Thomas Terziu

Dr. Julie Butler

DSC-140S

6 December 2024

Predicting the Popularity of Video Games using Data Analytics

Video games are one of the most popular forms of media people consume. Because of this popularity, several companies such as Microsoft and Activision have entered the video game market to try to get a share of the cash consumers are willing to pay to purchase a game. Being able to get a metric of what qualities make a game popular alongside what would increase the sales of a game would be useful information for these companies to boost profits. With access to a dataset that includes statistics that we could compare against sales and ratings, we would be able to recognize patterns that could drive success within the gaming industry. This data could be utilized to identify key factors, such as console of origin and playtime, that influence the game's popularity. This would allow the developers to change aspects of the game to provide consumers with what they are looking for.

To identify these qualities that could drive success in ratings and sales, we were able to get a spreadsheet of video game console data from a study conducted by Joe Cox, a professor of Financial Technology at the University of Portsmouth in the United Kingdom. This data includes a plethora of information on console videogames. This dataset includes information such as the title of the game, the publisher, the review score, sales, and a plethora of statistics on how long it took for someone to complete a game, given how they play it, whether casually or trying to get to the end as quickly as possible (Cox). This data is stored in the form of both numerical data for things such as review score and object data which contains data such as game genre for example.

To investigate this data, we will be using cell-based Python code. This is because of Pythons benefits when it comes to program syntax, wide range of support, and its wide accessibility to libraries that can perform high quality data analysis (University of Dublin).

To get a basic overview of notable columns of data its useful to mention are that the average amount of players that a game can have are 2, the review scores range between 1-100 with the average score being 69, and the sales on average are $500,000 and range from $15,000,000 and $10,000. Notable game publishers in this study are and to EA, Ubisoft, Sony, 2K, NAMCO, Disney, and Nintendo (Cox). The game consoles these games could be on are the Xbox 360, Nintendo DS, PlayStation 3, and Nintendo Wii.

When investigating both sales and rating, it's important to get a view of the entire picture. For this, we can make a heat map of correlation scores so we can determine the relationship between different columns of data. We will do this by utilizing Matplotlib and NumPy, which are python libraries that help calculate statistics and visualize data (Matplotlib; NumPy). As a result, we get Figure 1 which excellently displays all numerical values within the data frame and what their correlation is to one another. This will speed up our analysis of determining how to increase review rating and sales of a game. Both values exhibit a subtle correlation, as indicated by their low correlation scores. This can be observed by matching the color of the block to the corresponding number on the legend. Correlation values closer to 0 indicate no correlation, while those closer to 1 represent the strongest correlation. Some notable columns of interest for that we would want to investigate is playtime due to their higher correlation. Since there are several columns for playtime, we can generalize the column we investigate to "Length.All

PlayStyles.Average” since it’s an average of all playtimes and helps include of demographics of people that play the game not cater to only one specific playstyle.

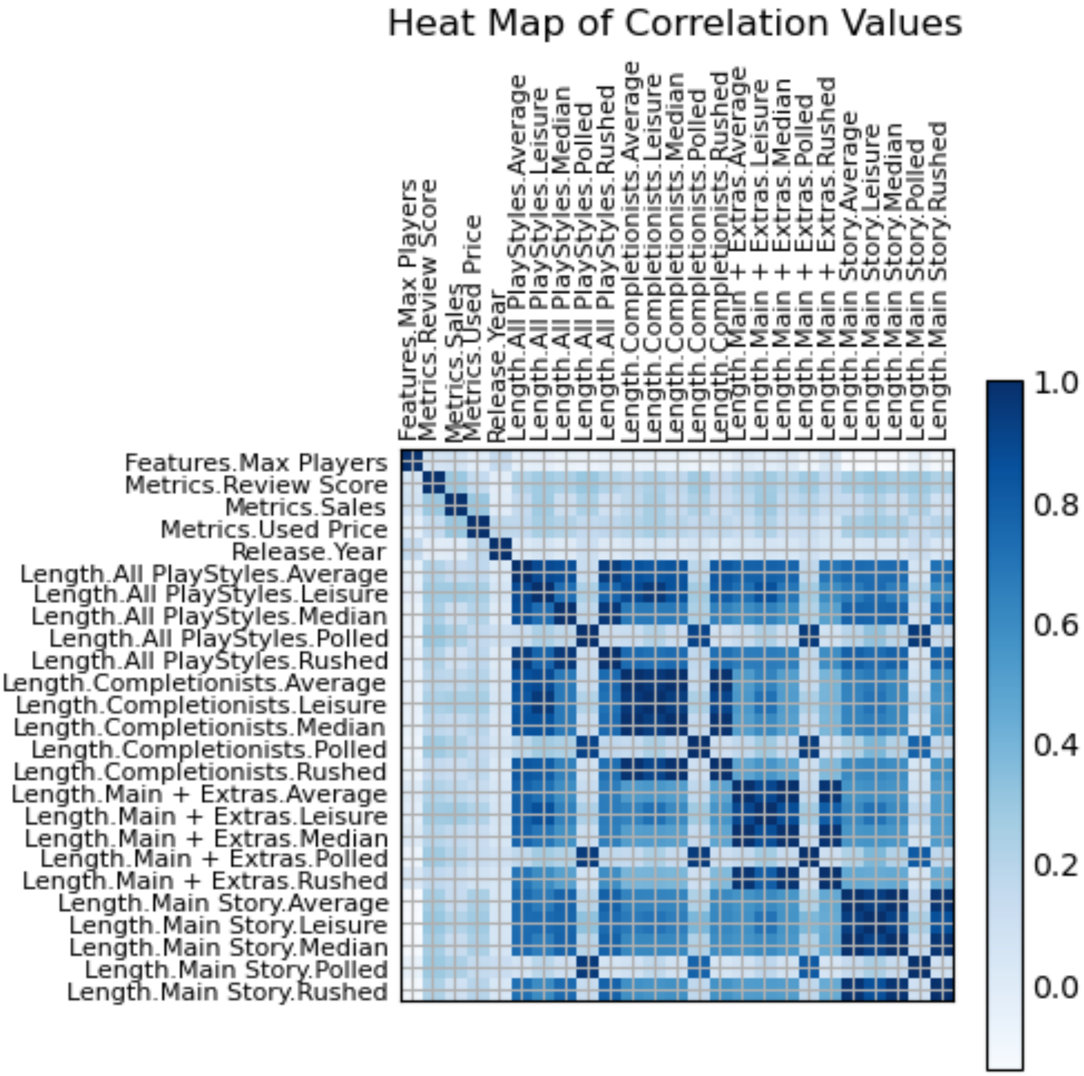

Figure 1: Heat Map of Numerical Data

. To get a general overview of if there is a correlation between “Metric.Sales” and Length.All PlayStyles.Average”, we can run a T-Statistic Test. This test can be used to determine test whether two groups of data are statistically significant. Usually, a T-Statistic test is done by using the formula:

$$t = \frac{\bar{x} - \mu}{\frac{s}{\sqrt{n}}}$$

x̄ = average of the sample of data

μ = assumed average

s = standard deviation

n = total number of data points

Thankfully, this process is simplified within Python using the NumPy library. By using the code:

```
t_statistic = stats.ttest_ind(x,y)
```

We can determine the T-Statistic of playtime and sales. When this command is returned, we get a T-Statistic of -19.91 which indicated that out datasets are not closely related, making this test unreliable. To get more reliable findings, we can graph both values on a scatter plot and graph a line of best fit to confirm a positive correlation. This is performed in Figure 2 where we can observe a slight positive statistical correlation between the two columns of data which indicates a small relationship between the two columns of data. This demonstrates that if your game has a longer campaign, then the sales will increase. Something that is important to consider is that there are several outliers within the data. They will not be removed since it doesn't accurately represent the time people spent playing these games.

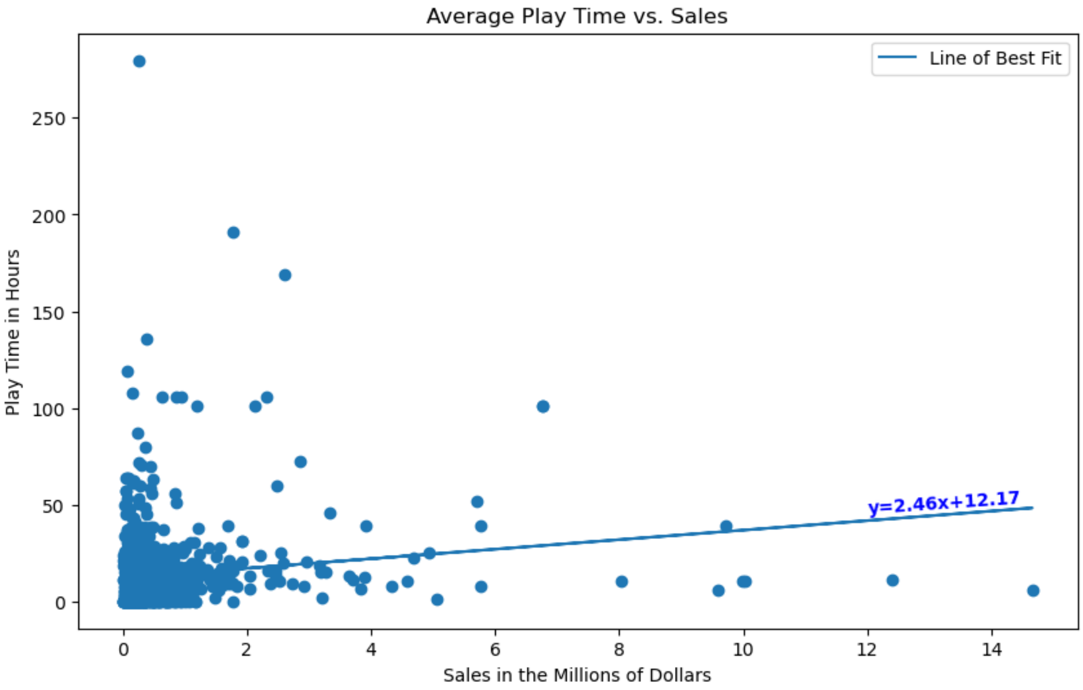

Figure 2: Average Play Time vs. Sales

To get a general overview of the review score data, we first must set are bar as to what the average score of the data is and what are highs and lows are. This can be done easily with A violin chart which shows us a average review score between 60 and 80 out of 100.

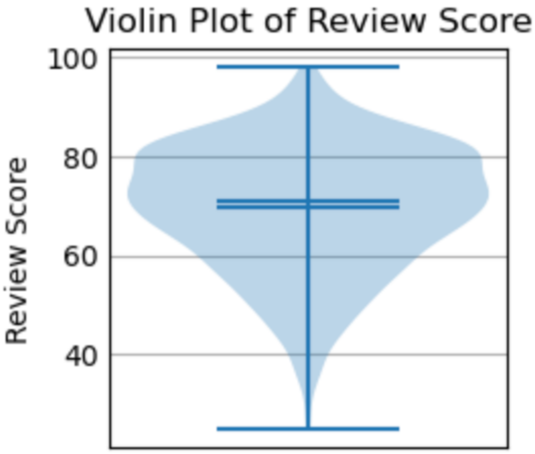

Figure 3: Review Score Violin Plot

From this point, we are able to use one of the positive correlations from Figure 1 and preform a Chi-Squared test to see if there is significant correlation. Traditionally, the formula for a Chi-Squared test would be:

$$x^2 = \sum \frac{(O - E)^2}{E}$$

x^2 = Chi-Squared

O = Observed Value

E = Expected Value

This is a simplified process within Python. By using the SciPy library within python which can be used to do advanced statistics (SciPy). This test can be easily implemented by using the following code:

```
crosstab = pd.crosstab(x,y)
c, p, dof, expected = scipy.stats.chi2_contingency(crosstab)
```

We can get the Chi-Squared value once we print the p-value . When we test our review score against our average playtime, we get a P-Value of 3.13e-10 which indicates a statistical relevance. between these two data columns. When we compare this to our correlation matrix in Figure 1, we can see that this is a slight positive correlation which shows that although small, review does go up when average play time increases. If we want to extend our search further for how to get the highest review score, we can graph a bar chart to see which console has the best game ratings. We can pull this data out within Figure 5 where on average the PlayStation 3 has the highest review score. But it is important to mention that most other console are not too far behind that statistic making this test also fairly unreliable.

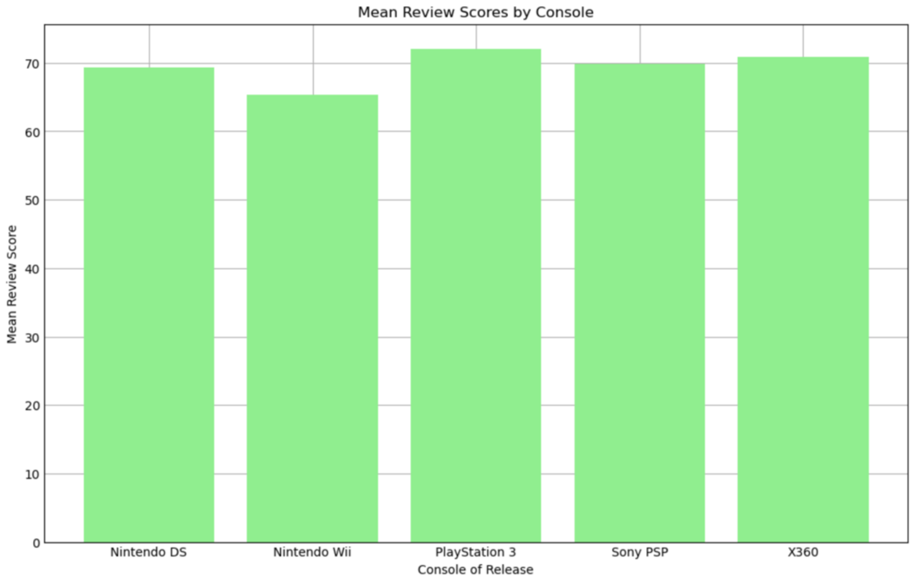

Figure 4: Review Score by Console

To investigate the correlation further, we can run a machine learning algorithm to predict the review score based on the console of choice and the average playtime which we can use to identify if there any sort of relationship between the data. Using the python library SciKit Learn, we can run a machine learning algorithm called K-Nearest Neighbors, which predicts a set of data based on how close the data points are to one another. To do this, need to find the optimal number of neighbors which will help us get a greater prediction accuracy. To do this we can cycle through numbers of neighbors and record their accuracy when predicting the data. When doing this, we can an optimal number of neighbors of 15 which returns an prediction accuracy of 65%. This was achieved by putting the games into two catagories based on average review score. Games above a review score of 69 were put into the above average category while games under this were put into the below average category. We can graph these predictions to see what the algorithm predicted vs to what was true which can be seen in Figure 5. The graph displays the accuracy of the predictions based on colour with darker colors meaning for that category wasn't predicted accurately and lighter colors having a higher accuracy. We can see that the algorithm does a similar job when predicting both above and below average review scores with more accuracy within above average games. We can use this information alongside the information from our P-Value and from our graph in Figure 4 that there is a small positive increase within review score based on both console and games.

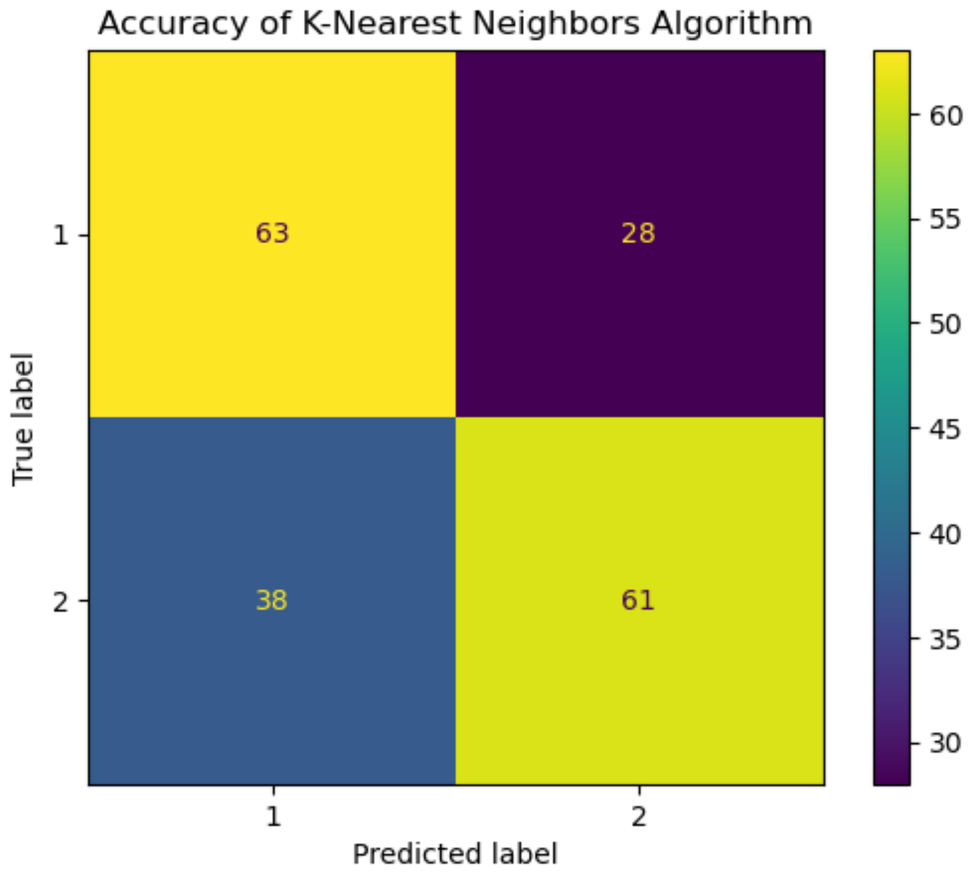

Figure 5: Machine Learning Algorithm Accuracy

Sources of error that could have had while investigating these findings are that the Xbox 360 console has more games than any other console which could have changed the results to be bias to the Xbox within both sales and review score. Another potentional source of error is our lack of consideration into PC gaming and the platforms the games exist on their such as Steam or the Epic Games Launcher. This could impact our data since we only took a sample of games from 2004-2008 based on relevant consoles at the time.

When determining how to make a blockbuster game, there are a lot of factors such as playtime that should be taken into consideration when developing your game. Key features such as the length of the story, the multiplayer aspect, and the console could make or break your chances at great reviews and sales. This is proven by the results we got within our linear line regression in Figure 2 and our results within our Chi-Squared test and machine learning. Developers need to get data from their consumers so they can produce a hit game that will have great ratings and great reviews.

Work Cited

Cox, Joe. “What Makes a Blockbuster Video Game? An Empirical Analysis of US Sales Data.” *University of Portsmouth*, John Wiley and Sons Ltd, 26 July 2023, researchportal.port.ac.uk/en/publications/what-makes-a-blockbuster-video-game-an-empirical-analysis-of-us-s.

University College Dublin. "Why Do Data Analysts Use Python?" *UCD Professional Academy*, https://www.ucd.ie/professionalacademy/resources/why-do-data-analysts-use-python/?utm_source=chatgpt.com. Accessed 6 Dec. 2024.

Matplotlib: Python Plotting Library. https://matplotlib.org/. Accessed 6 Dec. 2024.

NumPy: The Fundamental Package for Scientific Computing with Python. https://numpy.org/. Accessed 6 Dec. 2024.

SciPy: Scientific Computing Tools for Python. https://scipy.org/. Accessed 6 Dec. 2024.