
Sports	0.1673	0.060	2.797	0.005	0.050	0.285
Strategy	-1.0314	0.087	-11.846	0.000	-1.202	-0.861
E	0.4626	0.048	9.595	0.000	0.368	0.557
E10+	0.2910	0.055	5.338	0.000	0.184	0.398
M	0.0321	0.049	0.652	0.515	-0.064	0.128
User_Score_Squared	0.0125	0.005	2.281	0.023	0.002	0.023



The residuals in this model look better in this one. This is the "best" statistical model I could find, trading between residuals and R^2 value (0.258).

Model 3:

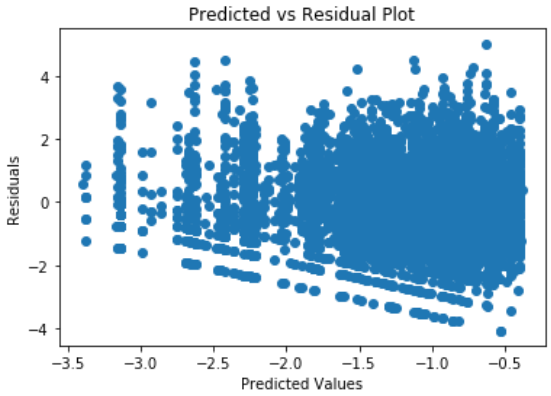
ln(Global_Sales) = β_0 + Platform + Genre + Rating

Parameters for Model 3:

	coeff	std err	t	P>|t|	[0.025	0.975]
const	-1.1408	0.056	-20.421	0.000	-1.250	-1.031

3DS	-0.1346	0.122	-1.107	0.268	-0.373	0.104
DS	-0.1935	0.082	-2.352	0.019	-0.355	-0.032
GBA	-0.3790	0.107	-3.536	0.000	-0.589	-0.169
GC	-0.4817	0.088	-5.472	0.000	-0.654	-0.309
Other	-0.3123	0.147	-2.125	0.034	-0.601	-0.024
PC	-1.5326	0.072	-21.178	0.000	-1.674	-1.391
PS	0.2672	0.122	2.199	0.028	0.029	0.505
PS3	0.2969	0.067	4.401	0.000	0.165	0.429
PS4	-0.1736	0.102	-1.699	0.089	-0.374	0.027
PSP	-0.3908	0.085	-4.622	0.000	-0.557	-0.225
PSV	-0.7546	0.134	-5.627	0.000	-1.018	-0.492
Wii	0.1951	0.079	2.475	0.013	0.041	0.350
X360	0.1161	0.065	1.775	0.076	-0.012	0.244
XB	-0.6901	0.074	-9.298	0.000	-0.836	-0.545
XOne	-0.0997	0.123	-0.810	0.418	-0.341	0.142
Adventure	-0.7017	0.096	-7.272	0.000	-0.891	-0.513
Fighting	0.0362	0.083	0.434	0.664	-0.127	0.200
Misc	0.1970	0.082	2.414	0.016	0.037	0.357
Puzzle	-0.3887	0.137	-2.835	0.005	-0.657	-0.120
Racing	0.0175	0.072	0.244	0.808	-0.124	0.159

Role-Playing	0.0424	0.063	0.668	0.504	-0.082	0.167
Shooter	0.0573	0.061	0.942	0.346	-0.062	0.177
Simulation	0.2553	0.090	2.824	0.005	0.078	0.433
Sports	0.1119	0.065	1.716	0.086	-0.016	0.240
Strategy	-0.4594	0.095	-4.838	0.000	-0.646	-0.273
E	0.2047	0.053	3.839	0.000	0.100	0.309
E10+	-0.0207	0.059	-0.349	0.727	-0.137	0.096
M	0.3839	0.052	7.363	0.000	0.282	0.486



The general shape of the plot changed drastically in this model. The residuals overall did not really improve, however. The R^2 value is much lower in this model (0.172).

Test Set Evaluation:

First, I made a dataframe for the test set of X that matches models 1,2,3. Using those models, I predicted on the X_test. Then I computed the SSE (Sum of Squares Error) and MSE (Mean Squared Error). I found the RMSE(Root Mean Squared Error) by taking the square root of the MSE.

Model	Mean Squared Error	Root Mean Squared Error
1	2.830	1.68
2	2.609	1.62
3	2.828	1.68

A value of 1.68 means that on average, our predictions on video game global sales are off by 1.68 million dollars compared to their actual global sales value. Our second model performed the best since it has the lowest RMSE which turned out to be our "best" statistical model that we could build. Model 1 and Model 3 performed about the same, but Model 3 is better than Model 1 due to having a lower MSE. This is surprising considering that the R^2 value of the first model is higher than the other two models, but Models 2 and 3 are better statistical models because their residuals look better. If we had a valid model, it is likely that our predictions on the test set would be more accurate and result in a RMSE lower than 1.62.

Prediction Using Machine Learning Methods

Feature Selection:

I use the set of features in the first multiple linear regression model I used earlier. So this includes all the significant variables found at the univariate level. I do not worry about the issue of collinearity and keep Critic_Score and Years_Since_Release in the model. This is because collinearity has no impact on actual predictive power (I tried removing Critic_Score and had worse predictive power for Linear Regression. I did not experiment further). It has an effect on the model's coefficients. Since machine learning is not concerned with the coefficients as much as the prediction, collinearity is not an issue in the way it is for statistical models where the coefficients are the concern.

Model building:

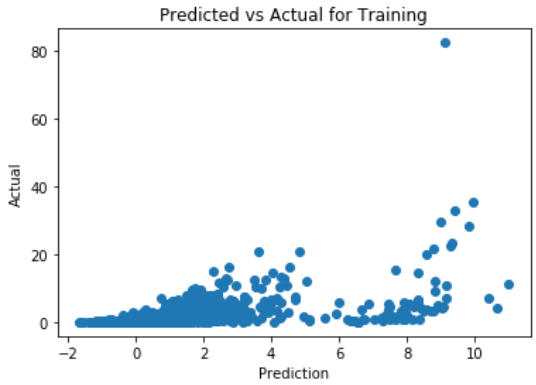
I used sci-kit learn to build several different models. For each model I had a natural log-transformed global_sales model and a non-log model. I utilized either a GridSearchCV or RandomizedSearchCV with 5-fold validation to select the best model before performing predictions. I computed the MSE for each model summarized below.

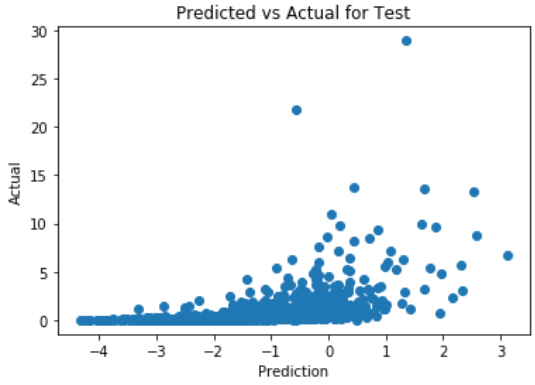
Model	Mean Squared Error (non-log)	Mean Squared Error (log)
Linear Regression	2.005	2.208
Ridge Regression	2.010	2.215
Support Vector Regression	1.919	1.758
Random Forest Regression	1.479	1.825
Extreme Gradient Boosting Regression	1.517	5.914

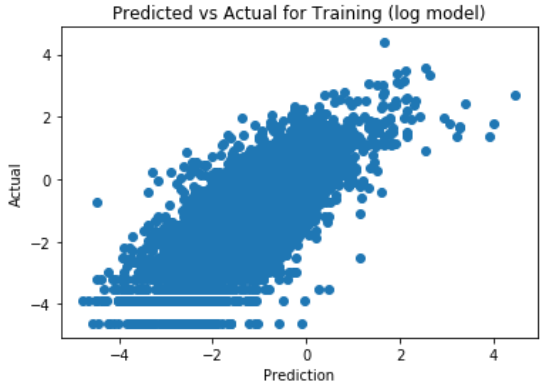
Check the Capstone Project 1 Part 2 notebook to see the space of hyperparameters chosen. Also note that I scaled the Support Vector Regression model with the median instead of the mean due to the presence of many outliers.

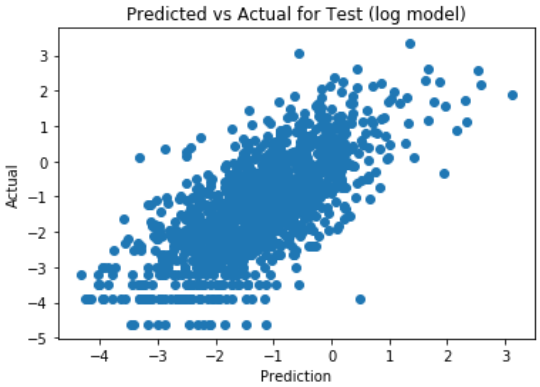
Given that in our statistical model, the natural log transformation improved our model, but it had a negative effect on the predictive power on almost all the machine learning models tried above, especially for XGBoost. Support Vector Regression was an exception which would be examined further.

Since this was a bit strange, I investigated predicted vs actual plots for Linear Regression for the training and test data and non-log and log models.





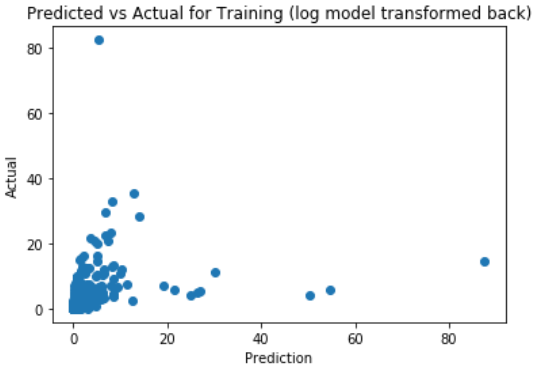


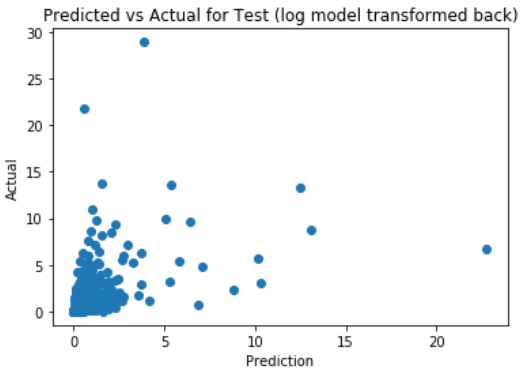


The first thing I noticed was that the relationship between the predicted vs actual exhibited a strong linear relationship.

In the log transformed model we trained the model using the log transformed response variable. This meant that the model was optimized to minimize the residuals and MSE for this variable. The outliers do not have an effect as much and the distance between them are small. When the MSE was computed, the predicted values were put through the exponential function to have the MSE on the same scale as the other models. When this happened, the distance between the points and the regression line may no longer be minimized because the model was trained with the log data. Further research or knowledge on how the linear regression model algorithm from sci-kit learn may be helpful on this end.

We can examine the plots if we transformed back into the non transformed scale.



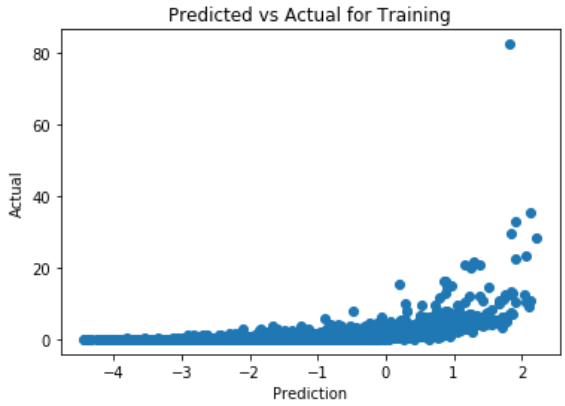


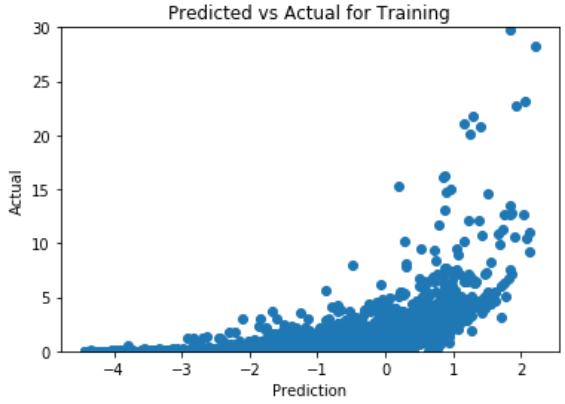
In the case with no transformations, outliers were only present toward the right side of the plot. This would mean that the actual value was much higher than the predicted. Given the scale of the plot, this was typically the case. Big outliers such as Wii Sports at around 82 million tilt the regression line upward.

In the two plots above, there are outliers present on the extreme y, low x and extreme x, low y. This would average out the regression line toward somewhere around the middle and make a more robust model. This actually is worse for our case because sales is very volatile. It may have happened by chance due to how the data was split into the training set and test set that a less robust model performed better than the robust model due to the presence of more outliers in the test set.

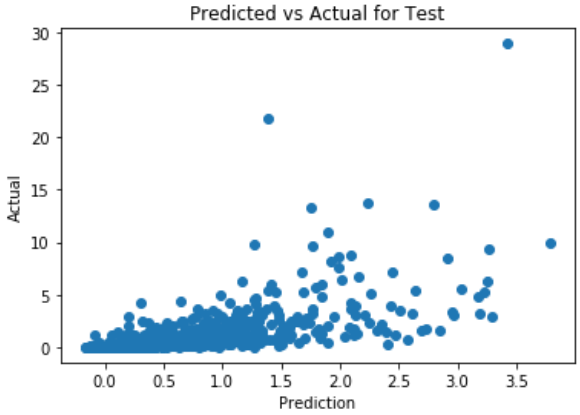
The Support Vector Regressor was the only model to perform better with a log transformed model. The non-log model performed a bit better than the Linear Regression model. This is likely attributed to that the Radial Basis Function (RBF) is a non-linear method and as we saw with the predicted vs actual plots of the non-log model did not have much of a linear relationship.

The Support Vector Regression algorithm aims to use only a subset of the training data. It aims to include points with errors within ε. This means that outliers are ignored. Our problem with the log model as seen in the Linear Regression case had an issue with outliers. Because we had to apply the exponential function to convert the data to match the scale of the MSE of other models, the outliers had a much larger effect. In this case, since we have a more linear relationship instead of the non-linear relationship in our non-log model we have a more accurate prediction.

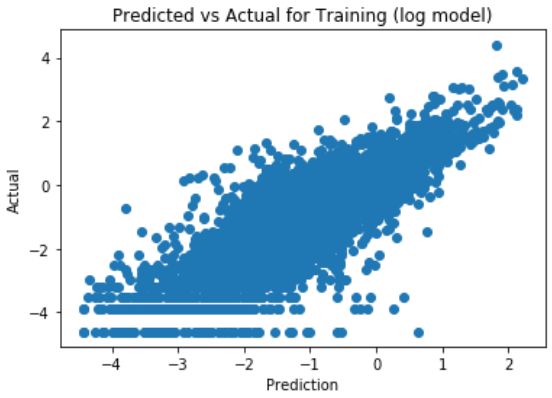


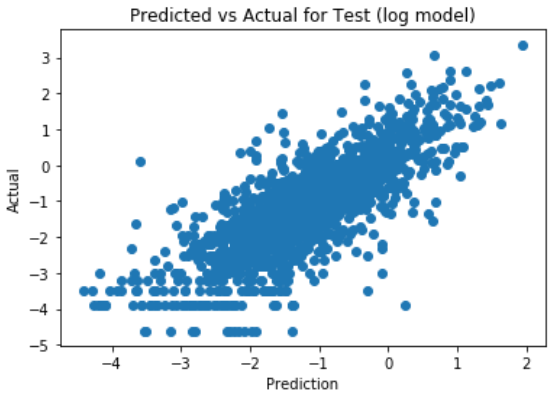


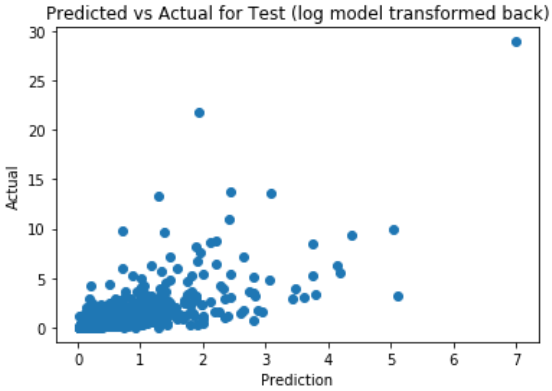
Above we see how the training data looks like for the non-log model for the Support Vector Regressor model. It looks relatively similar to the Linear Regression non-log tranformed data. The difference is more points toward the middle/right side of the plot which leverages the regression line a bit more upward. As we saw earlier with how the non-log transformed model was better because it was less robust, we see something similar here. Again, this may be due to how the data was split into train and test so that it coincidentally favored a less robust model due to more outliers in the test set.



Above see how the test data looks like. It looks similar to how the non-log transformed test data looked like for Linear Regression except a bit more linear here. This resulted in a slightly more accurate predictive model.







Above we see a very strong linear relationship. There are no outliers here unlike earlier in the Linear Regression model. This is because only the points within the ε range. The lesser outliers in the third plot are more centered around the center where the regression line would be drawn. This results in a lower MSE.

Ridge Regression performed about the same as Linear Regression. This makes sense because they are both linear models. Ridge Regression has an edge over Linear Regression when collinearity is an issue. Here it is the case that collinearity is not an issue, considering that removing the term with the highest VIF resulted in worse predictive power. When the beta coefficients of Ridge Regression are small, the Sum of Squares Error will increase while the penalty is minimized. In our case, the MSE increases as well due to poor generalization. This resulted in a slightly worse model when compared to Ordinary Linear Regression.

The Random Forest Regressor performed the best out of all the models. It is scale invariant, unlike the Support Vector Regressor. It is similar to the Decision Tree Regressor except that the Decision Tree Regressor has a bigger issue with overfitting than the Random Forest Regressor does. Both of these models can learn very complex data. Random Forest Regressors aim to minimize the variance and the Sum of Squared Error. Since we are concerned with the MSE calculated from the SSE, this results in the best model when using the MSE as the accuracy metric.

The Extreme Gradient Boosting Regressor (XGBoost) performed very similarly to the Random Forest Regressor. This is because both of them are based on decision trees. XGBoost is great in many scenarios and is a good all-rounder. It performs quickly and is robust to overfitting. It aims to reduce bias instead of reducing variance. In our particular problem with volatile sales, reducing variance is more effective, so Random Forest performed better.

Note that I used a GridSearchCV with XGBoost and only a RandomizedSearchCV with Random Forest. One possible unexamined reason why XGBoost performed worse is a poor choice of hyperparameters. But Since I used a RandomizedSearchCV with Random Forest, that model can be improved by using a GridSearchCV instead.

Limitations

Since the data was taken from Metacritic, any conclusions or trends found in the analysis applies only to the population of video games found there. Since they did not have certain platforms such as NES, conclusions and trends can only be generalized to the platforms hosted on Metacritic. There also may have been a bias toward more well-known, popular, or newer titles. Since the data does not include any values with no Critic_Count, Critic_Score, User_Score, and User_Count, some titles are lost when null values were dropped in the data cleaning stages. It is not the case that all null values came from Platforms not hosted on Metacritic.

The Genre category is not well-defined in the data. As seen in the Exploratory Data Analysis, the Action genre's average sales per game was nearly identical to the average of the average sales per game of every other genre. Games such as Grand Theft Auto were tagged as Action instead of Shooter, when really it is both. It would be better if the Genre category allowed games to have multiple genres instead of one. It is rarely the case a game is only one genre.

The 'Other' category for Developer, Publisher, and Platform do not actually represent all other categories not listed. It is strictly referring to how they were defined in the Data Cleaning notebook. Any trends or conclusions involving those categories would only extend to those categories.

Some groups are under-represented and have bias associated with them. One such group is for the Platform variable. The Playstation has a low sample size but very high average sales per game. Under Model 1 it had the highest coefficient and under Model 3 it was still fairly high compared to other Platforms. This is likely due to that the critics and users only rated the more popular games for that system. So there is some bias here involving only including the popular games. If the sample size for these under-represented categories were higher, they would represent their true effect better.

The data also may not be accurate. I saw earlier in the Exploratory Data Analysis that there were three obviously miscategorized data points. There could be more errors in the data that are not as obvious.

The data that were split into the 80% train and 20% test may have been uneven in that the test data could have more outliers than the training data. Given the volatileness of the sales data, some conclusions or results may have only been obtained due to how the data was split into the test and training groups.

Conclusion

Since the assumptions of linear regression were not met, the models I found at the end were invalid. This means that the set of significant predictors may not be good predictors for Global_Sales. It also means that our parameter estimates for the coefficients are not reliable. The model with the best

predictive power was Model 2, which was the “best” statistical model I could build. It had a MSE of 2.609.

The Random Tree Regressor performed the best out of all the machine learning models I tried. It had a MSE of 1.649. Not every machine learning model was tried, so I do not know if Random Tree Regression is actually the best model or not for this data. The hyperparameters were fairly arbitrarily chosen as well so the models that were attempted could have been better optimized. Sometimes a RandomizedSearchCV was used when a GridSearchCV would result in a better model. Due to computational restraints, I could not try every possibility.

Overall, when comparing the predictive power of the “best” statistical model versus a machine learning model, the machine learning model performs better. It is also easier because the machine learning model does not have as many assumptions as the statistical model would need to satisfy to have a valid model. The only time the statistical model performed better was for the log model in the XGBoost case. The question is not completely answered because we did not have a valid statistical model.

Given that it takes much longer to develop a valid statistical model, for prediction problems, machine learning has an edge over statistical models. This may only be the case with a larger data set. In this case, there was 6825 data points with our training and test groups both included. The disadvantage of machine learning is computational time. For large datasets and large hyperparameter spaces, it takes longer to get an accuracy metric.

Future Work

In a future study, it may be worth trying to use a non-linear model or a generalized linear model instead of a linear regression model while trying to build a valid statistical model. It is clear from all the attempts in trying to remedy the non-constant variance that using a linear regression model does not work in this dataset. Finding a valid model would be worth in figuring out which variable is a helpful predictor in determining the success of a game measured by Global Sales.

It may also be worth looking around for better data. In particular, it would be good to see if there are other data that includes all the genres or more variables of interest. This new data would have to contain the name of the game so that it can be joined on that column.

Since this data is from Dec 22 2016, newer data may now be available. Originally, this dataset was a combination of two web scrapes hosted on Kaggle. To utilize the newer data, webscraping the data from VGChartz Video Game Sales and Metacritic would be required.