Video Game Final

Kashish Pandey

12/7/2021

Introduction

Video games have served as an engaging means of entertainment since the 1960s. With classic games such as Pong, Space Invaders, Galaxy Game, and Super Mario Bros, there has been a significant evolution within the quality of games over the past 60 years. The global gaming market is worth $173.70 billion as of 2020[11]; it's safe to stay there is a considerable market and industry for these games. There are so many different factors that draw people into the world of gaming, whether it be for relaxation, competition, or even gratification. Video games have proven to improve basic visual processes, enhance executive functioning, improve everyday skills (such as hand-eye coordination), and ease anxiety and depression to a certain degree[11]. These are reasons why the general population is drawn to video games, but I want to know what actually makes a high-quality video game and if it is possible to predict the global sales of games given all the features.

Overall Objectives

This project utilizes a video game dataset containing games from 1980 - 2016 from Kaggle. The first portion of this project jumps into EDA (exploratory data analysis). Essentially, I wanted to analyze the data before performing any models on it. I also wanted to fully clean up the model and plot the basic information out first to understand what was going on; I dropped missing values, rescaled the axis, fixed the scaling of variables, and mutated some columns in order to do so.

Regarding the second portion of the project, I wanted to see what types of models could best predict global sales. The variables used for the model were Critic_Score,User_Score,Genre,Year_of_Release,Critic_Count, User_Count, Rating, publisher_top, developer_top,num_of_platform while predicting global sales. I used Linear Regression, Support Vectors Machines, Lasso, Ridge, and Random Forest models. I decided to use the metric of RMSE (root mean square error) to compare these models because it was a concise way to measure the actual values versus the predicted values(the outcome). I was able to see which models were overfitting and underfitting. Through hyperparameter tuning, I found optimal values to further improve the models.

Importing Libraries

In terms of the libraries used within this project, I used a total of 8. 'Tidyverse' was used to utilize packages such as ggplot. 'Testthat' played an essential role in checking whether each of the model's RMSE was less than 2. Both the packages 'tree' and 'ranger' assisted in creating the Random Forest model. 'Caret' was used in order to create the linear regression model (baseline model). 'Elasticnet' in machine learning refers to L1+L2, and it was used within this project to develop a Lasso(L1) model and Ridge model(L2). 'Corrplot' assisted in my EDA of the data before creating any machine learning models; more specifically, it helped create a correlation matrix of global sales for each platform and genre. The final package used was 'kernlab' which assisted in creating 3 different types (linear, polynomial, and radial) of Support Vector Machine models.

Creating RMSE function for Analyzing the Models

RMSE gives more importance to the highest errors, making it more sensitive to outliers[19]. But through more complicated models like Random Forest, Ridge and SVM, we can combat both outliers and overfitting. The RMSE value provides an in-depth understanding of the distance between the actual versus predicted values. Moreover, the reason I had chosen to use the RMSE over MAE (mean absolute error), MSE (mean squared error), MAPE (mean absolute percent error), Accuracy, and others was because of the key question I wanted to answer. I wanted to answer was if it was possible to predict Global Sales given Critic Score, User Score, Genre, Year_of_Release, Critic_Count, User Count, Rating, publisher top, developer top, and num of platforms. With that in mind, when predicting a numerical output such as global sales, seeing if the model is extremely close to the true value versus not gives a good idea of the performance for this specific project.

- The equation for RMSE is as follows: $\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}$ [17]
- Within the equation, it is taking the square root of the mean of true values subtracted by the predicted values and squaring it

Different Models Used

The models I used were Linear Regression, Support Vectors Machines, Lasso, Ridge, and Random Forest. I had specifically picked these models out for a couple of reasons. First, it is essential to understand why I chose supervised machine learning over unsupervised machine learning techniques for this project. The critical difference between supervised and unsupervised machine learning stems from the input. Within supervised machine learning (what I used within this project), I am using known and labeled data as input. In addition, supervised machine learning is essentially learning from the training dataset by making predictions and adjusting for the correct answer. Whereas unsupervised machine learning uses unlabeled data as input and discovers the inherent structure of unlabeled data, this was not needed for this specific project.

In terms of the appropriate time use any of these models it simply depends on the problem you are trying to solve.

Linear Regression Model

Typically, you would use a linear regression model when you are modeling a linear relationship.The way I used linear regression for this project was to simply have it as a baseline model to generally understand the model's results. Baseline models are an important way to interpret the model with less complexity [12].

- Linear regression is a technique that predicts a single output value based on training data [12].
- The equation for linear regression is as follows: $y = \beta_0 + \beta_1 x + \varepsilon$ [18].
- β_0 refers to the intercept.
- $\beta_1 x$ refers to the slope.
- ε refers to the random error component.

Support Vector Machine Models

Support Vector Machines work best for regression and classification problems; this means that SVM can both solve linear and non-linear problems. In technical terms, a Support Vector Machine finds a hyperplane (line) in N-dimensional space(N — the number of features) that distinctly classifies the data points [14]. In this case, I used different kernel functions (polynomial and radial) within SVM to see if either of them could predict global sales more accurately.

L1 and L2 Models

Lasso and Ridge models are used to reduce model complexity and prevent over-fitting. Both Lasso and Ridge Lasso regression are essentially regularized linear regression. Compared to ridge regression(L2), within lasso regression (L1), instead of penalizing high values, the lasso model sets these values equal to zero instead. There is a chance to end up with fewer features because of the method lasso uses; lasso is essentially keeping the most important features and getting rid of the rest, and this is where lasso can have the upper hand over ridge. Whereas, in ridge regression, instead of getting rid of features that do not contribute to the model, ridge regression minimizes its impact on the trained model. Ridge keeps all the features but is only significantly impacted by the most important features. In this case, I used L1 and L2 to see which one could predict global sales more accurately.

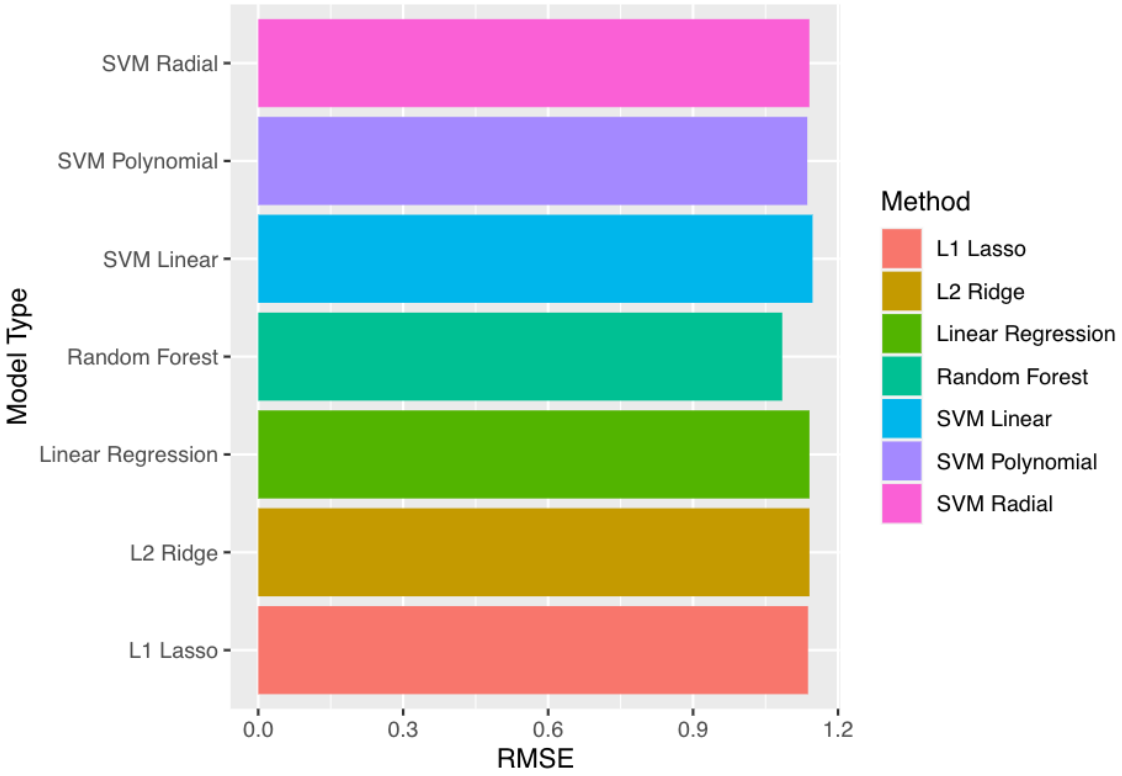
Random Forest Model

The random forest model uses a large number of individual decision trees, and each individual tree in the random forest spits out a class prediction; the class with the most votes becomes our model's prediction [15]. For my specific model, I used trainControl() because it helps specify a particular number of parameters[15].Within trainControl(), I used the method = repeatedcv because it allowed for parameters to be repeated accordingly[15]. I also used splitrule = extratrees, variance because extratrees helps us specify, and variance is used because it is the default typically[15]. Lastly, by using method = ranger because it is performing recursive partitioning(fast implementation of random forests)[15]. In this case, I fine-tuned the random forest model a bit to see if these extra features random forest has could improve prediction.

```
knitr::include_graphics('docs/all_rmse.png')
```

```
##              Method     RMSE
## 1 Linear Regression 1.140914
## 2        SVM Linear 1.147282
## 3    SVM Polynomial 1.136365
## 4        SVM Radial 1.140746
## 5          L1 Lasso 1.137891
## 6          L2 Ridge 1.140906
## 7     Random Forest 1.084544
```

```
knitr::include_graphics('docs/RMSE.png')
```



The model performance was based on the RMSE value that was outputted. The random forest model did the best compared to Linear Regression, Support Vectors, Machines(linear, polynomial, radial), Lasso(L1), and Ridge(L2) models. As mentioned before, the RMSE represents the Root Mean Squared Error. The RMSE measures the difference between predicted and observed values[19]. The random forest model getting the lowest RMSE means that it had the lowest difference between predicted and observed values. The RMSE for the random forest model was around 1.08. Overall, it makes sense as to how random forest did the best. Through of use of 'extratrees,' it is essentially creating an extremely randomized tree[16]. Each tree nod is combined with a random choice of a certain number of attributes; the best one is determined[16]. Moreover, it is building an entirely randomized tree whose structures are independent of the target variable values of the learning sample[16]. By being able to specify specific parameters, this is what lead to its performance. The Linear regression, SVM Linear, and Ridge models had remarkably similar performance with an RMSE around 1.14. SVM Polynomial and the Lasso model had an RMSE of around 1.13, which is not significantly better than the previous models mentioned.

Future

In terms of ways I could continue to improve upon this program in the future, exploring more types of random forest models could be the best way to further understand the model as it had the best performance. I also believe that exploring other machine learning models such as KNN or Naive Bayes could improve the RMSE. Perhaps analyzing another dataset from within the last five years could also be an exciting endeavor!

References

1. Fan, Xijin Ge, Jianli Qi and Rong. Chapter 9 The Game Sales Dataset | Learn R through Examples. https://gexijin.github.io/learnR/the-game-sales-dataset.html.

2. “R - How to Change Legend Title in Ggplot - Stack Overflow.” https://stackoverflow.com/questions/14622421/how-to-change-legend-title-in-ggplot.

3. “How to Put Labels over Geom_bar for Each Bar in R with Ggplot2 - Intellipaat Community.” https://intellipaat.com/community/16343/how-to-put-labels-over-geombar-for-each-bar-in-r-with-ggplot2.

4. “Color Hex Color Codes.” https://www.color-hex.com/.

5. Datanovia. “Top R Color Palettes to Know for Great Data Visualization” https://www.datanovia.com/en/blog/top-r-color-palettes-to-know-for-great-data-visualization/.

6. “R - Editing Legend (Text) Labels in Ggplot - Stack Overflow.” https://stackoverflow.com/questions/23635662/editing-legend-text-labels-in-ggplot.

7. “Ggplot2 Reference and Examples (Part 2) - Colours.” http://rstudio-pubs-static.s3.amazonaws.com/5312_98fc1aba2d5740dd849a5ab797cc2c8d.html.

8. “Video Games Sales Regression Techniques.” https://kaggle.com/yonatanrabinovich/video-games-sales-regression-techniques.

9. “Sales of Video Games (Analysis & Visualization).” https://www.kaggle.com/tnyont/sales-of-video-games-analysis-visualization

10. “Analysis of Videogame sales.” https://www.kaggle.com/rohitbokade94/analysis-of-videogame-sales

11. “Exploring the Pros and Cons of Video Gaming.” https://online.concordia.edu/computer-science/pros-and-cons-of-video-gaming/.

12. Montgomery, Douglas C., Elizabeth A. Peck, and G. Geoffrey Vining. Introduction to Linear Regression Analysis. John Wiley & Sons, 2021. https://books.google.com/books?hl=en&lr=&id=tCIgEAAAQBAJ&oi=fnd&pg=PP13&dq=linear+regression&ots=lfufWxg0Jt&sig=sXmg2mZwoECjeYwSgguIW192PQc#v=onepage&q=linear%20regression&f=false

13. Corporate Finance Institute. “Heteroskedasticity.” https://corporatefinanceinstitute.com/resources/knowledge/other/heteroskedasticity/.

14. Gandhi, Rohith. “Support Vector Machine — Introduction to Machine Learning Algorithms.” Medium, July 5, 2018. https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

15. Analytics Vidhya. “Random Forest | Introduction to Random Forest Algorithm,” June 17, 2021. https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/.

16. Geurts, Pierre, Damien Ernst, and Louis Wehenkel. “Extremely Randomized Trees.” Machine Learning 63, no. 1 (April 2006): 3–42. https://doi.org/10.1007/s10994-006-6226-1.

17. Linares, Kevin. “Linear Regression Equation in LaTex Using TexMaths under LibreOffice:” Kevin A. Linares (blog), September 17, 2015. https://linareskevin.wordpress.com/2015/09/17/linear-regression-equation-in-latex-using-texmaths-under-libreoffice/.

18. LaTex: Embedding Maths Equations | Data Science and Machine Learning.” https://www.kaggle.com/getting-started/a.

19. Moody, James. “What Does RMSE Really Mean?” Medium, September 6, 2019. https://towardsdatascience.com/what-does-rmse-really-mean-806b65f2e48e.