What Can Predict a Game’s Userscore?

Aiden Sutherland

2024-08-06

Warning: package 'here' was built under R version 4.3.3

Warning: package 'knitr' was built under R version 4.3.3

# 1. Summary/Abstract

In this project, I explored factors influencing the user scores of video games and attempted to predict these scores using various predictors. The data were collected from multiple sources, including a TidyTuesday dataset, a Kaggle dataset, and a manually curated dataset. I focused on the top 25 games based on user scores, owners, and playtime metrics, resulting in a final dataset of 56 observations with 13 variables.

I aimed to determine the predictors of a game’s user score and whether these scores could be accurately predicted. My analysis included data cleaning, merging, and exploratory data analysis (EDA) to understand the relationships between variables. I then applied linear regression and Random Forest models to predict user scores, with a cross-validated Random Forest model showing the best performance.

The Random Forest model, after 10-fold cross-validation, achieved an RMSE of 0.3968 and an MAE of 0.2896, indicating effective prediction of user scores based on the included predictors. The results suggest a promising approach for predicting game user scores and provide insights into the factors contributing to higher scores.

Despite limitations such as a small dataset and potential predictor biases, the project highlights the potential of machine learning in understanding and predicting user preferences in video games. Future work could expand the dataset and explore additional predictors to enhance the model’s accuracy and applicability.

# 2. Introduction

## 2.1 General Background Information

I have long asked myself what makes a good video game, and if certain video games are worth the prices they are being sold for. There have been many instances where I spent a 60 dollars on a heaping pile of garbage, and then spent 10 dollars on some of the best fun I have had at a computer screen. Is a game defined by the amount of replay-ability it has or the cost:quality ratio? I wanted to find out what made people decide what made a game “good”.

## 2.2 Description of data and data source

The data I have collected stems from a few sources, one dataset is from a [tidy-tuesday exercise](https://github.com/rfordatascience/tidytuesday/blob/master/data/2019/2019-07-30/video_games.csv), one dataset from [kaggle](https://www.kaggle.com/datasets/destring/metacritic-reviewed-games-since-2000), and another I personally went through and gathered on the internet (the class-game.csv in the data folder.) This data holds a range of data going from overall metascore to median playtime on a video game while also including the development team and the publisher.

## 2.3 Questions/Hypotheses to be addressed

What effects a game’s user score and can the user score be predicted?

There was a project that was done with similar data that can be found clicking [here](https://towardsdatascience.com/analyzing-video-games-data-in-r-1afad7122aab) done by Hamza Rafiq, however I looked for data that added one more predictor that (given recent trends in the video game market) I was sure would be a predictor.

# 3. Methods

I found 2 different data sets that contained different kinds of sentiment about certain video game titles, 1 by metascore, and one by userscore. I then combined them, found the top 25 games for userscore, amount of owners, median playtime, and average playtime then put them in the final dataset. Overall there were 56 obs with 13 variables. I later added another column with the game class (with 2 factors, one being “AAA” and the other being “indie”)

## 3.1 Data Acquisition

When it came to obtaining the data, the project mentioned before done by Hamza Rafiq was very helpful in obtaining the data for this project, however when it came to obtaining data for the game class variable, I had to manually search up the classes for each game after narrowing down the data to the top 25 games for each category. The website that was integral in getting this data can be found [here](https://www.moddb.com/search?), just type in any game and all the info you would want will pop up.

## 3.2 Data import and cleaning

In the Data folder of this repository, there are 2 data sets that are the ground work for the whole project. One I imported from a different GitHub repo that can be found in the sources folder (I also downloaded it just in case) and the other is from Kaggle, which I downloaded directly from the site and imported it manually. The cleaning process involved me joining the 2 data sets, making sure there were no duplicates of any kind in any variable (the publisher column required me to change anything that contained Warner Brothers to just Warner Brothers). From then I broke down the final joined data set to the top 25 publishers and games based on user score and units sold. I then joined my class\_game.csv to the top 25 games and used that as the final dataset.

## 3.3 Statistical analysis

For the statistical analysis, I split the cleaned\_games.RData into training and test datasets, then trained and fit a random forest and linear regression model.

# 4. Results

## 4.1 Exploratory/Descriptive analysis

[Figure 1](#fig-result) shows a boxplot comparing game userscore by different game development classes. There seems to be a slight trend leaning towards AAA games having a higher userscore.

|  |
| --- |
| Figure 1: USerscore by game development class. |

[Figure 2](#fig-result2) shows a scatter plot showing average playtime compared to user score. I separated the 2 different game classes by color given that at the beginning of this project I was so sure that this was going to be a predictor. There seemed to be a clear negative correlation indicating the higher the score, the lower the average playtime.

|  |
| --- |
| Figure 2: USer Scores by Average playtime |

[Figure 3](#fig-result3) shows a scatterplot figure of user scores vs meta scores. There is a clear positive linear relationship between these 2 variables.

|  |
| --- |
| Figure 3: MetaScore by UserScore |

For more visualizations for other possible predictors, please refer to the eda.qmd file.

Here is a summary table of the predictors.

load("~/GitHub/Sutherland-PII-project/code/eda-code/first\_model.RData")  
glm\_summary <- summary(glm\_model)  
  
# Extract coefficients  
coefficients <- glm\_summary$coefficients  
  
# Convert to a data frame for better manipulation  
coeff\_df <- as.data.frame(coefficients)  
  
kable(coeff\_df, digits = 4, caption = "Summary of GLM coefficients")

Summary of GLM coefficients

|  | Estimate | Std. Error | t value | Pr(>|t|) |
| --- | --- | --- | --- | --- |
| (Intercept) | -261.5250 | 85.7016 | -3.0516 | 0.0554 |
| classindie | 196.6554 | 62.5253 | 3.1452 | 0.0515 |
| average\_playtime | -0.0132 | 0.0043 | -3.0345 | 0.0561 |
| metascore.x | 0.0457 | 0.0126 | 3.6328 | 0.0359 |
| price | 9.1478 | 2.9094 | 3.1442 | 0.0515 |
| owners | 0.0000 | 0.0000 | 0.8236 | 0.4706 |
| median\_playtime | 0.0056 | 0.0031 | 1.7701 | 0.1749 |
| developerAvalanche Studios, Feral Interactive (Mac), Feral Interactive (Linux) | 100.1761 | 32.1217 | 3.1186 | 0.0525 |
| developerBig Huge Games, 38 Studios | 91.0206 | 29.6980 | 3.0649 | 0.0548 |
| developerBioWare | 83.1080 | 26.5817 | 3.1265 | 0.0522 |
| developerBohemia Interactive | -2.4522 | 0.6616 | -3.7063 | 0.0341 |
| developerCD PROJEKT RED | -6.0563 | 2.2918 | -2.6426 | 0.0775 |
| developerColossal Order Ltd. | -4.8384 | 2.2063 | -2.1930 | 0.1159 |
| developerConcernedApe | -64.3381 | 20.5118 | -3.1366 | 0.0518 |
| developerCrate Entertainment | 47.3048 | 14.6821 | 3.2219 | 0.0485 |
| developerCrowbar Collective | -113.0518 | 35.8617 | -3.1524 | 0.0512 |
| developerDodge Roll | -64.1329 | 20.5862 | -3.1153 | 0.0527 |
| developerExpansive Worlds | 97.3023 | 31.0515 | 3.1336 | 0.0519 |
| developerFacepunch Studios | -240.7257 | 75.8657 | -3.1731 | 0.0504 |
| developerGas Powered Games | 147.8665 | 47.0856 | 3.1404 | 0.0516 |
| developerGone North Games | -31.1850 | 11.8720 | -2.6268 | 0.0785 |
| developerGrimlore Games, THQ Nordic | -182.3277 | 57.8098 | -3.1539 | 0.0511 |
| developerGSC Game World | 99.9851 | 31.8982 | 3.1345 | 0.0519 |
| developerHaemimont Games | -5.0764 | 1.1041 | -4.5977 | 0.0193 |
| developerLucasArts | 175.0968 | 55.7738 | 3.1394 | 0.0517 |
| developerOverhype Studios | -177.5800 | 56.1538 | -3.1624 | 0.0508 |
| developerParadox Development Studio | -86.4723 | 27.6740 | -3.1247 | 0.0523 |
| developerPlatinumGames, Little Stone Software | 84.0932 | 26.7810 | 3.1400 | 0.0517 |
| developerRe-Logic | -15.1921 | 5.0994 | -2.9792 | 0.0586 |
| developerRed Hook Studios | -155.7398 | 49.3506 | -3.1558 | 0.0510 |
| developerRelic Entertainment | 238.0692 | 75.7882 | 3.1412 | 0.0516 |
| developerRemedy Entertainment | 175.1284 | 55.7579 | 3.1409 | 0.0516 |
| developerRockstar Games | 132.6040 | 42.1904 | 3.1430 | 0.0515 |
| developerSEGA | 98.6794 | 30.9556 | 3.1878 | 0.0498 |
| developerTango Gameworks | -282.0103 | 89.6551 | -3.1455 | 0.0514 |
| developerThe Sims Studio | 106.7330 | 34.2863 | 3.1130 | 0.0528 |
| developerTripwire Interactive | -6.1639 | 1.8606 | -3.3128 | 0.0453 |
| developerTroika Games | 83.8928 | 26.7542 | 3.1357 | 0.0518 |
| developerUbisoft Montreal | 87.5056 | 28.1868 | 3.1045 | 0.0531 |
| developerValve | 175.6198 | 55.7911 | 3.1478 | 0.0514 |

## 4.2 Basic statistical analysis

Below are the statistically significant predictors from the glm model ran on the data (>.05).

# Filter for statistically significant results (p-value < 0.05)  
significant\_coeff\_df <- coeff\_df[coeff\_df$`Pr(>|t|)` < 0.05, ]  
kable(significant\_coeff\_df, digits = 4, caption = "Statistically Significant Coefficients from GLM")

Statistically Significant Coefficients from GLM

|  | Estimate | Std. Error | t value | Pr(>|t|) |
| --- | --- | --- | --- | --- |
| metascore.x | 0.0457 | 0.0126 | 3.6328 | 0.0359 |
| developerBohemia Interactive | -2.4522 | 0.6616 | -3.7063 | 0.0341 |
| developerCrate Entertainment | 47.3048 | 14.6821 | 3.2219 | 0.0485 |
| developerHaemimont Games | -5.0764 | 1.1041 | -4.5977 | 0.0193 |
| developerSEGA | 98.6794 | 30.9556 | 3.1878 | 0.0498 |
| developerTripwire Interactive | -6.1639 | 1.8606 | -3.3128 | 0.0453 |

## 4.3 Full analysis

load("~/GitHub/Sutherland-PII-project/code/analysis-code/rf\_metrics.RData")  
load("~/GitHub/Sutherland-PII-project/code/analysis-code/rf\_cv\_metrics.RData")  
load("~/GitHub/Sutherland-PII-project/code/analysis-code/lm\_metrics.RData")

For the first model, after splitting the game data into test and training data, I ran 2 different models to see which one has better predicting power.

Below in the linear regression model for the training data, after training the model and making predictions, here are the MAE and RMSE statistics for the LM model as well as a plot for observed and predicted models

print(paste("RMSE: ", rmse\_lm ))

[1] "RMSE: 0.547575922934759"

print(paste("MAE: ", mae\_lm))

[1] "MAE: 0.416483516483517"

print(paste("R-squared: ", r\_squared\_lm))

[1] "R-squared: 0.254962972742657"

|  |
| --- |
| Figure 4: LM model predictions Vs observed |

Here is the Random Forest models MAE and RMSE statistic with the observed vs predicted plot.

print(paste("RMSE: ", rmse\_rf))

[1] "RMSE: 0.420145975206875"

print(paste("MAE: ", mae\_rf))

[1] "MAE: 0.305963809523809"

print(paste("R-squared: ", r\_squared\_rf))

[1] "R-squared: 0.561378834500798"

|  |
| --- |
| Figure 5: RF model predictions Vs observed |

Due to the RF model having a lower RMSE and MAE, while also having a higher r-squared, I proceeded to cross-validate the Random Forest model. Below is the results of my cross validation.

print(paste("RMSE (Cross-validated): ", rmse\_rf\_cv))

[1] "RMSE (Cross-validated): 0.396829326393237"

print(paste("MAE (Cross-validated): ", mae\_rf\_cv))

[1] "MAE (Cross-validated): 0.289591428571427"

print(paste("R-squared: ", r\_squared\_rf\_cv))

[1] "R-squared: 0.608711855983782"

|  |
| --- |
| Figure 6: RF\_CV model predictions Vs observed |

After cross validating the Random Forest model, the r-squared value slightly increases while the MAE and RMSE slightly decreases, so the cross validated model is somewhat better than the RF model.

# 5. Discussion

## 5.1 Summary and Interpretation

We applied Random Forest regression to predict user scores of games based on 11 predictor variables. The model was trained and cross-validated using 10-fold cross-validation. The optimal configuration, with mtry = 82, was selected based on minimizing the Root Mean Square Error (RMSE). The cross-validated RMSE and Mean Absolute Error (MAE) were found to be 0.3968 and 0.2896, respectively, indicating that the model predicts user scores with an average deviation of approximately 0.3968 units and an average absolute error of 0.2896 units. These results suggest that the Random Forest approach effectively captures the relationships between predictors and user scores, demonstrating promising performance in predicting game user scores.

## 5.2 Strengths and Limitations

One extreme limitation to this project was the fact that I only had the top 25 game classes. Provided I had the time (between classes and work) to match all of the games mentioned with its respective class I feel that the models would fit a lot better. One strength I could always rely on was how detailed the data could get. Depending on how you clean/wrangle it you could use the data for a multitude of projects that involve different hypothesis. Again, i belive having more data to go through in this project could conclude more concrete models for prediction.

## 5.3 Conclusions

Given the r-squared value of the cross validated Random Forest Model, we can conclude that we can roughly conclude the userscore from the 11 predictor variables in the games.csv file. Some of the predictors (like owners and metascore) would need to be predicted if you were to want to predict the userscore of an unreleased game. Overall this project was really fun due to it pertaining to a topic I find truly fascinating being a video game consumer myself. In the future I would probably look for another scoring metrics besides metascore and userscore in order to broaden consumer sentiment.

# 6. References

tidy-tuesday exercise - (https://github.com/rfordatascience/tidytuesday/blob/master/data/2019/2019-07-30/video\_games.csv) kaggle Metacritic games - (https://www.kaggle.com/datasets/destring/metacritic-reviewed-games-since-2000) Analyzing Video Games Data in R, Hamza Rafiq - https://towardsdatascience.com/analyzing-video-games-data-in-r-1afad7122aab