Developing and Evaluating a Video Game Rating Model

Kevin Camp

2022-11-22

1.0 Introduction

This report satisfies the guidelines for the Choose Your Own project in the Data Science Capstone course. My project goal is to analyzing a dataset of video game ratings by users in order to create a recommendation system. The recommendation system will predict ratings by user for games that are missing user ratings in the dataset. The dataset is large; that presents a challenge, in that large datasets are difficult to manage. It also presents an opportunity, in that every data point can be used as a predictor for each missing value in the overall set. The aim of the project is to achieve that using machine learning methods.

1.1 Overview

The project uses data from this repository split into two sets. One set—"vg"—is used to train the model, and contains more than 15 thousand rows of video games. Each row provides columns detailing the name of the game (Name); the platform on which the game was released (Platform); the year of release (Year_of_Release); publisher (Publisher); five columns of sales data, including North America, European Union, Japan, Other, and Global; critic ratings (Critic_Score); critic rating count (Critic_Count); user ratings (User_Score); user rating counts (User_Count); developer (Developer); and rating (Rating). The other set—"validation"—is similar in content to vg, and is used as the final hold-out test set to see how well the model performs. Table 1.1 presents a selection of rows from vg.

Table 1.1. First ten rows of vg dataset

Name	Platform	Year_of_Release	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Critic_Score	Critic_Count	User_Score	User_Count	Developer	Rating
Wii Sports	Wii	2006	Sports	Nintendo	41.36	28.96	3.77	8.45	82.53	76	51	8	322	Nintendo	E
Super Mario Bros.	NES	1985	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24	NA	NA		NA		
Wii Sports Resort	Wii	2009	Sports	Nintendo	15.61	10.93	3.28	2.95	32.77	80	73	8	192	Nintendo	E
Pokemon Red/Pokemon Blue	GB	1996	Role-Playing	Nintendo	11.27	8.89	10.22	1.00	31.37	NA	NA		NA		
Tetris	GB	1989	Puzzle	Nintendo	23.20	2.26	4.22	0.58	30.26	NA	NA		NA		
New Super Mario Bros.	DS	2006	Platform	Nintendo	11.28	9.14	6.50	2.88	29.80	89	65	8.5	431	Nintendo	E
Wii Play	Wii	2006	Misc	Nintendo	13.96	9.18	2.93	2.84	28.92	58	41	6.6	129	Nintendo	E
New Super Mario Bros. Wii	Wii	2009	Platform	Nintendo	14.44	6.94	4.70	2.24	28.32	87	80	8.4	594	Nintendo	E
Duck Hunt	NES	1984	Shooter	Nintendo	26.93	0.63	0.28	0.47	28.31	NA	NA		NA		
Nintendogs	DS	2005	Simulation	Nintendo	9.05	10.95	1.93	2.74	24.67	NA	NA		NA		

In table 1.2, the summary statistics on the vg set reveal almost 12 thousand unique games. This number is less than the total rows because a number of games are released on multiple platforms (*e.g*, Xbox, Playstation).

Table 1.2. Summary of vg set

Rows	Columns	Unique games	Unique platforms	Average global sales	Average critic score	Average user score	Number of genres
15929	16	11563	31	0.53	68.94	7.14	13

The vg data contains other useful pieces of information for creating a game recommendation algorithm. For example, the year a given game was released. The rating submission dates in the vg set range from 1980 to 2020.

Another example is the genre information for each game. Most games in the vg set have one genre listed. In fact, only two entries the dataset—rated by 0 users/critics—have no genre (nor even a name) listed. These variables, along with the others previously mentioned, will be useful predictors in the model.

Table 1.3. Rows in vg lacking genre information

Name	Platform	Year_of_Release	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Critic_Score	Critic_Count	User_Score	User_Count	Developer	Rating
	GEN	1993		Acclaim Entertainment	1.78	0.53	0.00	0.08	2.39	68.94356	NA	7.142462	NA		
	GEN	1993		Acclaim Entertainment	0.00	0.00	0.03	0.00	0.03	68.94356	NA	7.142462	NA		

1.2 Executive summary

Through data science techniques, we can create a strong prediction model capable of recommending games to individuals. This can be beneficial for providing entertainment suggestions to individuals. An important question is: how can we model this efficiently with a large dataset? Machine learning methods are capable of achieving this, and that is the strategy I pursue in this report.

In the sections that follow I discuss my methods, clean and organize my data, develop my model, and finally evaluate the predictions generated by the model. I find success in predicting game ratings based on five variables present in the dataset. The result is a suitably low measure of difference between the predictions my model generates and the actual rating values in a tranche of the dataset.

2.0 Methods and analysis

For this project, my goal is to develop a video game recommendation model. This model will aim to accurately predict game ratings in the final hold-out test set as if they were not known to me. The means of evaluating my success will be minimizing root mean squared error (RMSE). The RMSE is the standard deviation of the residuals, a common measurement used in statistics to evaluate accuracy of projections. This project has a defined target of reaching the lowest possible RMSE value given the available data.

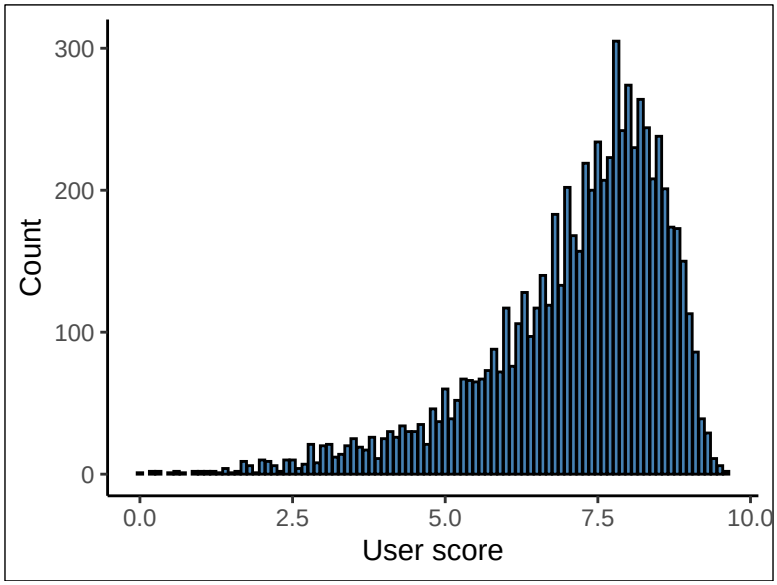
The key dependent variable of interest for recommending games is the user score. This rating measures how much a user enjoyed a given game. In the vg data, ratings range from a minimum of 0 to a maximum of 9.6. The overall mean value for the game ratings in the dataset is 7.14, and the standard deviation is 0.99. Table 2.1 displays the number of observations (n), mean, standard deviation (sd), minimum and maximum ratings, the range of the rating scale, and the standard error (se).

Table 2.1. Summary statistics for user scores

n	mean	sd	min	max	range	se
15929	7.142462	0.99253	0	9.6	9.6	0.0078641

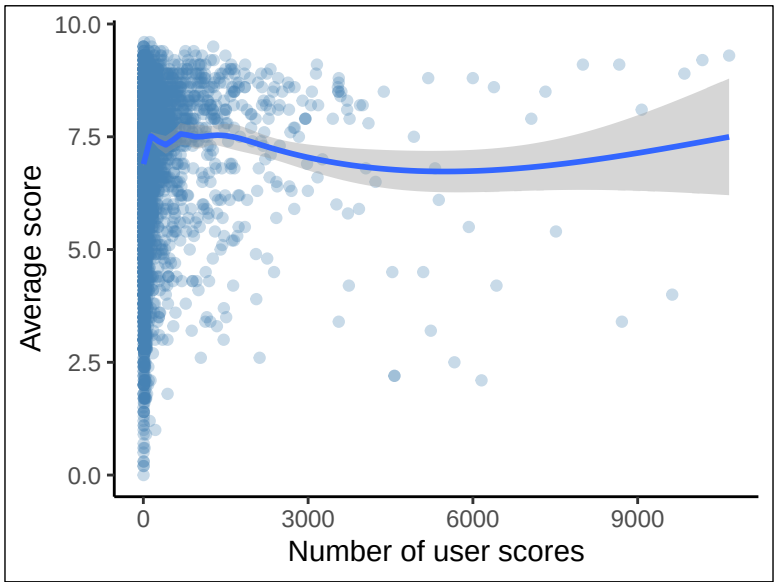
Further exploring the data reveals additional patterns in user ratings. As shown in figure 2.1, we see that the most common rating is 7.8. Additionally, the data appear to be left-skewed. A left-skewed distribution, as shown by the long left tail in figure 2.1, indicates that ratings concentrate on the right side of the tail, such that reviewers tend to give high ratings to games.

Figure 2.1. Count by user score in vg



As shown in figure 2.2, most games have very few scores. Additionally, there does not appear to be any real trend in the data. Games with few ratings, have scores both low and high, and similarly, games with many ratings have both high and mediocre scores.

Figure 2.2. Count by user score in vg



Accurate prediction of game scores provides justification for recommending games. In brief, if my model predicts a game will receive a score of 1, I would not encourage someone using this recommendation system to play the game. The reverse is true if my model predicts a game will receive a score of 9.

2.1 Techniques and processes

My techniques and processes will include the following: 1. cleaning the data, 2. exploring and visualizing the data using the tidyverse package, 3. using insights gained to develop a model concept, 4. creating the model, and 5. evaluating the model on the validation set by measuring the RMSE.

2.1.1 Data cleaning

First, I update vg, multiplying user score by 10 to scale to the provided critic score. Therefore, the final range of user scores goes from 0 to 100. This first data cleaning effort ensures that the scores are comparable. Table 2.2 displays the first five rows of the updated vg set with scores converted.

Table 2.2. First five rows of updated vg dataset

Name	Platform	Year_of_Release	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales	Critic_Score	Critic_Count	User_Score	User_Count	Developer	Rating
Wii Sports	Wii	2006	Sports	Nintendo	41.36	28.96	3.77	8.45	82.53	76.00000	51	80.00000	322	Nintendo	E
Super Mario Bros.	NES	1985	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24	68.94356	NA	71.42462	NA		
Wii Sports Resort	Wii	2009	Sports	Nintendo	15.61	10.93	3.28	2.95	32.77	80.00000	73	80.00000	192	Nintendo	E
Pokemon Red/Pokemon Blue	GB	1996	Role-Playing	Nintendo	11.27	8.89	10.22	1.00	31.37	68.94356	NA	71.42462	NA		
Tetris	GB	1989	Puzzle	Nintendo	23.20	2.26	4.22	0.58	30.26	68.94356	NA	71.42462	NA		

2.1.2 Data exploration and visualization

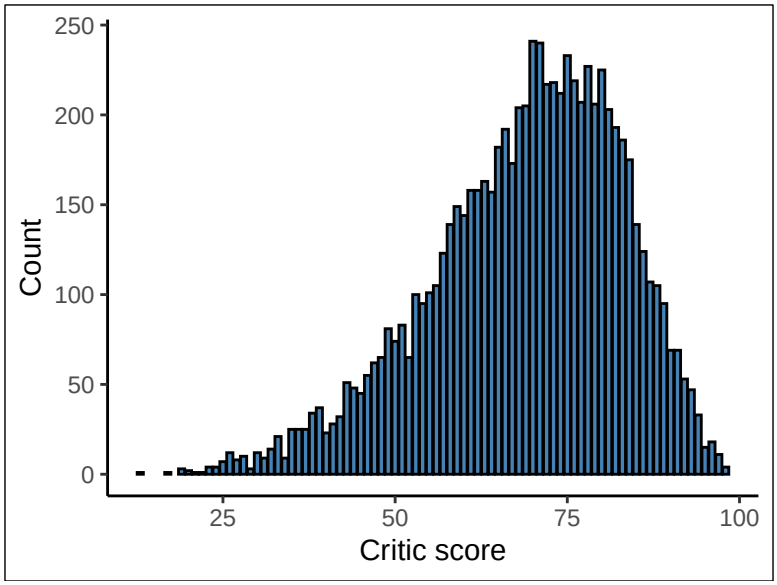
As demonstrated in table 1.2, the vg set contains 11,563 unique games. Summary statistics on ratings by critic are shown in table 2.3. The average critic score is 68.9, with a standard deviation of 9.6. Critics on average rated movies slightly lower than users, but with somewhat less spread.

Table 2.3. Summary statistics for critic scores

n	mean	sd	min	max	range	se
15929	68.94356	9.634147	13	98	85	0.0763342

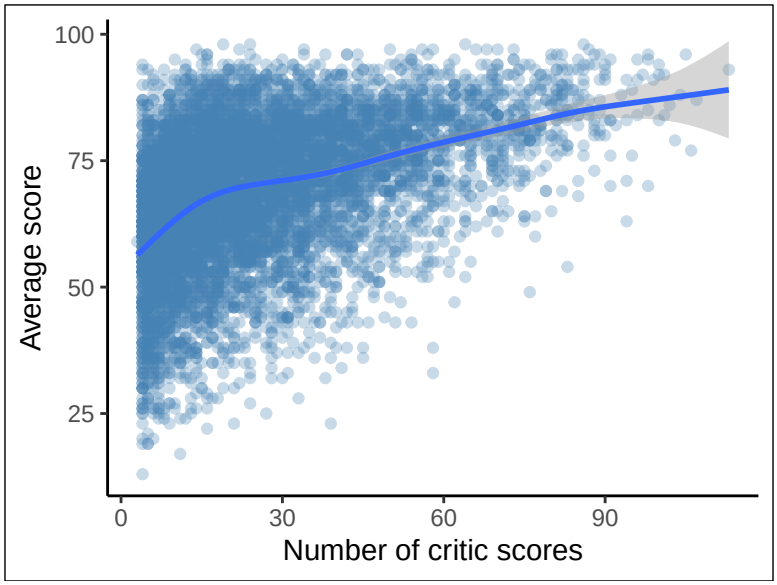
Figure 2.3 shows the distribution of critic ratings per game. Similarly to user ratings, the ratings are left-skewed. This shows that, much like users, critics tend to give higher ratings to games.

Figure 2.3. Distribution of critic scores



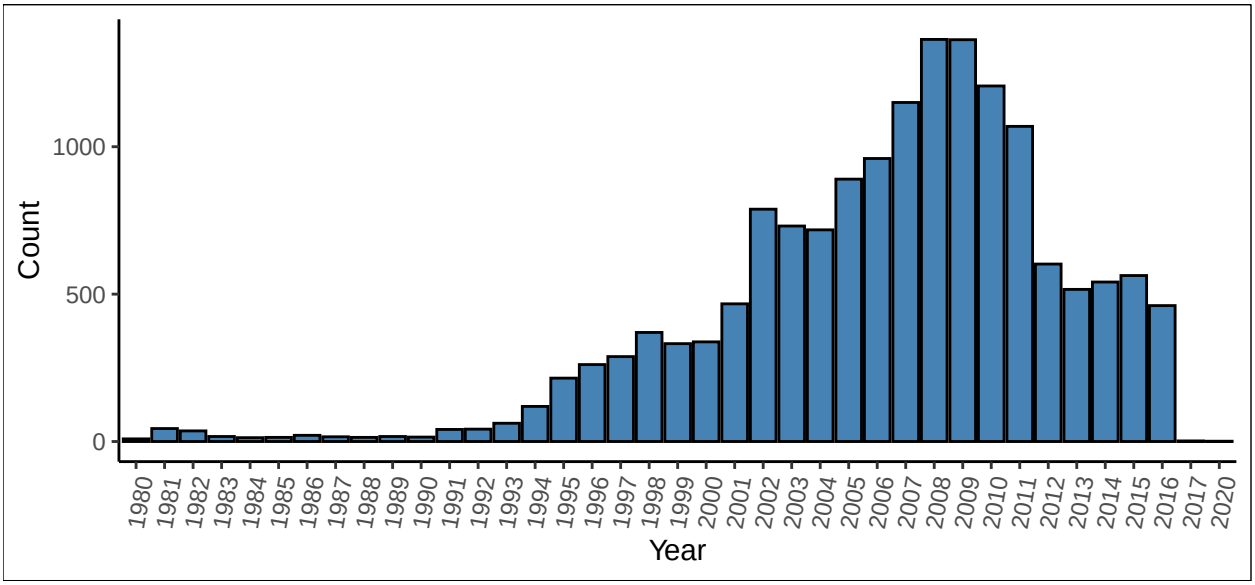
There appears to be a correlation between the number of scores that a game receives from critics, and the average score received from critics (figure 2.4). Games that garner more attention from critics are perhaps more unique, interesting, or simply better games; if so, it is not surprising that there would be a positive relationship between the number of critic scores a game receives and the average score.

Figure 2.4. Average game ratings by critics



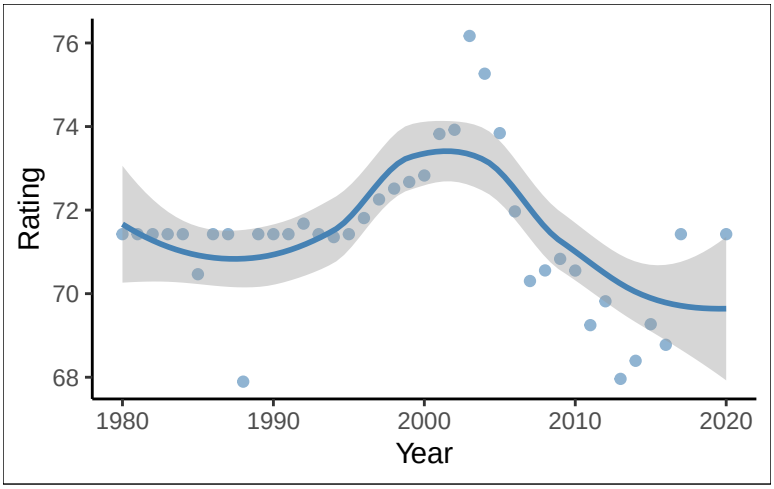
We can also investigate the number of games by release year. As we see in figure 2.5, more games were released in the late 2000s. Very few games in the dataset were released in the 1980s, slowly increasing in the mid 1990s. The number of games released in a single year peaked in the years 2008 and 2009. The number of games released by year declined at that point, and was fairly stable from 2012 to 2016.

Figure 2.5. Count of game releases by year



The average rating by year of game release appears to have a time trend. Ratings held relatively constant from the 1980s to the early 1990s, peaked in the 2000s, and have declined since. Average ratings by year range from the high 60s to the mid 70s out of 100. There are few outliers in the data.

Figure 2.5. Average ratings per year



A simple linear regression model provides more information on the relationship between release year and average rating. Two models are estimated and results reported in table 2.4. Model 1 estimates the effect of the release year and squared release year on average rating; the squared term is included to account for the non-linearity of average ratings by release year illustrated in figure 2.9. In model 2, a cubic term is added for comparison because the curve of average ratings by release year appears to have a point of inflection. The significant coefficients in both models indicate that release year impacts average rating, and that the impact varies over time.

Table 2.4. Linear models of average user score by game release year

	Model 1	Model 2
(Intercept)	-26428.64 **	-8746.11 **
	(8581.47)	(2859.47)
I(as.numeric(Year_of_Release)^2)	-0.01 **	0.01 **
	(0.00)	(0.00)
I(as.numeric(Year_of_Release))	26.54 **	
	(8.59)	
I(as.numeric(Year_of_Release)^3)		-0.00 **
		(0.00)
N	39	39
R2	0.23	0.23

*** p < 0.001; ** p < 0.01; * p < 0.05.

Note: standard errors in parentheses.

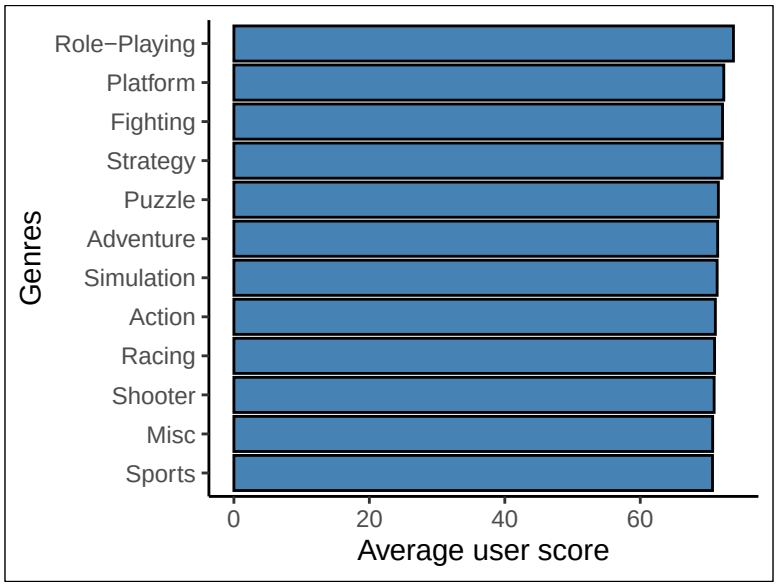
Games in the vg data are categorized into various genres. The distinct genres included in the dataset are listed in table 2.5. There are 12 unique genres (as well as the aforementioned "no genres listed" indicator, omitted from the table).

Table 2.5. Statistics by genres in vg

Genre	Count	User_mean	Critic_mean
Action	3158	71.09	67.69
Adventure	1270	71.43	68.02
Fighting	815	72.14	68.91
Misc	1682	70.69	68.31
Platform	841	72.33	68.49
Puzzle	566	71.53	68.35
Racing	1176	70.97	68.42
Role-Playing	1446	73.75	70.70
Shooter	1240	70.90	69.78
Simulation	857	71.35	68.79
Sports	2208	70.67	70.40
Strategy	668	72.09	70.32

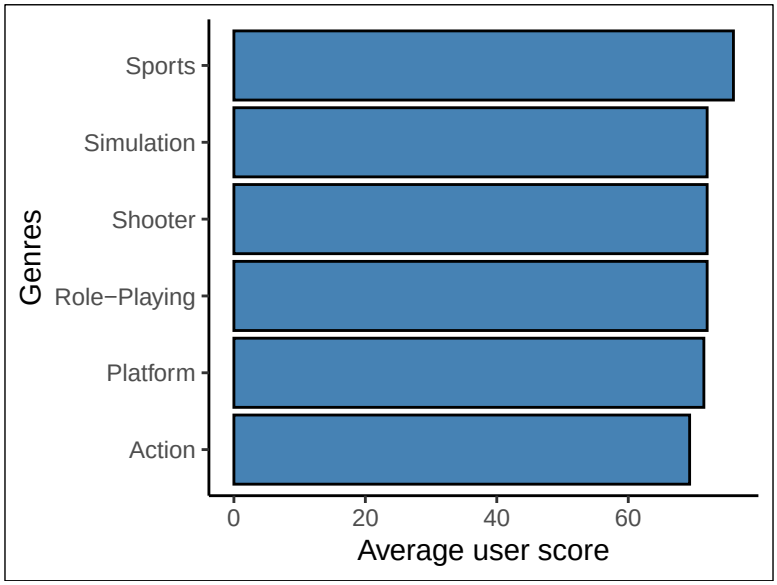
Role-playing games are the highest rated on average among the vg genres. Figure 2.6 shows they rate above 70 on average. Conversely, the lowest rated games on average belong to the sports genre. The figure provides evidence that ratings vary by genre, though minimally.

Figure 2.6. Average user score by genre



Do genre effects interact with user effects? To investigate, we can take a random sample of 10 games from vg and explore how they are rated on average across genres. In figure 2.7 we see the average ratings by genre for our random sample. The average ratings and the ranking of genres differs in figure 2.7 compared to figure 2.6. Interestingly, the highest rated genre in figure 2.7, sports, was the lowest rated genre on average in figure 2.6. This suggests that, although there may be overall favorite genres for gamers, certain users have distinct genre preferences. These genre preferences will help inform our final model.

Figure 2.7. Average score by genre, random sample of users



2.1.3 Insights gained

The exercise of data exploration and visualization resulted in several useful insights. Ratings data are unbalanced—a large number of games have very few ratings. Substantial publisher to publisher variation is

present in the vg data. Certain publishers release games that are rated higher than their counterparts. The ratings of games are correlated with their release years. Finally, genres matter for game video game ratings overall overall.

2.2 Modeling approach

My approach to developing the model is to begin with a naive attempt using a simple average. From there, I will tune my model, adding predictors incrementally in an effort to minimize RMSE. I plan to use predictors based on the insights detailed above, to ultimately produce a model that accounts for various unique characteristics of each game.

Values of the key dependent variable, ratings, in vg are discrete. In other words ratings values are measured by counting. One improvement in my model is that it will allow for ratings predictions that are continuous. The result should be a more sensitive measure and, by extension, a lower RMSE.

3.0 Results

In this section I develop my model to minimize the RMSE of predicted game ratings and actual ratings in the vg dataset, then present the results of my model. As previously described, I adopt an iterative approach, adding predictors to the model one-by-one.

3.1 Model results

The most straightforward model for a video game recommendation system would simply predict the same rating for each game, not taking into account unique characteristics of the game, its sales numbers, or the year the game was released. Such a model would use the average to minimize the RMSE. With the vg data, a model using only a simple average of game ratings results in a RMSE of 9.9250, as shown in table 3.1.

Table 3.1. RMSE by method I

method	RMSE
Only estimate is the average	9.925

The RMSE of 9.9250 is somewhat high—it would translate to regularly missing ratings predictions by nearly 10 percentage points. We can refer back to table 2.1 to confirm that our current model RMSE is equal to the standard deviation of user scores in vg. A reasonable goal is to improve upon this RMSE. In an effort to do so, we can tweak the model so that instead of using a simple average as the estimate, we incorporate the critic score average. Table 3.2 adds a new row to report the RMSE for our critic model.

Table 3.2. RMSE by method II

method	RMSE
Only estimate is the average	9.9250
Add critic–specific effect	8.4753

We see that our updated model has improved RMSE, lowering it to 8.4753. The next step is to account for platform-specific effects. Table 3.3 adds another new row to show the results of the model incorporating an estimate for the platform effect.

Table 3.3. RMSE by method III

method	RMSE
Only estimate is the average	9.9250
Add critic–specific effect	8.4753
Add platform effect	8.1933

Incorporating an estimate for the effect of platforms improves our model performance once again. Now we need to address the variation of ratings over time in our data. We will add a time effect using the variable of a game's release year. The resulting model RMSE is provided in table 3.4.

Table 3.4. RMSE by method IV

method	RMSE
Only estimate is the average	9.9250
Add critic–specific effect	8.4753
Add platform effect	8.1933
Add year effect	8.4003

The measure of release year interestingly represents an setback to the model, increasing RMSE from 8.1933 to 8.4003. The logical final step to round out the model is adding a treatment for game sales. The result of incorporating sales effects into the model is depicted in Table 3.5.

Table 3.5. RMSE by method V

method	RMSE
Only estimate is the average	9.9250
Add critic–specific effect	8.4753
Add platform effect	8.1933
Add year effect	8.4003
Add sales effect	8.2021

Finally, we model the effect of publishers on user ratings. Certain publishers could have a knack for making more popular, better-received games. Table 3.6 reveals these results and the RMSE decrease associated with added publisher controls.

Table 3.6. RMSE by method VI

method	RMSE
Only estimate is the average	9.9250
Add critic–specific effect	8.4753
Add platform effect	8.1933
Add year effect	8.4003
Add sales effect	8.2021
Add publisher–effect	7.9849

With all variables included, our model results in an RMSE of 7.9849 on the vg ratings data. The culmination of the project is to evaluate the model's performance on the final hold-out test set.

3.2 Model performance

The first step of evaluating the final model on the validation dataset is to repeat our data cleaning procedure in order to get the measure of user ratings into a numerical value and scaled with critical ratings. Then, we will append columns for the final model effects—critic, platform, sales, and publisher. Finally, we will calculate RMSE, measuring success by minimizing its value.

Before performing the final evaluation, we need to address not applicable (NA) values in the variable predictors appended to the validation set. Table 3.7 lists these. To deal with them, I convert each NA to the average value of the corresponding effect in the validation set.

Table 3.7. Check for NA values in validation set

	NAs
mu_user	0
b_c	272
b_p	0
b_y	0
b_s	12

With the NA values converted, we can evaluate the final model. Results are shown in table 3.8.

Table 3.8. Final RMSE evaluation

method	RMSE
Critic-average, platform, sales, publisher model	11.0988

My model's final RMSE is 11.0988. The effects included in the model allow us to make predictions that are reasonably close to a given user score out of 100.

4.0 Conclusion

This project involved making predictions on video game user scores using machine learning techniques. The contribution of this work is demonstrating how simple methods can result in strong predictive power on variables in large datasets. Individuals who want to entertain themselves with video games could find this study of interest. But there are many other potential practical applications for the techniques used in this analysis. I will discuss those after summarizing the report and addressing the limitations of this work.

4.1 Report summary

With the aim of developing an accurate video game recommendation model, I performed analysis on a large dataset. The data were thousands of video game ratings with corresponding information on number of users giving ratings, critical reception, release time, genre, sales figures, and publishers. My efforts involved summarizing the data, cleaning the data, performing exploratory analysis and data visualization. The result was added insight which, in turn, I put to use while considering strategies for modeling video game ratings. I developed my model, electing for an iterative process to drive RMSE lower in increments. Adding in effects by critic score, genre, platform, publisher, and sales allowed my model to account for the rich variability observed across all. Ultimately, I was able to achieve the goal of an appropriately low RMSE when evaluating my model's predictions compared to the ratings in the final hold-out test set.

4.2 Limitations

My project, while successful, is not without drawbacks. In large part my analysis was confounded by many missing user-rating values. The same was true for ratings by critics. Genre information for the video games present in the dataset was lacking, and future data collection should prioritize listing multiple genres per game. Additionally, a crucial missing data component was unique user identifiers—being able to disentangle user-specific individual effects would likely have improved my model considerably.

Aside from that, my analysis uncovered some odd patterns in the data—for example, average game ratings increasing substantially once roughly 600 weeks have passed since the first rating. This is an example of a phenomenon that is difficult to explain, and perhaps inadequately addressed in my analysis. Another possible limitation of this work involves the handling of NA values in the vg and validation sets. I opted to convert those to the column average because it was a simple choice. But it is possible that closer investigation would uncover a better way to deal with the NAs. Regardless, the large number of NAs, particularly for the key dependent variable of user scores, was a detriment to my study.

4.3 Future work

Future analysis would benefit from improved data. Even beyond what I mentioned in the previous section, there are likely more than 12 unique genres that are informative with respect to video games. In addition, it is likely that new or more advanced modeling techniques could push the RMSE down further. I am satisfied with the predictive power of my recommendation algorithm, but there is likely room for improvement in future studies.

It is easy to imagine applications of this work to areas outside of video recommendations. For example, a game recommendation system would dovetail nicely with this analysis. Other natural extension of this work could apply to shopping or eating at restaurants. Models such as mine could play a role in better understanding how people enjoy leisure time; this has both lifestyle and policy implications. Future studies should focus on these important concepts.