Are Video Game Review Scores Associated with Their Sales Performance: A Multi-Method Approach with Two Stacking Regressors

Mariana Botelho Pereira
Federal Fluminense University
Av. Gal. Milton Tavares de Souza, s/nº
Niterói, RJ, 24210-346

Abstract

This work aims to analyse the performance of different machine learning methods in a dataset about video game sales. Four estimators were chosen to be used in two stacking regressors with different final estimators. Since not all features have a straight connection, e.g. not all games that have a high critics score sold a lot of copies, the estimators performed poorly, showing that this assumption is somewhat correct. Further analysis is needed with a wider variety of games and other estimators, as well as other ensemble methods.

In the world of video games, professional game reviews are seen as a skewed metric. Many times they evaluate high selling games with lower scores than expected. This project aims to try to predict critics scores based on game sales and other features present in the dataset, which will be discussed hereinafter.
To manipulate the dataset and analyse the data, the pandas and Scikit-learn libraries were used. These were chosen for the ease of use and vast documentation found online.

The Dataset and the Data

For this analysis, the 'Video Game Sales with Ratings' dataset was chosen from Kaggle. This particular dataset is an extension to the 'Video Game Sales' dataset, with extra features featuring ratings. The dataset consist of 16 features and 16717 instances, presenting categorical, textual and numerical data. The features are the following:

- Name: The game's name;
- Platform: Platform of the game's release;
- Year_of_Release: Year of the game's release (until 2016);
- Genre: Genre of the game;
- Publisher: Publisher of the game;
- NA_Sales: Sales in North America (in millions);
- EU_Sales: Sales in Europe (in millions);
- JP_Sales: Sales in Japan (in millions);
- Other_Sales: Sales in the rest of the world (in millions);
- Global_Sales: Total worldwide sales (in millions);
- Critic_Score: Aggregate score compiled by Metacritic's staff;
- Critic_Count: The number of critics used in coming up with the Critic_Score;
- User_Score: Score by Metacritic's subscribers;
- User_Count: Number of users who gave the User_Score;
- Developer: Company responsible for creating the game;
- Rating: The ESRB rating.

The dataset presents a handful of interesting data as well as some interesting insights into the game industry. One of the ideas behind using this dataset is to show that not always the specialised critics get the rating right, e.g. Wii Sports sold 82.53 million copies worldwide and received a score of 76 by the critics, while Mario Kart Wii with a score of 82 sold only 35.52 million copies, less than half of the other game but with a higher score.
One of the biggest issues of the dataset is that a lot of games don't have a Critic_Score, which includes the majority of games released before 2001, the year that the Metacritic's website was launched. Due to this and the lack of a few other information, the scope went from 16717 instances to 6825 instances, which is still a lot of data.

Method

Cleaning and Pre-Processing the Data

As stated before, a lot of instances lacked critical information (Critic_Score) while others lacked other information (e.g. Developer and Rating). Those instances were eliminated, as well as the feature which presented the name of the games.
The features with categorical data (Platform, Genre, Publisher, Developer and Rating) were pre-processed using the Ordinal Encoder, where they became categorical values between 0 and $n - 1$, where n is the amount of unique values in the feature (each feature has its own n).
Initially, the idea was to pre-process the numerical columns inside the pipelines using the One Hot Encoder, but due to its nature and the dataset, some data was lost during the encoding process, specifically all the values in the Year_of_Release feature. Due to that, the idea was scraped

and only the manual pre-processing and the Ordinal Encoder were used. This also reduced the scope of the project, turning it into something more manageable.

Estimators

To evaluate the data, four algorithms were chosen (Linear Regression, Lasso, ElasticNet and Decision Tree Regressor) which were used in two instances of a Stacking Regressor, with two final estimators (the standard, Ridge CV, and Random Forest Regressor). This amounts to 10 estimations, but as we'll see forward, only 7 were used.
For sheer simplicity, we decided to keep the standard parameters in all the estimators.

Linear Regression Linear Regression is one of the most basic regression methods. It tries to model a relationship between two variables, fitting a linear equation in the observed data (Barron 1997).

Lasso Lasso is a method similar to the Linear Regression, but this one uses shrinkage, which is a process that shrink the data towards a central point, like the mean (Pedregosa et al. 2011). It also uses L1 regularisation, meaning it weights errors at their absolute value (Ye 2020).

ElasticNet ElasticNet is a linear regression method which uses combined L1 and L2 priors as regulariser (Pedregosa et al. 2011).

Decision Tree Regressor Decision Tree is a method that uses a decision tree to go from observations about an item to a conclusion (Pedregosa et al. 2011).

Ridge CV RidgeCV is a method similar to the Lasso, but uses L2 regularisation, which weights errors at their square (Ye 2020).

Random Forest Regressor Random Forest Regressor is an ensemble method that uses a 'forest' of decision trees and calculate the average of the results, giving a more accurate result (Pedregosa et al. 2011).

Stacking Classifier This ensemble method 'stack' the results of all the other methods and uses a final estimator to calculate the final prediction. In this case, was used two 'stacks', one with the standard final estimator, RidgeCV, and one with the Random Forest Regressor (Pedregosa et al. 2011).

Cross Validation

To get a better grasp on the results and a more homogeneous training set, a cross validation method was used, more specifically the *cross_validate* and *cross_val_predict*.
The *cross_validate* method returns the chosen scores (discussed below) as well as the time it took to fit the data.
The *cross_val_predict* method returns the values that the method predicted.

Results

To evaluate the methods, the following metrics were chosen: R2, Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Max Error. On a first analysis, the values presented in the following tables were found.

Estimators	R2	RMSE
First Stacking		
Linear Regr.	-0.809725	13.748134
Lasso	0.335111	10.278894
ElasticNet	0.317713	10.416875
Decision Tree	-0.072956	13.120767
Ridge CV	-0.121943	11.996236
Second Stacking		
Linear Regr.	-0.809725	13.748134
Lasso	0.335111	10.278894
ElasticNet	0.317713	10.416875
Decision Tree	-0.089699	13.242441
Random Forests	0.356857	10.185403

Table 1: R2 and RMSE Values

Estimators	MAE	Max Error
First Stacking		
Linear Regr.	9.196593	140.836709
Lasso	8.094631	39.441593
ElasticNet	8.230083	38.736773
Decision Tree	10.038388	53.400000
Ridge CV	8.457285	107.253460
Second Stacking		
Linear Regr.	9.196593	140.836709
Lasso	8.094631	39.441593
ElasticNet	8.230083	38.736773
Decision Tree	10.149744	55.000000
Random Forests	7.921508	41.406000

Table 2: MAE and Max Error Values

As we can see, the values for the Linear Regression, Lasso and ElasticNet were the same across both stacking methods, which was a surprise. Since we used a cross validation method, we expected a slight variation between evaluations. To reduce the scope of the project, only the first results for these methods were used.
A slight variation was present on the Decision Tree method, which, following the similarities across the other methods, was also a surprise.
Thus, for ease of visualisation, the scores are presented in the following graphs, and to gain a better understanding of the predictions vs. the real results, a few scatter plots were plotted.

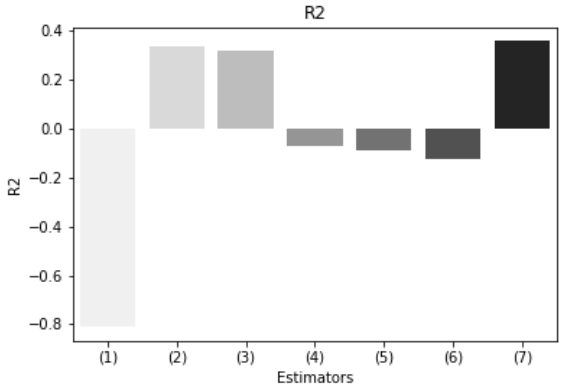

Figure 1: R2 Scores, where: (1) Linear Regression; (2) Lasso; (3) ElasticNet; (4) Decision Tree (RidgeCV); (5) Decision Tree (Forests); (6) RidgeCV; (7) Random Forests.

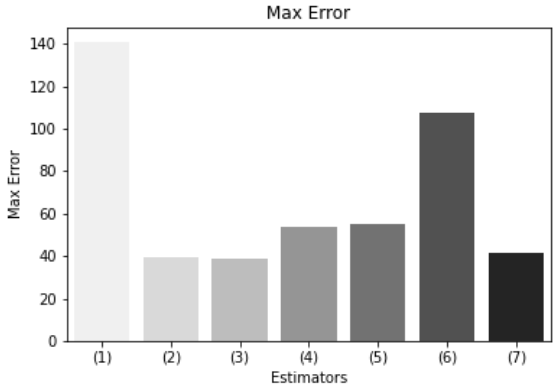

Figure 4: Max Error, where: (1) Linear Regression; (2) Lasso; (3) ElasticNet; (4) Decision Tree (RidgeCV); (5) Decision Tree (Forests); (6) RidgeCV; (7) Random Forests.

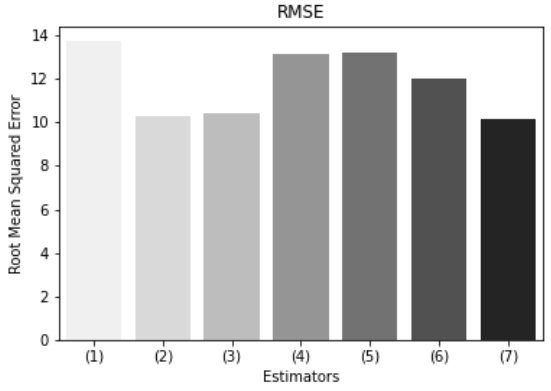

Figure 2: RMSE Scores, where: (1) Linear Regression; (2) Lasso; (3) ElasticNet; (4) Decision Tree (RidgeCV); (5) Decision Tree (Forests); (6) RidgeCV; (7) Random Forests.

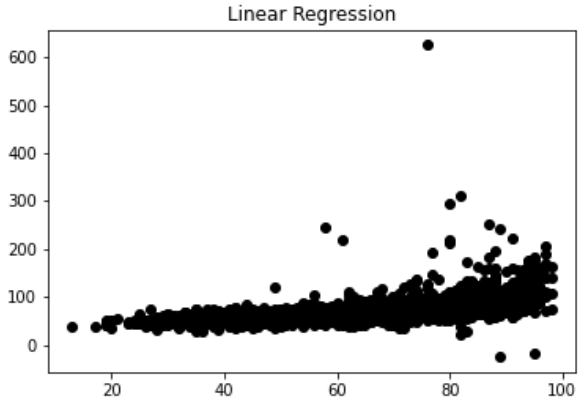

Figure 5: Linear Regression Scatter Plot

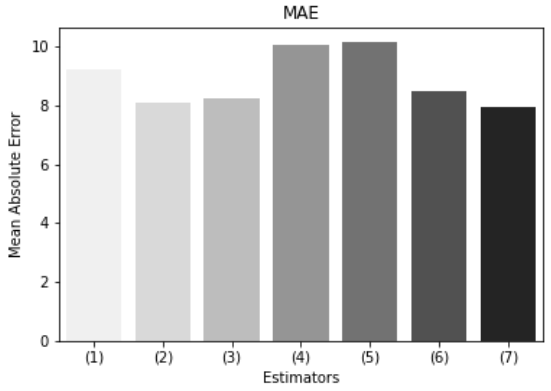

Figure 3: MAE Scores, where: (1) Linear Regression; (2) Lasso; (3) ElasticNet; (4) Decision Tree (RidgeCV); (5) Decision Tree (Forests); (6) RidgeCV; (7) Random Forests.

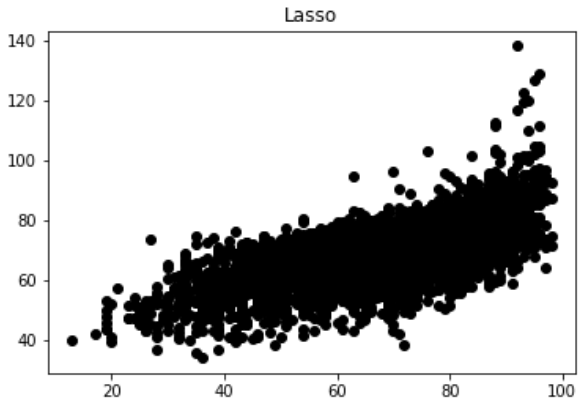

Figure 6: Lasso Scatter Plot

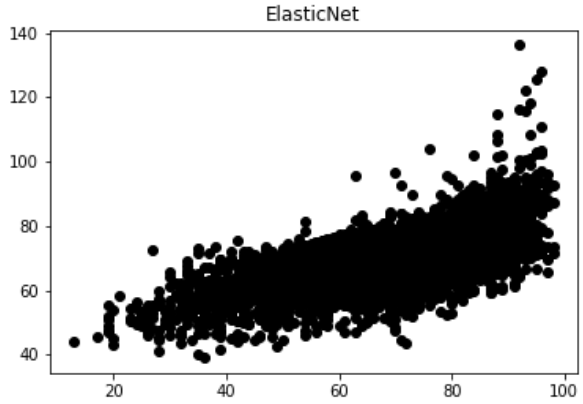

Figure 7: ElasticNet Scatter Plot

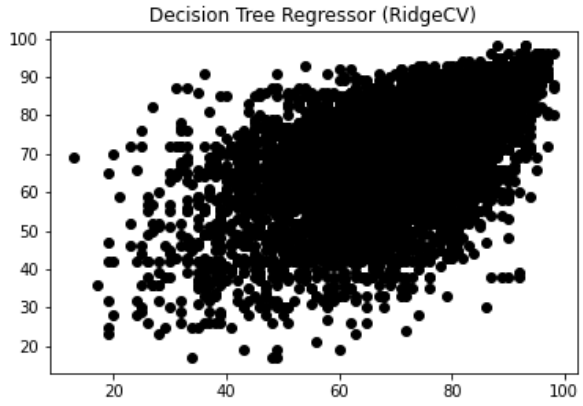

Figure 8: Scatter Plot of the Decision Tree from the First Stacking Method

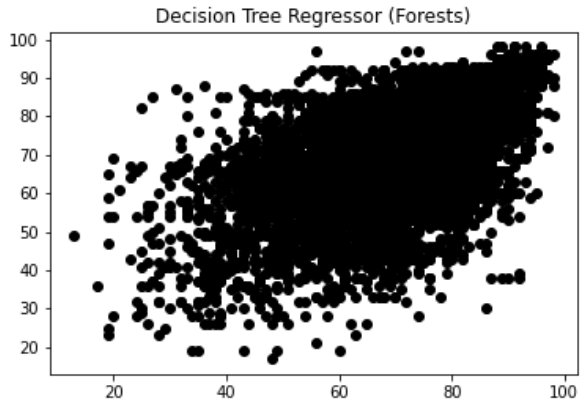

Figure 9: Scatter Plot of the Decision Tree from the Second Stacking Method

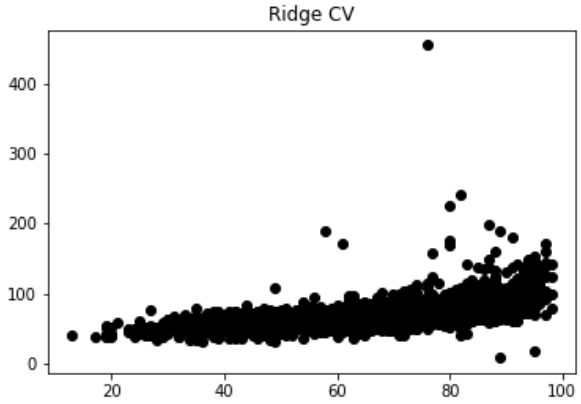

Figure 10: Ridge CV Scatter Plot

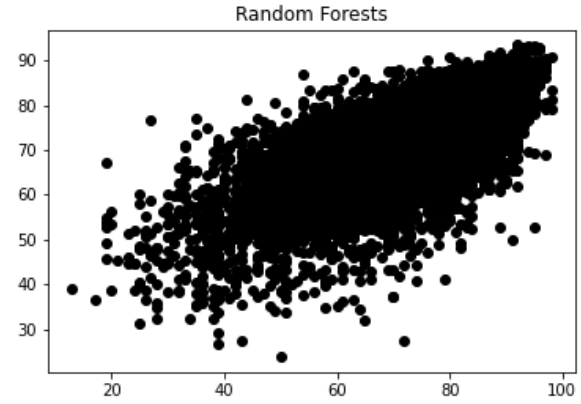

Figure 11: Random Forests Scatter Plot

Looking at the information provided, we can see that the Linear Regression method performed poorly overall and with an outlier (seen on the scatter plot), which was also seen on the Ridge CV method.
Both the Lasso and ElasticNet methods had the second and third lowest mean absolute error, lowest root mean squared error, smallest max error and highest R2 values, losing the first place to the second stacking method, using Random Forests.

Conclusion

Compared to all the methods used, the Lasso, ElasticNet and the second stacking method had somewhat satisfactory results, with the lowest errors but still high. Reviewing the R2 metric, where an estimator with a metric score as close as possible to 1 is considered good, we can see that neither method was able to attain such value - the closest of the three being the second stacking method with a score of 0.356857.

Therefore, with such high errors in general, one can conclude that the score given by critics does not have a direct relation with the sales of a game, though further evaluations should be made, specially using other parameters on the methods.

References

Barron, A. 1997. Linear regression.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Ye, A. 2020. 5 machine learning regression algorithms you need to know.