Video Game Recommendation System

HarvardX Data Science PH125.9x Capstone - Video Game

Erik Martins Tonon

September 2022

Contents

Introduction

Video games have been around for some years now and it is remarkable how it has changed over the years. From hungry-yellow dots running away from ghosts to Virtual Reality headsets where you are fully immersed in another world. In the early times, people used to wait in line at Brookehaven National Laboratory to play Tennis for Two, an electronic tennis game that is unquestionably a forerunner of the modern video game era. (Brookhaven (n.d.))

In 2020, the revenue from the worldwide PC Gaming market was estimated at almost 37 U.S. billion dollars (Clement (2021)) and it is known that video games are also being introduced to different sectors through *'Gamefication'*, which by definition is simply applying game mechanics to increase user engagement.

In this study, we are going to explore the different point of views of Video Game sales using Machine Learning algorithms. The goal of this project is to predict video game sales based on the dataset selected.

Starting with exploratory analysis, the data set variables will be analyzed and with the insights gained, the appropriate machine learning method will be applied.

Exploratory Analysis

The Video Game dataset has 16.598 observations with 11.493 unique video games and 11 variables such as Name of the game, Platform it was released, Year, Genre, Publisher and the Unit Sales in millions.

As seen in the table 1 bellow, *'Wii Sports'* has sold around 41.49 million units in North America whereas *'Super Mario Bros'* sold approximately the same amount of units but globally.

Table 1: First 10 rows of dataset.

Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
1	Wii Sports	Wii	2006	Sports	Nintendo	41.49	29.02	3.77	8.46	82.74
2	Super Mario Bros.	NES	1985	Platform	Nintendo	29.08	3.58	6.81	0.77	40.24
3	Mario Kart Wii	Wii	2008	Racing	Nintendo	15.85	12.88	3.79	3.31	35.82
4	Wii Sports Resort	Wii	2009	Sports	Nintendo	15.75	11.01	3.28	2.96	33.00
5	Pokemon Red/Pokemon Blue	GB	1996	Role-Playing	Nintendo	11.27	8.89	10.22	1.00	31.37
6	Tetris	GB	1989	Puzzle	Nintendo	23.20	2.26	4.22	0.58	30.26
7	New Super Mario Bros.	DS	2006	Platform	Nintendo	11.38	9.23	6.50	2.90	30.01
8	Wii Play	Wii	2006	Misc	Nintendo	14.03	9.20	2.93	2.85	29.02
9	New Super Mario Bros. Wii	Wii	2009	Platform	Nintendo	14.59	7.06	4.70	2.26	28.62
10	Duck Hunt	NES	1984	Shooter	Nintendo	26.93	0.63	0.28	0.47	28.31

On the other hand, table 2 shows the bottom 10 video games, and titles like *'Woody Woodpecker in Crazy Castle 5'* and *'Spirits & Spells'* have the lowest Global Sales.

Table 2: Bottom 10 rows of dataset.

Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
16591	Mega Brain Boost	DS	2008	Puzzle	Majesco Entertainment	0.01	0.00	0.00	0	0.01
16592	Chou Ezaru wa Akai Hana: Koi wa Tsuki ni Shirube Kareru	PSV	2016	Action	dramatic create	0.00	0.00	0.01	0	0.01
16593	Eiyuu Densetsu: Sora no Kiseki Material Collection Portable	PSP	2007	Role-Playing	Falcom Corporation	0.00	0.00	0.01	0	0.01
16594	Myst IV: Revelation	PC	2004	Adventure	Ubisoft	0.01	0.00	0.00	0	0.01
16595	Plushees	DS	2008	Simulation	Destineer	0.01	0.00	0.00	0	0.01
16596	Woody Woodpecker in Crazy Castle 5	GBA	2002	Platform	Kemco	0.01	0.00	0.00	0	0.01
16597	Men in Black II: Alien Escape	GC	2003	Shooter	Infogrames	0.01	0.00	0.00	0	0.01
16598	SCORE International Baja 1000: The Official Game	PS2	2008	Racing	Activision	0.00	0.00	0.00	0	0.01
16599	Know How 2	DS	2010	Puzzle	7G//AMES	0.00	0.01	0.00	0	0.01
16600	Spirits & Spells	GBA	2003	Platform	Wanadoo	0.01	0.00	0.00	0	0.01

Name

If we take a closer look to the variable *'$name'*, it is possible to see that the global unit sales in million per video game is still pretty high and consist mostly of titles from the publisher Nintendo, such as *'Mario Kart Wii'* and *'Duck Hunt'*.

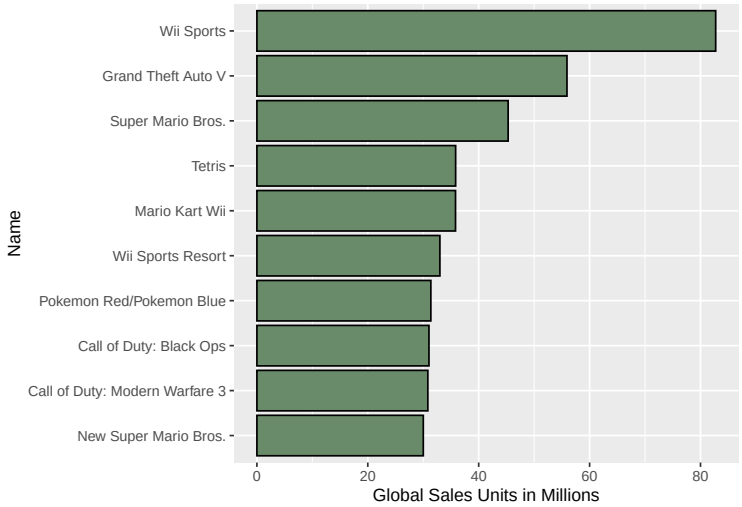

Figure 1: Top 10 Video Games

Platform

The variable *'$Platform'* shows us the platform responsible for launching these games, for example, *'Wii'* and *'PS4'* among 31 unique platforms available in the dataset.

It is possible to analyze that, even though the average unit sales only presented *'Nintendo'* games at Figure 1, *'PS2'* and *'X360'* games are leading the sales with over 1200 Million unit sold as shown in Figure 2.

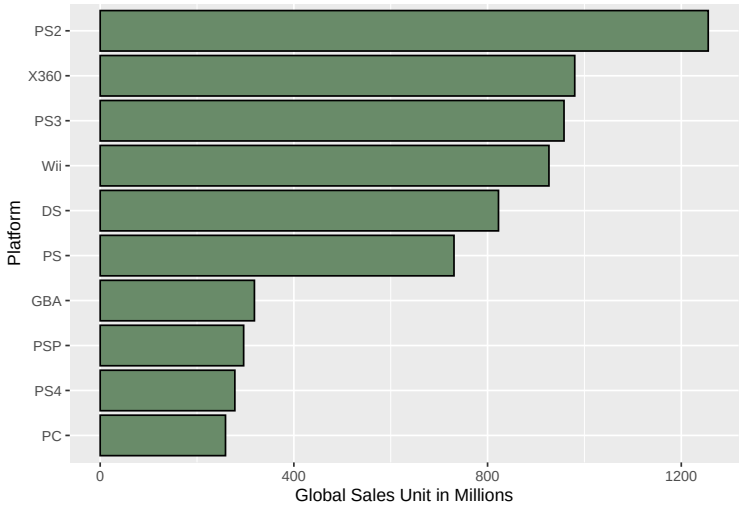

Figure 2: Top 10 Global Sales per Platform

Year

The variable *'$Year'* contains the year the video game was released, from 1980 to 2020 releases, and bellow at Figure 3, we see that most games were released between 2000 and 2015 being 2008 the release year with the highest unit sales (Figure 4).

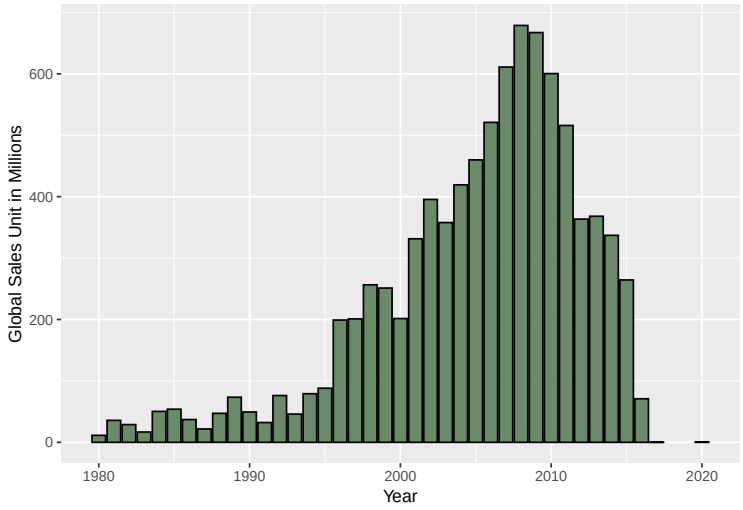

Figure 3: Global Sales Unit per Release Year

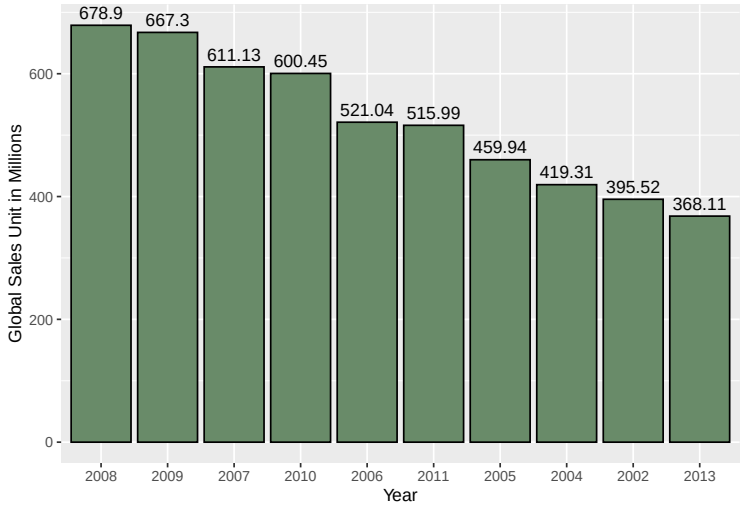

Figure 4: Top 10 Release Years with most Global Sales Unit

Genre

The variable *'$Genre'* contains 12 unique genres of the game, and as seen in Figure 5 below, in average, *'Platform'* games have the most units sold.

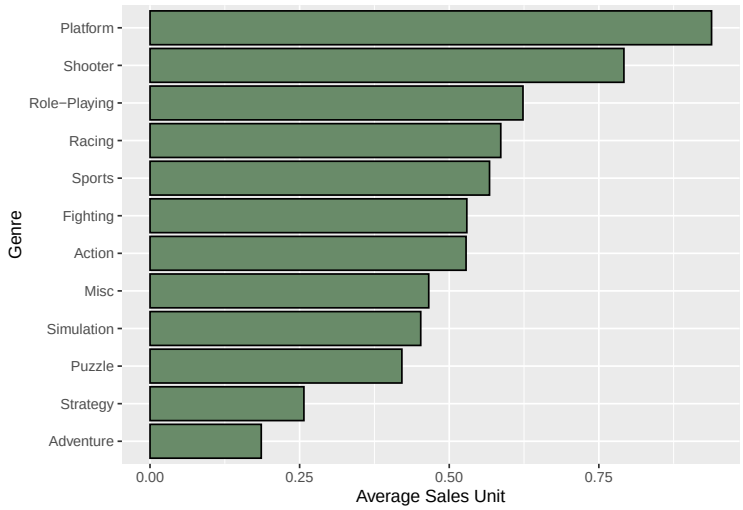

Figure 5: Average Global Sales Unit per Genre

While in average, *'Platform'* games have the most units sold, in total, *'Action'* games have more than 100 million units sold (Figure 6).

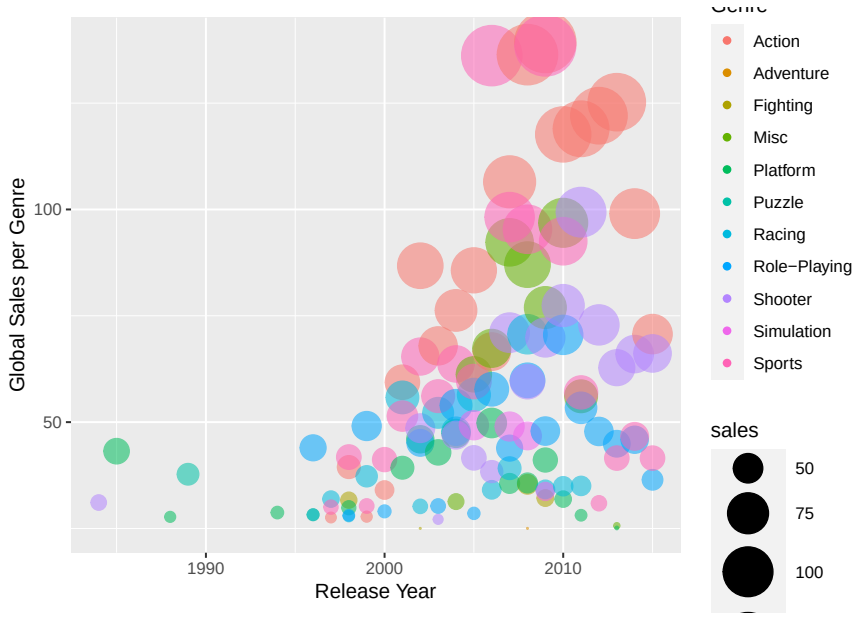

Figure 6: Global Sales Unit per Genre

This demonstrates variance between genres and can potentially improve our algorithm.

Publisher

The variable *'$Publisher'* contains 579 unique publishers and as presumed before, *'Nintendo'* has the most sold units followed by *'Electronic Arts'*.

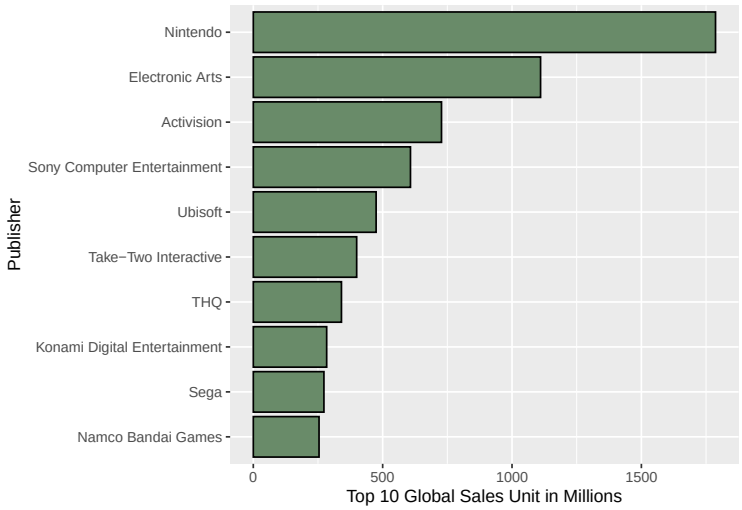

Figure 7: Global Sales Unit per Publisher

Sales Regions

The variable *'$Sales Regions'* is split into 5 variables, 4 regions and 1 total, and analyzing each region individually we see that *'North America'* has led most of the sales around 2010, followed by *'Europe'*.

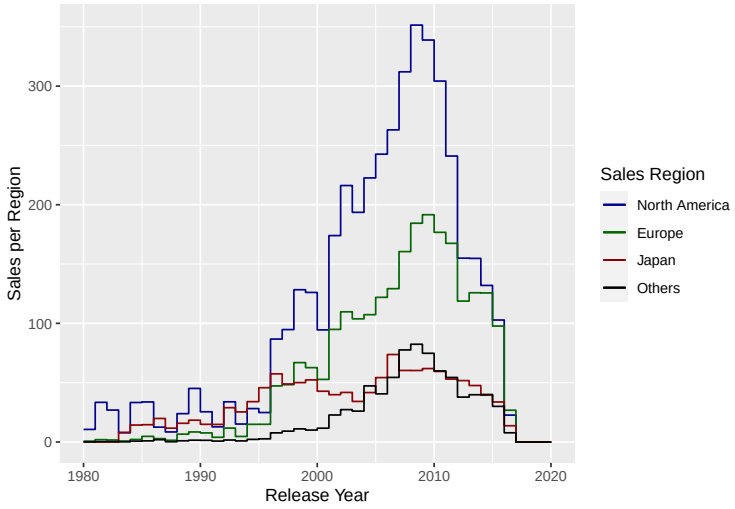

Figure 8: Sales per Sales Region

Methods

The interpretation for RMSE is the typical error we make when predicting. If this number is larger than 1, it means our typical error is way off the reality, which is not good. (Irizarry (2020))

Naive Model

The naive model predicts that the video games will have the same sales regardless of Name, Platform, Year, Genre, Publisher or Rank, and the estimate that minimizes the RMSE is the average of all sales.

Method	RMSE
Simple Average	0.78722

Linear Regression

Linear Regression estimates the relationship between independent variables and it is mostly used for finding out the relationship between the variables and the forecast (Gupta (n.d.)).

Initially, let's fit the linear model for the 'Platform + Year + Genre + Publisher' and check the results.

```
# Fiting the model
set.seed(1)
fit <- lm(y_train ~ Platform + Year + Genre + Publisher, data = x_train)

# Predicting the values based on the test_set.
y_hat <- predict(fit, x_test)

# Predict on the test set, round and format.
pred_rmse <- format(round(RMSE(y_test, y_hat),5), nsmall = 5)

# Adding the results to rmse_results.
rmse_results <- rmse_results %>%
  rbind(c("Linear Regression", pred_rmse))
```

And as seen below, the RMSE is slightly better than the simple average, which means our model lacks variability and the algorithm needs improvements.

Method	RMSE
Simple Average	0.78722
Linear Regression	0.75785

Fitting the algorithm using all the features available has reduced drastically the RMSE.

Method	RMSE
Simple Average	0.78722
Linear Regression	0.75785
Linear Regression (All)	0.00523

Random Forest

Random Forest is learning method that can be applied to various prediction tasks and can be used for regression and classification models. The essence of the model is a combination of decision trees.

```
# Fitting the random Forest
set.seed(1)
fit <- randomForest(x = x_train, y = y_train , maxnodes = 10, ntree = 10)

# Make prediction
predictions <- predict(fit, x_test)

# Predict on the test set, round and format.
pred_rmse <- format(round(RMSE(y_test, predictions),5), nsmall = 5)

# Adding the results to rmse_results.
rmse_results <- rmse_results %>%
  rbind(c("Random Forest Regression", pred_rmse))
```

The current model has returned a *'% Var explained'* of *'80.08%'*, but we can do better.

```
##
## Call:
##  randomForest(x = x_train, y = y_train, ntree = 10, maxnodes = 10)
##                Type of random forest: regression
##                      Number of trees: 10
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 0.1234482
##                     % Var explained: 80.08
```

The scatter plot below shows that our prediction is really close to reality.

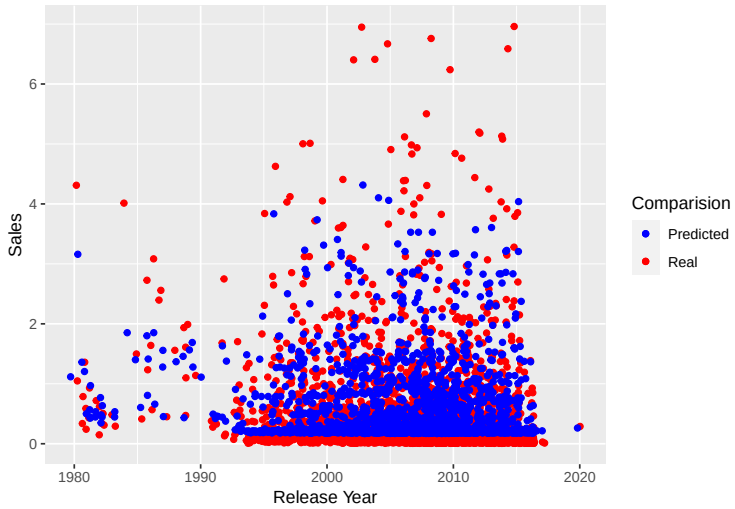

Figure 9: Prediction vs Real

And the accuracy of the Random Forest is *'0.30825'*:

Method	RMSE
Simple Average	0.78722
Linear Regression	0.75785
Linear Regression (All)	0.00523
Random Forest Regression	0.30825

As we have analyzed before in Figure 8, *'NA_Sales'* is the most important predictor from the train set.

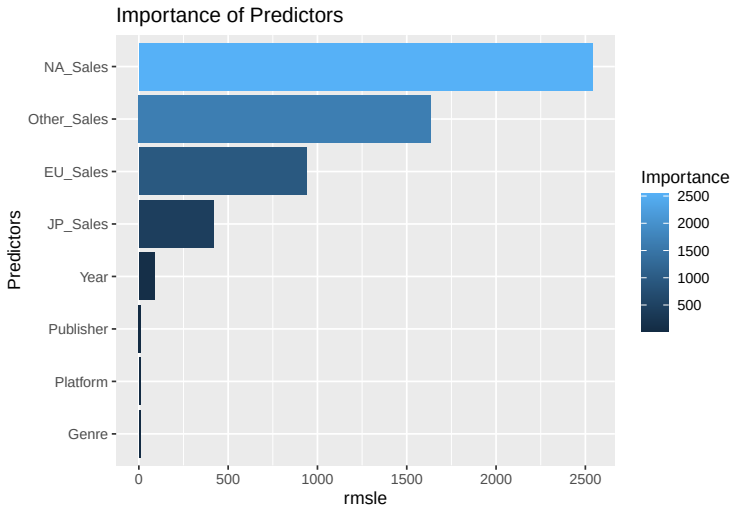

Figure 10: Importance of Predictors

Random Forest Tunning

Currently, our first Random Forest model is using 10 trees and a maximum number of terminal nodes of 10 too. After some exploration and testing, the best parameters for this set are 140 nodes and 1100 trees.

```
# Fitting the random Forest
set.seed(1)
fit <- randomForest(x = x_train, y = y_train , maxnodes = 140, ntree = 1100)

# Make prediction
predictions <- predict(fit, x_test)

# Predict on the test set, round and format.
pred_rmse <- format(round(RMSE(y_test, predictions),5), nsmall = 5)

# Adding the results to rmse_results.
rmse_results <- rmse_results %>%
  rbind(c("Random Forest Tunning", pred_rmse))
```

The current model has returned a *'% Var explained'* of *'96.87%'*, which is pretty good.

```
##
## Call:
##  randomForest(x = x_train, y = y_train, ntree = 1100, maxnodes = 140)
##                Type of random forest: regression
##                      Number of trees: 1100
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 0.01937237
##                     % Var explained: 96.87
```

And now, the scatter plot below shows better results in comparison to the previous version.

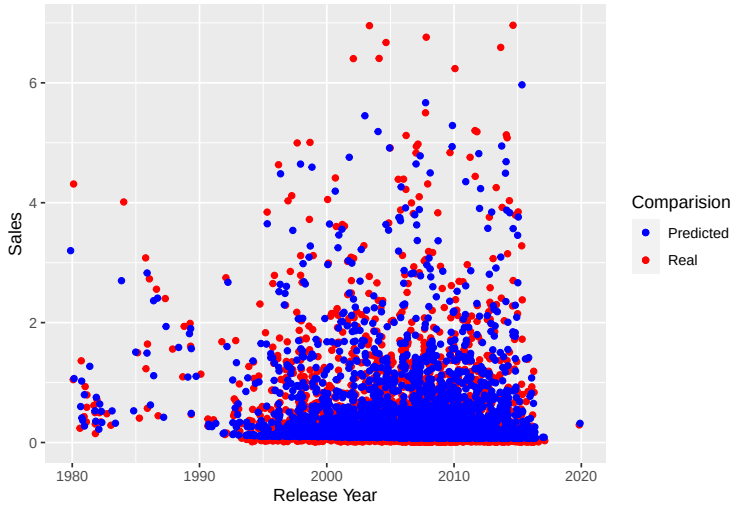

Figure 11: Prediction vs Real

As seen below, after around 200 trees, the difference in errors gets smaller between the trees.

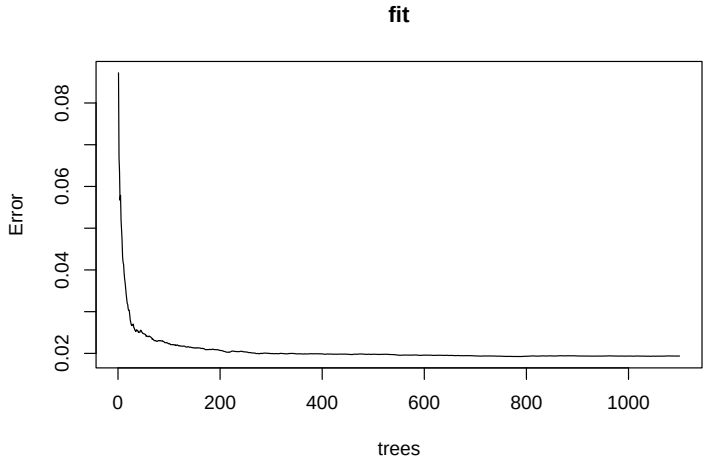

Figure 12: Errors per tree

And the accuracy of the Random Forest with tuning is *'0.13246'*:

Method	RMSE
Simple Average	0.78722
Linear Regression	0.75785
Linear Regression (All)	0.00523
Random Forest Regression	0.30825
Random Forest Tunning	0.13246

Results

The Linear Regression model has achieved the best results since the data set is pretty small, while the Random Forest model did a great job and presented similar outcome using 140 nodes and 1100 trees.

Both models were able to predict the quantity of sales according to the test set with an acceptable RMSE.

Conclusion

Given that the data set is pretty small and there are not many features, both models were acceptable but Linear Regression has given the best results for this study. We have seen the data, explored its features, identified some potential candidates for the algorithm and applied them.

The naive model provided us with the base for understanding the goal, the Linear Regression has proven the best model, and Random Forests has shown great results as well.

Having further information on regions and user profile, could improve this analysis.

References

Brookhaven. n.d. *The First Video Game?* Brookhaven National Laboratory. https://www.bnl.gov/about/history/firstvideo.php.

Clement, J. 2021. *Video Game Industry - Statistics & Facts.* Statista.

Gupta, Mohit. n.d. *ML | Linear Regression.* Geeks For Geeks. https://www.geeksforgeeks.org/ml-linear-regression/.

Irizarry, Rafael A. 2020. *Introduction to Data Science: Data Analysis and Prediction Algorithms with r.* CRC Press.