**Predicting Video Game Review Scores**

Team Amazing

DATS 6103, Final Project

**Hyunjae Cho, Chenyang Lu, Michael Pagan**

**Introduction**

As the next generation of gaming is among us, hundreds, if not thousands, of new (or remastered) games will be released in the coming years to keep gamers entertained. One way to know if a game is worth purchasing is to consider the reviews a game gets. G[ames do not usually flop](https://en.wikipedia.org/wiki/List_of_video_games_notable_for_negative_reception#Fallout_76_(2018)) in the reviews and our data show that, in fact, most games do quite well, with an average score of about 7/10. However, not all reviews are equal - recent controversy over [skewed scoring](https://www.forbes.com/sites/paultassi/2020/06/21/the-last-of-us-part-2-is-getting-predictably-user-score-bombed-on-metacritic/?sh=6a7c86f85c25) have come to light and some groups like [Kotaku have even stopped giving out numeric ratings in their reviews. Thereby, it is clear that it is in a gamer’s best interest to seek](https://www.polygon.com/reviews/2018/9/4/17689100/polygon-reviews-no-scores) multiple outlets for reviews since the outcome can vary depending on the group that reviewed it.

Our group is interested in using the reviews from popular websites for video game reviews to predict the ratings of the next generation’s games. Our project focused on scraping popular websites for video game reviews to create a model that can predict a game’s review score.

**SMART Question**

Given the importance a video game’s review may play in a gamer’s decision to buy a game, and the potential for disparate reviews to come from the various media outlets for reviews, we asked the following:

*Can we predict the review score a game will get based on the ratings from three major review websites for various genres and platforms?*

However, we were forced to refine this question after building our linear regression as there were insufficient features in the model to accurately predict score. Thus, we asked:

***Can we predict if a game will get a review score of 80% or better*** *based on the ratings from three major review websites for various genres and platforms?*

To answer these questions, we used several modeling techniques to predict the review scores.

**Methods**

The Selenium and BeautifulSoup Python libraries were used to scrape the title of the game reviewed, the platform it was reviewed on, the genre of the game, and the score it received from Metacritic, OpenCritic, and GameSpot with a total of 46,958 data points captured between all three sites. To be consistent between sites, only modern platforms were considered, such as PlayStation 4 and Xbox One, though the sites had reviews for nearly every platform ever created. Similarly, we narrowed the scope of the genres each site reviewed. Since Metacritic had a finite list of 17 genres compared to the tens of genres available on the other sites, Metacritic’s genres were used as the metric to obtain reviews from the other sites. However, there is some margin of error in this approach as there may be discrepancies in how a site classifies a game, such as “Sports” vs. “Soccer”. Nevertheless, the following platforms and genres were used:

**Platforms:**

* PC
* PlayStation 3
* PlayStation 4
* PlayStation Vita
* Nintendo 3DS
* Nintendo Switch
* Nintendo Wii
* Nintendo Wii U
* Xbox 360
* Xbox One

**Genres:**

* Action
* Adventure
* Fighting
* First-person
* Flight
* Party
* Platformer
* Puzzle
* Racing
* Real-Time
* Role-Playing
* Simulation
* Sports
* Strategy
* Third-Person
* Turn-Based
* Wrestling

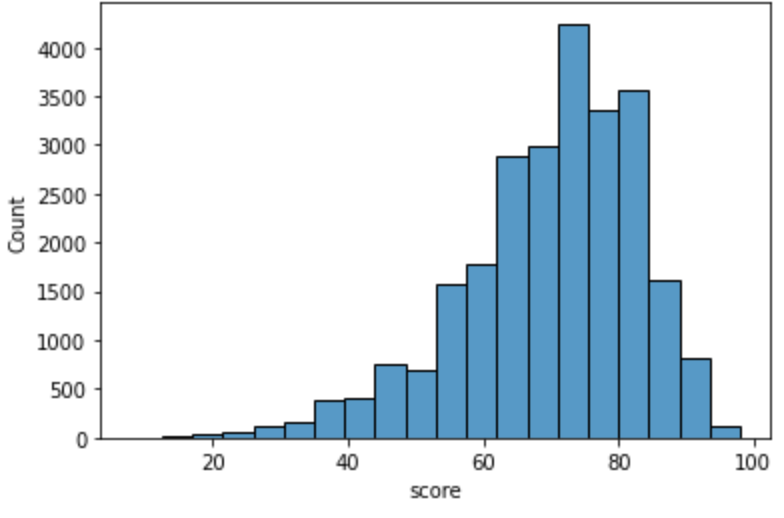
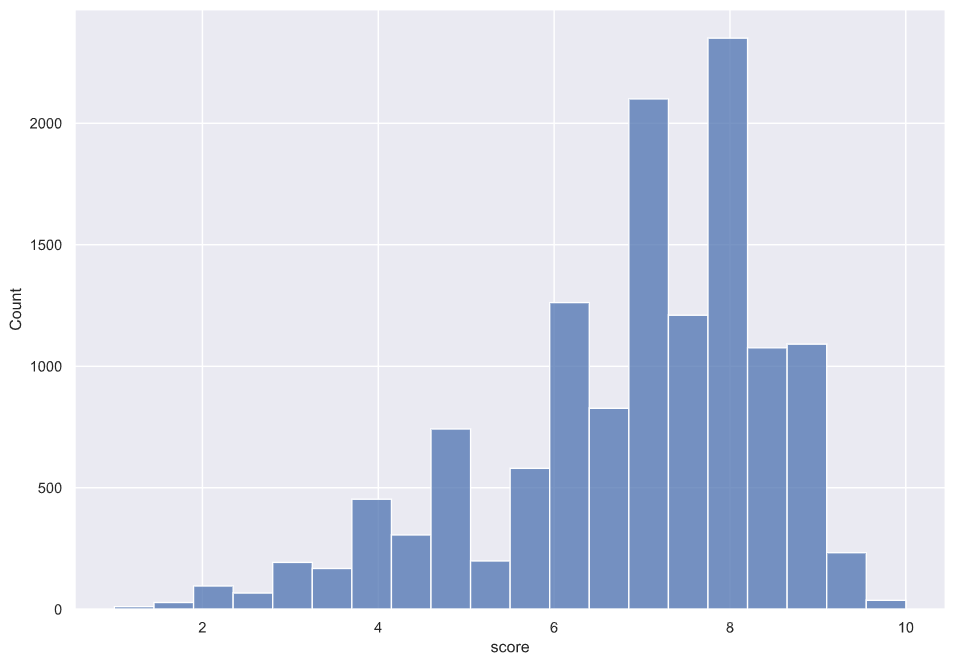
Linear regression models were constructed using a game’s score as the target variable and the platform and genre as the independent variables. Both independent variables were coded numerically to successfully integrate them into the model.

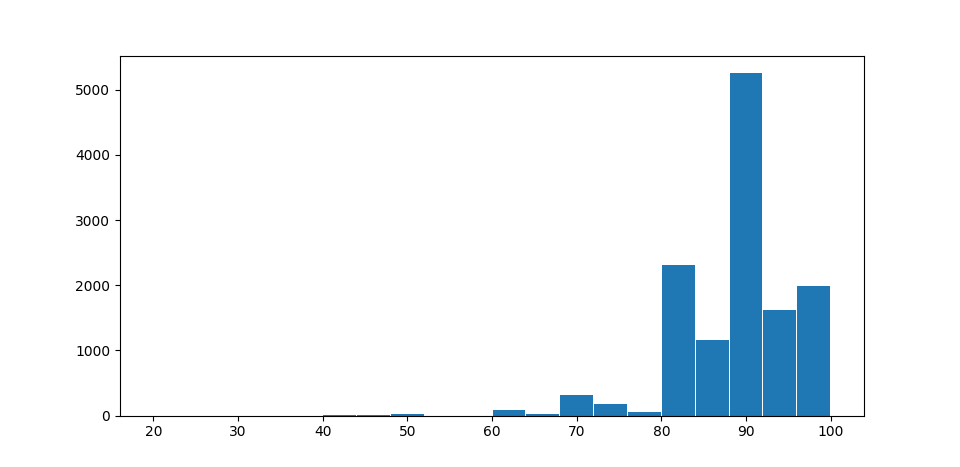
Logistic Regression, KNN (neighbors = 7), Linear SVC, SVC (gamma = auto), and Decision Tree (criterion='entropy', max\_depth=10, random\_state=1) classification models were made to predict if a game would score 80% or better.

All models were created using the scikit-learn Python library.

**Exploratory Data Analysis**

Data was scraped from [OpenCritic](https://opencritic.com/), [Metacritic](https://www.metacritic.com/), [GameSpot](https://www.gamespot.com/) using Selenium and BeautifulSoup and 46,958 data points were obtained. The score distribution for each site was relatively similar for GameSpot and Metacritic; OpenCritic rated games more highly than the other two sites **(Figure 1, Table 1**).



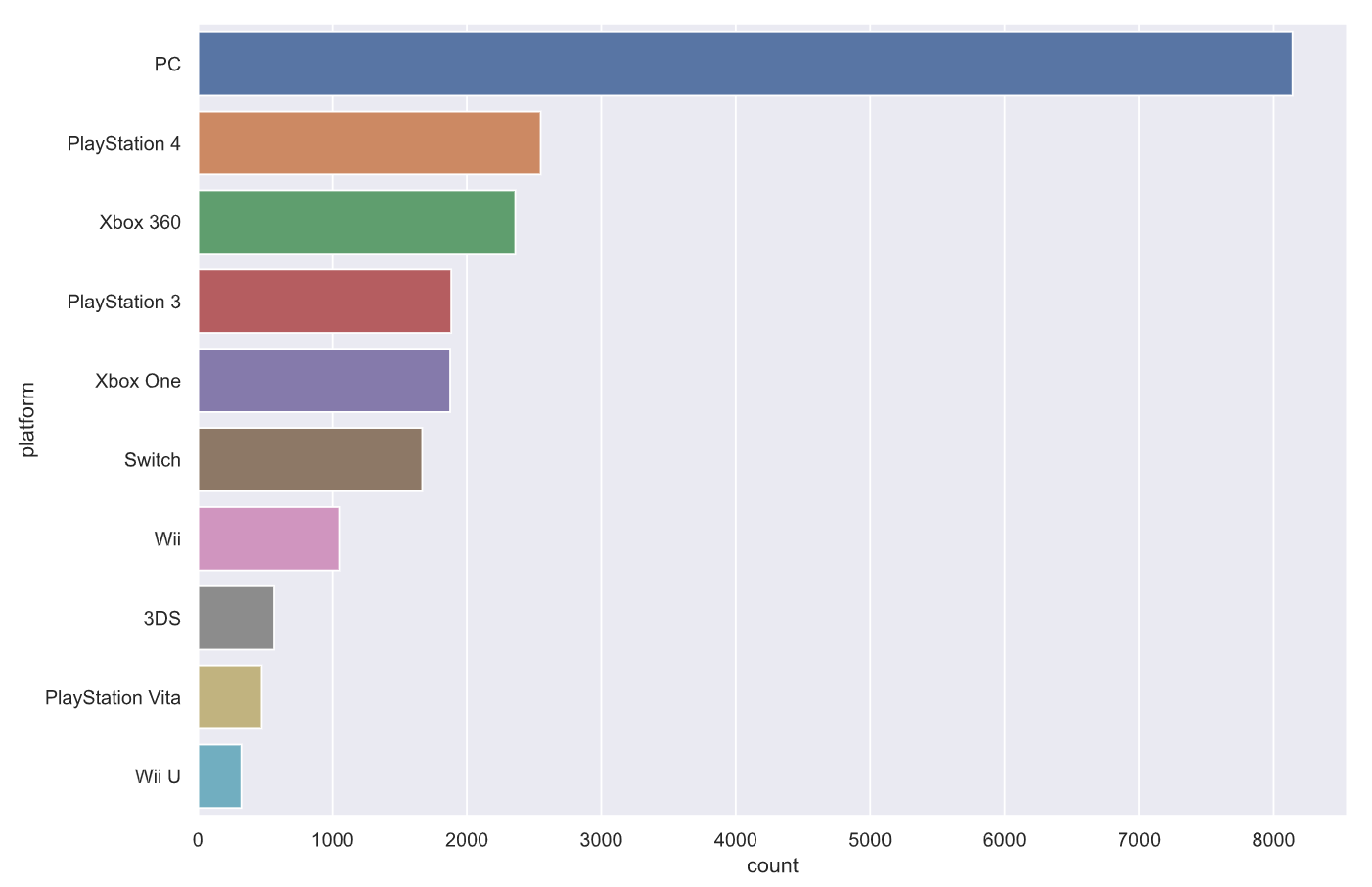
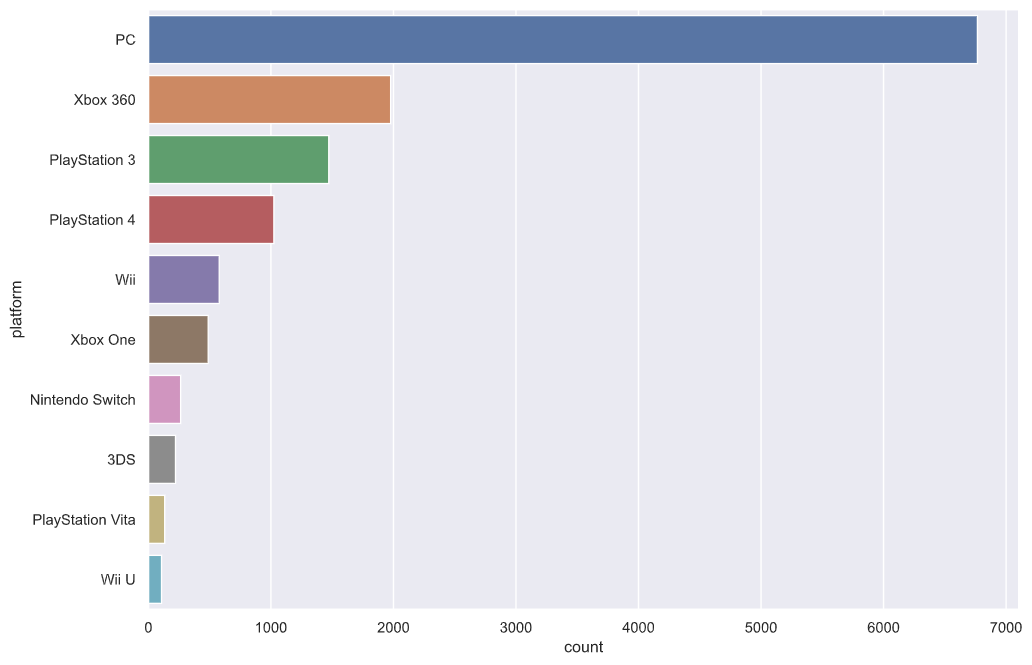


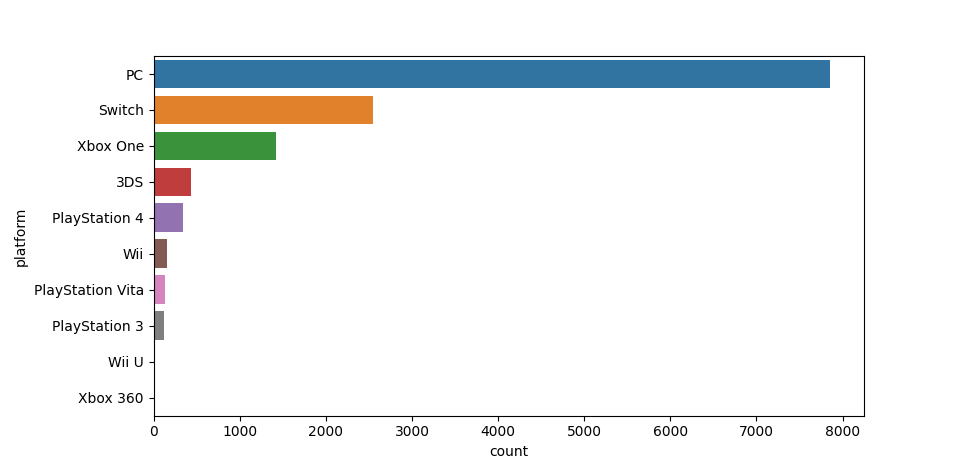
**Figure 1:** Score distribution from each website. Clockwise from top-right: Metacritic, GameSpot, OpenCritic.

**Table 1:** Mean and median for the score distribution of each website.

|  |  |  |
| --- | --- | --- |
| **Site** | **Mean** | **Median** |
| **OpenCritic** | 87.67 | 88 |
| **Metacritic** | 69.68 | 72 |
| **GameSpot** | 6.88 | 7 |

Between all three sites, games were most numerous on PC (**Figure 2, Table 2**); action was the most frequent genre (**Figure 3, Table 3**).

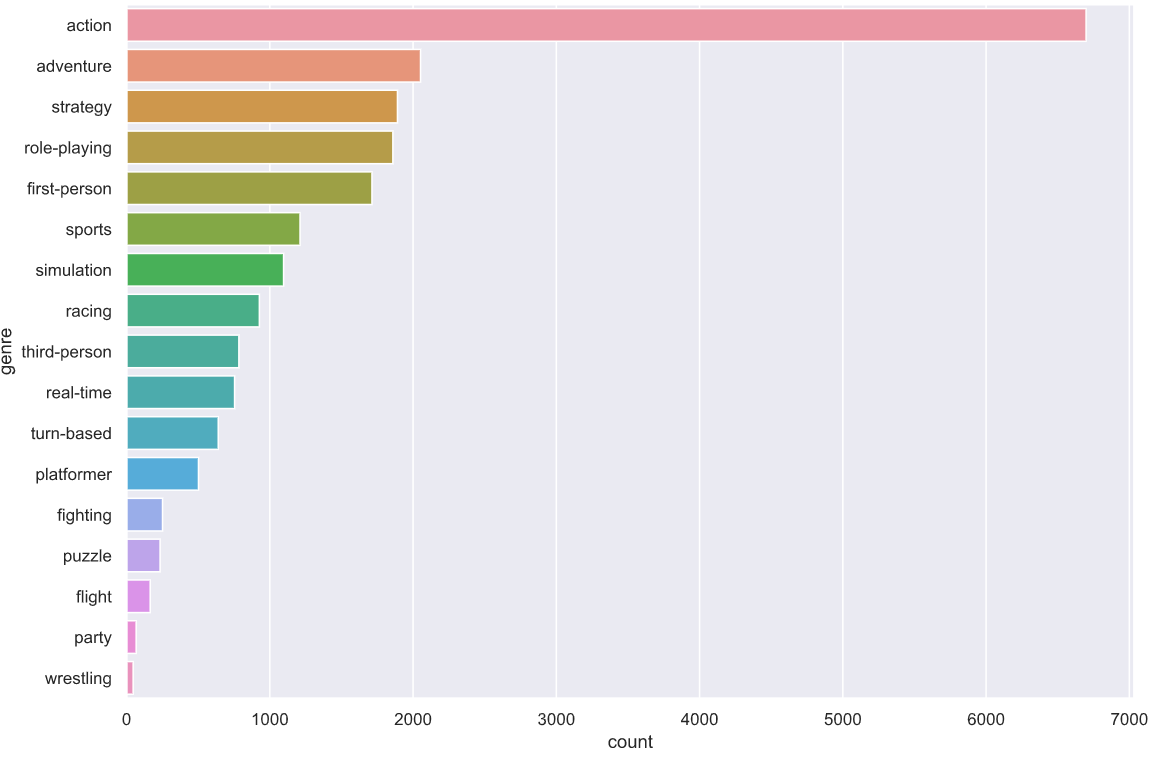
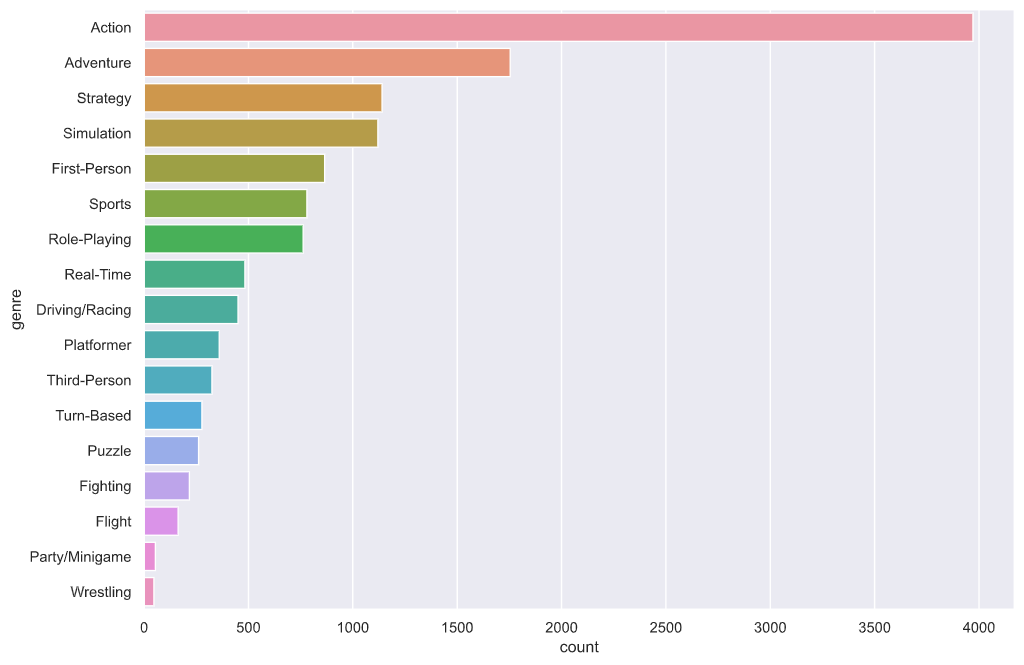


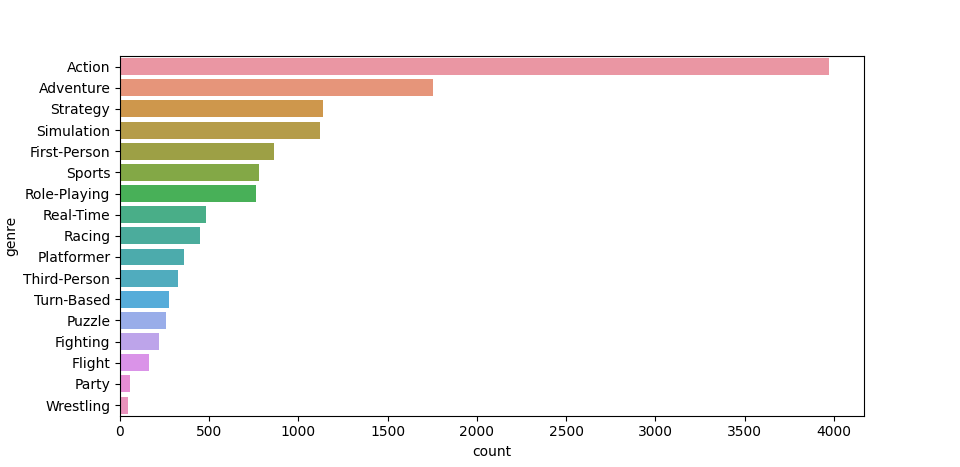


**Figure 2:** Count of platforms scraped from each website. Clockwise from top-right: Metacritic, GameSpot, OpenCritic.

**Table 2:** Platform counts from each website.

|  |  |  |  |
| --- | --- | --- | --- |
| **Platform** | **OpenCritic** | **Metacritic** | **GameSpot** |
| **PC** | 7856 | 8140 | 6766 |
| **Xbox 360** | 8 | 2360 | 1977 |
| **PlayStation 3** | 126 | 1884 | 1472 |
| **PlayStation 4** | 337 | 2549 | 1024 |
| **Will** | 153 | 1050 | 578 |
| **Xbox One** | 1416 | 1876 | 488 |
| **Nintendo Switch** | 2553 | 1668 | 263 |
| **3DS** | 436 | 566 | 221 |
| **PlayStation Vita** | 134 | 474 | 133 |
| **Wii U** | 8 | 323 | 107 |
| **Total** | **13027** | **20890** | **13029** |





**Figure 3:** Score count by genre. Clockwise from top-right: Metacritic, GameSpot, OpenCritic.

**Table 3:** Number of Counts by Genres

|  |  |  |  |
| --- | --- | --- | --- |
| **Genre** | **OpenCritic** | **Metacritic** | **GameSpot** |
| **Action** | 7590 | 6698 | 3971 |
| **Adventure** | 1658 | 2052 | 1754 |
| **Fighting** | 90 | 251 | 217 |
| **First-Person** | 335 | 1713 | 865 |
| **Flight** | 3 | 165 | 217 |
| **Party** | 41 | 67 | 54 |
| **Platformer** | 213 | 502 | 360 |
| **Puzzle** | 47 | 234 | 261 |
| **Racing** | 21 | 927 | 450 |
| **Real-Time** | 90 | 754 | 483 |
| **Role-Playing** | 91 | 1859 | 762 |
| **Simulation** | 460 | 1096 | 1120 |
| **Sports** | 458 | 1211 | 780 |
| **Strategy** | 1560 | 1891 | 1140 |
| **Third-Person** | 156 | 784 | 325 |
| **Turn-Based** | 21 | 640 | 277 |
| **Wrestling** | 51 | 146 | 47 |
| **Total** | **12885** | **20990** | **13083** |

As illustrated in **Figure 4**, OpenCritic reviewers tended to give higher scores than Metacritic and GameSpot. The first and third quartile of the OpenCritic scores ranged from 80 to 95. On the other hand, Metacritic and GameSpot boxplots range from 60 - 80 and 6 - 8, respectively. The party genre scored markedly lower than others, receiving a score ranging from 40 to 65 and 4.5 - 6.5 for Metacritic and GameSpot, respectively.

|  |  |  |
| --- | --- | --- |
| [OpenCritic Genre Score Boxplot] | [Metacritic Genre Score Boxplot] | [GameSpot Genre Score Boxplot] |

**Figure 4:** Game score by genre

We had similar results that reviewers from OpenCritic tended to give scores between 80 - 9. Scores ranged from 60 - 80 and 6 - 8 in Metacritic and GameSpot, respectively. Interestingly, Wii U had lower scores than other platforms in OpenCritic, and Wii had lower scores in Metacritic and GameSpot (**Figure 5**).

|  |  |  |
| --- | --- | --- |
| [OpenCritic Platform Score Boxplot] | [Metacritic Platform Score Boxplot] | [GameSpot Platform Score Boxplot] |

**Figure 5:** Game score by platform

**Modeling**

**Linear Regression**

We first built a linear model with two explanatory variables - platform, genre - but the model was a poor fit. Test and train scores for the linear regression (**Figure 6**)were all very small - no one model breaking >0.01 accuracy (**Table 4** ).

h(xi) = βi + Platform(xi) + Genre(xi)

OpenCritic:

Metacritic:

GameSpot:

**Figure 6**: Linear regressions score as a function of platform and genre for each website

|  |  |  |
| --- | --- | --- |
|  | Train Score | Test Score |
| OpenCritic | 0.00290 | 0.00160 |
| Metacritic | 0.00035 | 0.00075 |
| GameSpot | 0.00280 | 0.00380 |

**Table 4:** Linear regression test and train score

Classification models were then constructed to understand if we could predict if a score would receive an 80% or better rating from each of the three sites. We first utilized a logistic regression to fit our data to predict the scores. The resulting test and train accuracies for the logistic regression were high. However, **Figure 7** shows that the ROC-AUC value was low and our logistic regression was also poor fit.

**Logistic Regression**

|  |  |  |
| --- | --- | --- |
| [OpenCritic ROC-AUC]  Logistic ROC AUC value : 0.550 | [Metacritic ROC-AUC]  Logistic ROC AUC value : 0.508 | [GameSpot ROC-AUC]  Logistic ROC AUC value : 0.547 |

**Figure 7:** Logistic ROC AUC curve

Several other classification models were explored, including K-nearest neighbor, Decision Tree, SVM, and Linear SVM models (**Table 5**). Evidently, the models often predicted no better than the null model (**Appendix**), but the KNN models for each site appeared to be among the best models relative to the other classifications models that were made (**Figure 8, Table 6, Table 7**).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **OpenCritic** | | | | | |
|  | Logistic | KNN | Decision Tree | SVC | Linear SVC |
| **Score (train)** | 0.6165 | 0.7532 | 0.6581 | 0.6567 | 0.6257 |
| **Score (test)** | 0.6020 | 0.7479 | 0.6546 | 0.6152 | 0.6152 |
| **Cross Eval** | 0.5368 | 0.5652 | 0.3599 | 0.6299 | 0.6299 |
| **Metacritic** | | | | | |
|  | Logistic | KNN | Decision Tree | SVC | Linear SVC |
| **Score (train)** | 0.7745 | 0.7678 | 0.7748 | 0.7744 | 0.7744 |
| **Score (test)** | 0.7726 | 0.7673 | 0.7726 | 0.7725 | 0.7725 |
| **Cross Eval** | 0.6784 | 0.6192 | 0.7499 | 0.7740 | 0.6384 |
| **GameSpot** | | | | | |
|  | Logistic | KNN | Decision Tree | SVC | Linear SVC |
| **Score (train)** | 0.6493 | 0.6317 | 0.6569 | 0.6549 | 0.6493 |
| **Score (test)** | 0.6608 | 0.6328 | 0.6588 | 0.6639 | 0.6608 |
| **Cross Eval** | 0.6516 | 0.4705 | 0.5375 | 0.6475 | 0.6493 |

**Table 5:** Model Results

**KNN**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **OpenCritic** | | | | |
|  | Precision | Recall | F1 -Score | Support |
| 0 | 0.34 | 0.19 | 0.25 | 880 |
| 1 | 0.64 | 0.81 | 0.73 | 1726 |
| **Metacritic** | | | | |
|  | Precision | Recall | F1 -Score | Support |
| 0 | 0.77 | 0.99 | 0.87 | 3676 |
| 1 | 0.35 | 0.03 | 0.05 | 1082 |
| **GameSpot** | | | | |
|  | Precision | Recall | F1 -Score | Support |
| 0 | 0.66 | 0.91 | 0.77 | 1722 |
| 1 | 0.35 | 0.10 | 0.15 | 884 |

**Table 6 :** Precision, recall, F1-score for the KNN model for each website.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | | OpenCritic | | Metacritic | | GameSpot | |
| A | | A | | A | |
| 0 | 1 | 0 | 1 | 0 | 1 |
| P | 0 | 171 | 709 | 3623 | 53 | 1562 | 160 |
| 1 | 328 | 1398 | 1054 | 28 | 797 | 87 |

**Table 7** : KNN confusion matrix

|  |  |  |
| --- | --- | --- |
| [OpenCritic KNN (k = 7)] | [Metacritic KNN (k = 7)] | [GameSpot KNN (k = 7)] |

**Figure 8:** KNN Plots

**Results**

With just two categorical factors, the models did not perform as well as we would have liked them to when predicting a game’s review score. The accuracy of each model was negligible and signaled that we needed more factors, at the least, to construct a model that would accurately predict a game’s review score. Given the limited amount of data that were available for each review on the sites chosen, we refined our question to that of a binary outcome and applied it to various classification models, though many of these models did not perform much better than the null model (**Appendix**). We wanted to know which classification method would be better to predict if a game will get a score of 80% or better. KNN turned out to be the ‘best’ model relative to other methods.

For the OpenCritic dataset, the model was able to predict a score of 80% or better with about 75% accuracy; there was high precision for predicting games with a score of 80% or better, but low precision for games scoring less than 80%; and the model could correctly identify the actual scores of 80% or better, given the high recall for this group.

The KNN model for Metacritic and GameSpot showed results similar to one another. The accuracy for the model using Metacritic’s data was 10% higher than when using Gamespot’s. The model for both datasets were better at predicting scores less than 80% than those greater than 80%; actual scores of less than 80% were more readily identified than those scores higher than 80%. This is the opposite of what we saw from the OpenCritic dataset, likely because of the differences in the test-train split between the sites given OpenCritic’s tendency to score games more highly than the other sites.

**Conclusion**

Video game reviews are a popular way to understand if a game is worth buying. Given the disparate mechanisms of reviews and variable scoring between reviewers, our group was interested in understanding if we could predict a game's review score using reviews from popular review sites. We scraped three major websites to obtain the title, genre, platform, and score of 46,958 game reviews between three sites.

A linear regression model performed poorly when predicting the score of a game as a function of gaming platform and genre. Several classification models also performed poorly when predicting if a game would receive a score of 80% or better, but KNN resulted in the best relative model.

KNN models could predict a score of 8 / 80 or better with 63-74 % accuracy between the datasets. OpenCritic data was better for predicting scores and identifying actual scores for games scored greater than 8. Metacritic and Gamespot were better for predicting scores and identifying actual scores for games scored less than 8.

In the future, we would like to get more and continuous factors to create a more accurate linear regression. It would also be useful to create models that hone in specific genres or platforms and predict the score for each respective one to provide a granular insight into future reviews. Given that not all reviews are created equal, we are keenly interested in applying natural language processing techniques to evaluate review text to understand the topics and sentiment that comprises the various reviews and their styles.

**References**

List of video games notable for negative reception. (2020, December 12). Retrieved December 14, 2020, from <https://en.wikipedia.org/wiki/List_of_video_games_notable_for_negative_reception>

Movie Reviews, TV Reviews, Game Reviews, and Music Reviews. (n.d.). Retrieved December 14, 2020, from <https://www.metacritic.com/>

Plante, C. (2018, September 04). Polygon is updating its reviews program for 2018 - and saying farewell to scores. Retrieved December 14, 2020, from <https://www.polygon.com/reviews/2018/9/4/17689100/polygon-reviews-no-scores>

Tassi, P. (2020, June 21). 'The Last Of Us Part 2' Is Getting Predictably User Score Bombed On Metacritic. Retrieved December 14, 2020, from <https://www.forbes.com/sites/paultassi/2020/06/21/the-last-of-us-part-2-is-getting-predictably-user-score-bombed-on-metacritic/?sh=6a7c86f85c25>

The top critics in gaming. All in one place. (n.d.). Retrieved December 14, 2020, from <https://opencritic.com/>

Video Games Reviews &amp; News. (n.d.). Retrieved December 14, 2020, from <https://www.gamespot.com/>

**Appendix:**

**OpenCritic**

1. Linear model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scores(Train)** | **Scores(Test)** | **Intercept** | **Coefficients** | **Cross Eval. Avg** |
| 0.0029 | 0.0016 | 89.153 | [0.0283,-0.2453] | -0.1 |

1. Logistic regression

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1 - Score** | **Support** |
| **0** | 0.11 | 0.00 | 0.00 | 880 |
| **1** | 0.66 | 1.00 | 0.79 | 1726 |
| **Accuracy** |  |  | 0.66 | 1523 |
| **Macro Avg** | 0.39 | 0.50 | 0.40 | 1523 |
| **Weighted Avg** | 0.48 | 0.66 | 0.53 | 1523 |

1. KNN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1 - Score** | **Support** |
| **0** | 0.34 | 0.19 | 0.25 | 880 |
| **1** | 0.64 | 0.81 | 0.73 | 1726 |
| **Accuracy** |  |  | 0.60 | 1523 |
| **Macro Avg** | 0.50 | 0.50 | 0.49 | 1523 |
| **Weighted Avg** | 0.56 | 0.60 | 0.57 | 1523 |

1. Decision Trees

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1 - Score** | **Support** |
| **0** | 0.45 | 0.09 | 0.15 | 880 |
| **1** | 0.67 | 0.94 | 0.78 | 1726 |
| **Accuracy** |  |  | 0.65 | 1523 |
| **Macro Avg** | 0.56 | 0.52 | 0.47 | 1523 |
| **Weighted Avg** | 0.59 | 0.65 | 0.57 | 1523 |

1. SVC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1 - Score** | **Support** |
| **0** | 0.44 | 0.08 | 0.14 | 880 |
| **1** | 0.67 | 0.95 | 0.78 | 1726 |
| **Accuracy** |  |  | 0.62 | 1523 |
| **Macro Avg** | 0.56 | 0.52 | 0.48 | 1523 |
| **Weighted Avg** | 0.57 | 0.62 | 0.55 | 1523 |

1. Linear SVC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1 - Score** | **Support** |
| **0** | 0.00 | 0.00 | 0.00 | 578 |
| **1** | 0.62 | 1.00 | 0.77 | 945 |
| **Accuracy** |  |  |  | 1523 |
| **Macro Avg** | 0.31 | 0.50 | 0.38 | 1523 |
| **Weighted Avg** | 0.39 | 0.62 | 0.48 | 1523 |

**Metacritic**

1. Linear model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scores(Train)** | **Scores(Test)** | **Intercept** | **Coefficients** | **Cross Eval.Avg** |
| 0.00035 | 0.00075 | 0.2318 | [-0.0004, -0.0001] | -0.12 |

1. Logistic regression

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1 - Score** | **Support** |
| **0** | 0.77 | 1.00 | 0.87 | 3676 |
| **1** | 0.00 | 0.00 | 0.00 | 1082 |
| **Accuracy** |  |  | 0.77 | 4758 |
| **Macro Avg** | 0.39 | 0.50 | 0.44 | 4758 |
| **Weighted Avg** | 0.60 | 0.77 | 0.67 | 4758 |

1. KNN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1 - Score** | **Support** |
| **0** | 0.77 | 0.99 | 0.87 | 3676 |
| **1** | 0.35 | 0.03 | 0.05 | 1082 |
| **Accuracy** |  |  | 0.77 | 4758 |
| **Macro Avg** | 0.56 | 0.51 | 0.46 | 4758 |
| **Weighted Avg** | 0.68 | 0.77 | 0.68 | 4758 |

1. Decision Trees

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1 - Score** | **Support** |
| **0** | 0.77 | 1.00 | 0.87 | 3676 |
| **1** | 0.50 | 0.01 | 0.01 | 1082 |
| **Accuracy** |  |  | 0.77 | 4758 |
| **Macro Avg** | 0.64 | 0.50 | 0.44 | 4758 |
| **Weighted Avg** | 0.71 | 0.77 | 0.67 | 4758 |

1. SVC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1 - Score** | **Support** |
| **0** | 0.77 | 1.00 | 0.87 | 3676 |
| **1** | 0.00 | 0.00 | 0.00 | 1082 |
| **Accuracy** |  |  | 0.77 | 4758 |
| **Macro Avg** | 0.39 | 0.50 | 0.44 | 4758 |
| **Weighted Avg** | 0.60 | 0.77 | 0.67 | 4758 |

1. Linear SVC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1 - Score** | **Support** |
| **0** | 0.77 | 1.00 | 0.87 | 3676 |
| **1** | 0.00 | 0.00 | 0.00 | 1082 |
| **Accuracy** |  |  | 0.77 | 4758 |
| **Macro Avg** | 0.39 | 0.50 | 0.44 | 4758 |
| **Weighted Avg** | 0.60 | 0.77 | 0.67 | 4758 |

**GameSpot**

1. Linear model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Scores(Train)** | **Scores(Test)** | **Intercept** | **Coefficients** | **Cross Eval.Avg** |
| 0.0028 | 0.0038 | 6.802 | [0.0022, 0.0157] | -0.009 |

1. Logistic regression

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1 - Score** | **Support** |
| **0** | 0.66 | 1.00 | 0.80 | 1722 |
| **1** | 0.00 | 0.00 | 0.00 | 884 |
| **Accuracy** |  |  | 0.66 | 2606 |
| **Macro Avg** | 0.33 | 0.50 | 0.40 | 2606 |
| **Weighted Avg** | 0.44 | 0.66 | 0.53 | 2606 |

1. KNN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1 - Score** | **Support** |
| **0** | 0.66 | 0.91 | 0.77 | 1722 |
| **1** | 0.35 | 0.10 | 0.15 | 884 |
| **Accuracy** |  |  | 0.63 | 2606 |
| **Macro Avg** | 0.51 | 0.50 | 0.46 | 2606 |
| **Weighted Avg** | 0.56 | 0.63 | 0.56 | 2606 |

1. Decision Trees

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1 - Score** | **Support** |
| **0** | 0.67 | 0.96 | 0.79 | 1722 |
| **1** | 0.48 | 0.07 | 0.12 | 884 |
| **Accuracy** |  |  | 0.66 | 2606 |
| **Macro Avg** | 0.57 | 0.51 | 0.45 | 2606 |
| **Weighted Avg** | 0.60 | 0.51 | 0.56 | 2606 |

1. SVC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1 - Score** | **Support** |
| **0** | 0.67 | 0.97 | 0.79 | 1722 |
| **1** | 0.54 | 0.06 | 0.11 | 884 |
| **Accuracy** |  |  | 0.66 | 2606 |
| **Macro Avg** | 0.60 | 0.52 | 0.45 | 2606 |
| **Weighted Avg** | 0.63 | 0.66 | 0.56 | 2606 |

1. Linear SVC

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1 - Score** | **Support** |
| **0** | 0.66 | 1.00 | 0.80 | 1722 |
| **1** | 0.00 | 0.00 | 0.00 | 884 |
| **Accuracy** |  |  |  | 2606 |
| **Macro Avg** | 0.33 | 0.50 | 0.40 | 2606 |
| **Weighted Avg** | 0.44 | 0.66 | 0.53 | 2606 |