Sales Forecasting and the Utility of Predicting Video Game Sales using Machine Learning

Group 19: Michael Jung, Dani Roytburg

December 13th, 2023

1 Introduction

In the dynamic landscape of the gaming industry, the challenge of accurately forecasting video game sales is a formidable task, driven by the complex interactions among several factors. The nuances of predicting success in this field are exemplified by the numerous variables at play, ranging from the year of release and the choice of the publisher to the genre and platform. This complexity introduces a formidable hurdle for forecasters, demanding a sophisticated comprehension of the multifaceted nature of video game sales.

The importance of accurate prediction of video game sales cannot be overstated. It directly impacts the success and profitability of video game companies, allowing them to allocate resources efficiently, plan marketing campaigns, and make informed decisions about game development. Investors and stakeholders in the gaming industry rely on these forecasts to assess the potential return on investment, and understanding the factors that contribute to a game's success is vital for sustaining a competitive edge in the market.

This exploration will attempt to take advantage of machine learning techniques to provide sophisticated, optimal forecasts for video game producers looking to take new titles to market. The models provided will take advantage of the categorical nature of many of the characteristics of video games in order to produce viable estimates for sales success. Our main objective is to develop a predictive model capable of anticipating video game sales by taking into account key input factors such as ranking, category, platform, primary audience, and year of publishing. The goal of this project is to further our understanding of the complex dynamics that exist within the gaming industry. Our project expresses a novel approach to forecasting by integrating a variety of methods and ensembling them to produce final forecasts, as well as investigating the utility of analyzing the title of games as a predictive factor in forecasting. We hope to contribute to the existing literature on video game sales forecasting by implementing a suite of methods to determine optimal forecast models.

2 Background

In order to approach the problem of machine learning applications to video game sales forecasting, we looked at a few pieces of precedent work. Some of this work addresses head-on the problem space that we are looking at, while other publications focus more on the tangential features that could be added to anticipate video game sales impact.

The most direct application of our work comes from a team of researchers from Ganfang College in Guangzhou, China (Li et al. 2021). The investigators in this work sought to employ adaptive models to optimize potential forecasting using the same source as our dataset, which is the sales forecasting data available on the VgChartz website. While their set of features encompasses ours, they also scraped additional data relating to weekly sales through several definitions (sales on platform, hard copies v. software, etc.) and the sales ranking. To select from this diverse set of features, they used a hybrid between Pearson Correlation and feature selection based on variable importance from Random Forests to trim down their model. Of the models that they employed and the features selected with this technique, they discovered that the highest-performing model was an Adaboost model with an R2 value of 58.5%.

Relatedly, a team of researchers from Andhra University in India broached the same dataset with a variety of different approaches. They employed linear regression, support vector regression, decision trees and random forests in order to forecast video game sales. Their problem space and data was very similar to ours. Measured through root means squared error, this team achieved an optimal RMSE error of 1.4648, where the unit is millions of copies sold (Keerthana and Rao 2019). Finally, a team of researchers from Ireland investigated a different problem space. They extracted social media information about the level of outreach and sentiment in social media posts that relate to a video game, and used this information in order to determine an effect on sales (Malvankar et al. 2023). Ultimately, this team is developing a different set of solutions that are oriented towards evaluating the impact of social media on video game sales forecasting. However, it gave us important ideas regarding extracting information on video games and measuring impact through alternative scales. This also requires video game data that filters for a time where social media becomes a prevalent means of promoting products. Many of our titles come from before the invention of social media, and as such would not be able to directly incorporate this analysis. The team gathered observations of data (i.e. video games) on social media posts, rather than through a dataset on sales performance like we have.

3 Methodology

We employed a variety of techniques in order to capture the decision boundary for our estimates. For our purposes, we will need to use regression in order to estimate the final sales output of our models. From here, we can divide these different regression methods into categories, namely tree-based, linear-based and neural-network based models.

First, we will cover tree-based regressors. The first of these models is the decision tree regressor. Decision trees learn orthogonal decision boundaries by capturing thresholds on features that optimize a certain configuration of parameters by some metric (named entropy and gini impurity). Considering each feature and threshold therein, the model selects the optimal feature and split point to use to divide the data points. Then, it operates recursively, selecting the next best feature and threshold for the new subset of data and excluding the feature that it just split upon. This process continues until the tree has reached its maximum depth or a minimum number of samples post-split, at which point it will provide an estimate based on the average of the data points that it has partitioned. This creates a "tree" of different decision boundaries with a variety of different outcomes. This model is very helpful to us because it can take advantage of the boolean values that are created through our one-hot encodings that will invariably result from our categorical variables. Relative to methods that attempt to encode these 0/1 values with coefficients like linear models, a tree-based approach can handle binary data better and provide better average estimates.

We can extrapolate to more sophisticated implementations of the decision tree learning structure. We use two models that ensemble decision trees to increase their performance. The first is called a random forest. Random forests are, as the name suggests, collections of decision trees that attempt to learn the data using different combinations of the same feature. At each split in a random forest's tree, only a random subset of the features is used to make inferences. This allows for the different trees to produce different structures; abiding by the logic above, without such a randomization step the trees will inevitably replicate the same structure as one another and not learn unique configurations of the data. The final result is a composite average across each of the regression estimates of the trees. This sort of learning extends upon the initial benefits that we might employ using a decision tree because the randomization takes advantage of sparse data by detecting different optimal splits throughout the course of its estimates, thus avoiding the pitfalls of defaults based on splits which would otherwise be made easier. Providing this challenge to the model of working with limited features is an asset in making the model more robust. Aside from random forests, we employ gradient boosted trees. Gradient boosted trees ensemble decision trees not through random selection of features but rather through making new ones. The model begins with a poor-performing tree that intentionally has bad estimates. The model then calculates its initial error relative to the training set to produce a column called the residual, which is added as a new feature for the next learner. This process continues, with the subsequent residual values getting smaller and smaller and providing more information to the tree about areas where it is performing worse. This "boosting" process is helpful for our data because of its unpredictable nature. Since there are a wide variety of performers even when categorized by genre and platform, boosting allows for the model to learn the areas where its decisions are incomplete and incorporates a new set of numerical features into the analysis. This optimizes the model by providing a metric which corrects for sparsity.

Our second class of algorithms involves different strategies of capturing linear decision boundaries. We elected to test two models on the performance of our dataset. The first is an Elastic Net model. Elastic Net is based on a typical linear regression model which uses gradient descent in order to find optimal parameters to fit a line through the data. It thus provides different weights for each variable of analysis. However, Elastic net adds two different penalties to the loss function in order to discourage weight domination. These two penalties are L1, or "lasso" and L2, or "ridge." Elastic net uses a ratio to weigh these two penalties and add them to the loss function. We felt that Elastic net would be a helpful intervention for this model as we wanted to generate a clear model that can attempt to map our data through linear decision boundaries. The introduction of regularization parameters is also helpful to avoid model overfitting, which is a preeminent concern with unpredictable and sparse data. This helps through variable selection by shrinkage in the L1 parameter, eliminating variables that could be noise. The second linear decision model is a Bayesian ridge regression. This engages many of the same techniques as linear regression, but it optimizes the regularization penalty by incorporating Bayesian principles and updating the prior distribution based on observed data. Unlike Elastic Net, Bayesian Ridge Regression provides full probability distributions for each parameter, offering a more nuanced understanding of parameter uncertainty. The probabilistic framework is particularly advantageous in our case when dealing with limited data, allowing for a comprehensive assessment of the model's uncertainty and potential benefits in scenarios where interpretability and uncertainty quantification are crucial.

Finally, we use a multi-layer perceptron (MLP) as our neural network-based model. The multi-layer perceptron extends the logic of a single perceptron unit and transforms input data through a series of weighted connections and activation functions to produce an output. MLP uses an input layer, a series of hidden layers, and an output layer to produce estimates based on its nodes. Each node in the network is connected to every node in the next layer, and each connection is associated with a weight that is adjusted during the training process. The information feeds forward, particularly using an activation function to the weighted sum of its inputs which introduces non-linearity. MLP will be useful for our model because it engages in non-linearity, which is a critical component in modeling the decision boundary for our work. Unlike the linear models above, the introduction of activation functions and multiple levels of depth through weights can represent data points more and more as numerical inputs and provide more robust estimates. The introduction of depth through multiple layers also helps model video game data because of its unpredictability.

4 Data and Preprocessing

4.1 Data Description

We began this work by investigating possible outlets for our data. We found a strong hypothetical source for our data through a Kaggle contributor who had scraped the VgChartz.com website. This website is the hegemonic source for video game financial information, and as such is used by other researchers in this context.

The data includes 16,598 observations. Each row is a video game title released on a particular platform. The threshold for inclusion was selling more than 100,000 copies. The columns are the rank of the video game's sales, the title of the game, the platform it released on (there are 31 platforms represented), the genre (12 genres), the publisher (579 publishers), the sales in North America, in Europe, in Japan, and in other regions, as well as the total number of global sales, each in the unit of millions of copies sold. We are seeking to predict the number of global sales.

Looking at this data, we can make an initial observation that sales of video games are closely tied to the year at which they are released, because there is an enormous spike in the number of games sold as the data enters into the 21st century (Figure 1). This is an intuitive observation, considering the incrased availability, proliferation of platforms, and familiarity that increases over time with video games. They are undoubtedly a 21st century phenomenon.

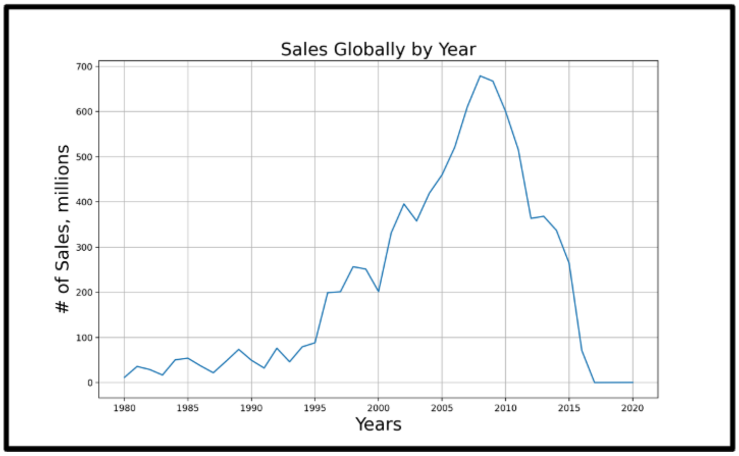

Figure 1: Sales of video games by year.

4.2 Preprocessing

In the preprocessing phase, several key steps were done to refine and prepare the dataset for subsequent analysis. Initially, individual names because the text of names without any feature extraction does not provide useful inferences for our model set. The names will be reintroduced in the dataset in the form of sentence embeddings, described under feature extraction. The rows containing missing values were also dropped, resulting in a dataset of 16,291 entries. There are 579 publishers, so we will downsize this to a top 5 number of publishers and categorize the rest as an 'Other' to encode 6 features. One-hot encoding was done to categorical values such as publishers, genres, and platforms which converts them into a binary format suitable for the machine learning models. Finally, to standardize the representation of regional sales, rather than using their raw values(as their sum would be the y value that we are trying to predict), the values for US, JP, EU, and Other Sales were scaled down proportionally by Global Sales. For instance, if 2 million copies were sold in the U.S. out of 10 million total copies sold, the US Sales category would be 0.2. These preprocessing decisions were made to improve the quality and applicability of the dataset for future modeling. First, the presence of the continuous variables provides a strong baseline upon which a regression model can be constructed. Second, the relative proportion of sales in a certain country is not entirely up to the number of copies sold but can be estimated as a part of a sales strategy on behalf of a global corporation. We used a train/test split of 70/30 for our data.

4.3 Feature Extraction

To create the embeddings of the titles in order to represent them in the dataset, we used the pre-trained HuggingFace transformer model DistilBert. DistilBert is a state-of-the-art transformer mechanism that employs similar techniques to the famed BERT model and achieves 97% of its accuracy, but with half of the required parameters, making the model faster and more efficient. Running this model over the titles, we took the final output layer of embeddings for each sentence, creating a 768-dimensional vector based on learned representations of the words. This turns out to be far too many dimensions for our model, so we used PCA reduction to reduce the dimensionality of this tensor to 100 dimensions. These become 100 embedding columns that are potentially re-introduced into the dataset.

4.4 Feature Selection

We elected to preserve the features that we had created because we wanted to give the model necessary information to make its inferences. We resolved the overproliferation of publisher titles by relegating non- top 5 publishers into the "Other" category. We also reduced the dimensionality of the dataset as described above using PCA on the text embeddings that we generated on the titles.

To demonstrate the use of preserving out features, we created a heatmap of the Pearson Correlation Coefficient. The low values of correlation across the different variables demonstrates that there is not enough colinearity to justify the exclusion of any column values. This heatmap omits the name embeddings created above for readability purposes, and we assume that there would be little colinearity between PCA-reduced BERT-based embeddings and any of the other tabular data (Figure 2).

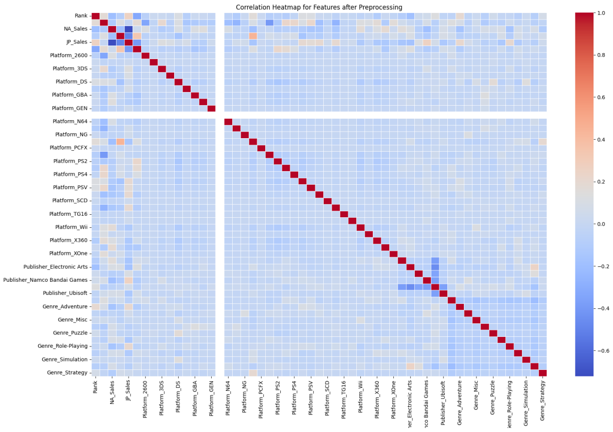

Figure 2: Heatmap of Pearson Correlations across features.

5 Model Choices: Selection, Hyperparameters, Evaluation Metrics

We elected to use each of the models described in our Methodology proportion in an ensemble to evaluate their effectiveness. Upon evaluating the models further, we might elect to remove the ones that perform poorly in order to improve the output of the ensemble model. To ensemble the models together, we trained a linear regression model on the output estimates of each model (with the y value being the actual value). Further model selection decisions will be made based on the results that we get in our training loop.

To evaluate the models, we used a classic suite of metrics for regression analysis. These include mean squared error, mean absolute error, and R^2. These metrics are standard for a reason; each provides a helpful analytic for describing the accuracy of a regression model. Mean squared error uses the following metric to evaluate the accuracy of the model, and it is helpful because its unit of measurement is the same as the unit used to measure y: $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$. Mean absolute error operates in a functionally similar matter, but this metric uses the absolute values as opposed to the squared values of the estimates in order to provide similar conclusions $\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$. Finally, the R^2 value provides a useful percentage component to determine the variance of sales which can be explained by the models that we create, determined as $1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$. This provides a highly intuitive variable percentage for our model explanation.

In order to determine the best parameters for each model, we used scikit-learn's GridSearchCV class, which uses cross-validation across an array of possible hyperparameters in order to find the best combination of hyperparameters for our work. The potential candidates that we selected and the optimal results retrieved are described in Figures 3 and 4. Note that for the tree models, our model yielded an output of 40 for its maximum depth, which would have included all of the possible features. To account for data sparsity, avoid overfitting and force the models to learn stronger representations, we reduced the max depth to 30 as we felt that this allowed the model to still garner a stronger understanding of the dataset without memorizing it too well.

Model	**Hyperparameter**	**Candidates**
BayesianRidge	alpha_1 alpha_2 lambda_1 lambda_2	[1e-6, 1e-5, 1e-4] [1e-6, 1e-5, 1e-4] [1e-6, 1e-5, 1e-4] [1e-6, 1e-5, 1e-4]
ElasticNet	alpha l1_ratio	[0.01, 0.1, 0.5] [0.2, 0.5, 0.8]
MLPRegressor	hidden_layer_sizes alpha learning_rate	[(50,), (50, 50), (50, 50, 50), [0.0001, 0.001, 0.01] [0.1, 0.01, 0.001, 0.0001]
RandomForestRegressor	n_estimators max_depth min_samples_leaf	[50, 100, 200] [10, 20, 30, 40] [1, 2, 4]
GradientBoostingRegressor	n_estimators learning_rate max_depth min_samples_leaf	[50, 100, 200] [0.01, 0.1, 0.2] [10, 20, 30, 40] [1, 2, 4]

Figure 3: Hyperparameters and Candidates for Each Model

Model	**Best Hyperparameters**
Decision Tree	Max depth = 30, Min samples = 2
Random Forest	Max depth = 30, Min samples = 2, # Trees = 100
Gradient Boosted Tree	Max depth = 30, Min samples = 2, Learning Rate = 0.1
Multi-Layer Perceptron	$\alpha = 0.001$, Hidden layers = (100) * 5
Bayesian Ridge	$\alpha_1 = 1 \times 10^{-4}$, $\alpha_2 = 1 \times 10^{-6}$, $\lambda_1 = 1 \times 10^{-6}$, $\lambda_2 = 1 \times 10^{-4}$
Elastic Net	$\alpha = 0.011$, $L1 = 0.2$

Figure 4: Best Hyperparameters for Each Model

6 Performance and Results

When running our model, we found that there were some interesting results. First, we made a counterintuitive observation that there was a slight degradation in model performance when we included the text embeddings. We will use R^2 to illustrate this point (Figure 5). While our linear models enjoy improvements in their performance (which will be discussed shortly), the tree-based and MLP models suffer from slight degradations to their performance. This could be because of the combination of embeddings and PCA reduction reducing the information in the title to noise, or it could be because the titles themselves do not constructively contribute to the quality of a regression estimate. Thus, our final model evaluations actually occur with the exclusion of these values.

Model	R2 with Text (%)	R2 without Text (%)
ElasticNet	18.97	14.67
BayesianRidge	18.91	14.67
GradientBoosting	85.97	87.04
RandomForest	83.42	85.53
MLPRegressor	46.49	51.67
DecisionTree	87.53	87.57

Figure 5: R2 Scores (as Percentages) for Each Model with and without Text

Here are the final performance metrics (Figure 6). We can see that among these models, it is the tree-based models, i.e. Decision Tree, Random Forest, and Gradient Boosted Tree where we see the highest performance. Surprisingly, the best among them is the Decision Tree, followed by the Random Forest and the Gradient Boosted Tree. The final model performance, measured by the ensemble, is at around 87%. These are encouraging results, and suggest that our model is successfully able to predict potential new video game sales. The linear-based models performed fairly poorly, likely due to the difficulty in isolating a decision boundary when we have a wide variety of one-hot encodings. The MLP model was an intermediate performer, demonstrating some faculty in learning the representations of our model but not as good as the tree-based structures.

Model	MSE	MAE	R2 Score (%)
ElasticNet	2.8563	0.5030	14.67
BayesianRidge	2.8562	0.5038	14.67
GradientBoosting	0.4339	0.0126	87.04
RandomForest	0.4843	0.0120	85.53
MLPRegressor	1.6177	0.2145	51.67
DecisionTree	0.4161	0.0132	87.57
Ensemble	0.4364	0.0127	86.96

Figure 6: Model Evaluation Metrics (Including Ensemble) as Percentages

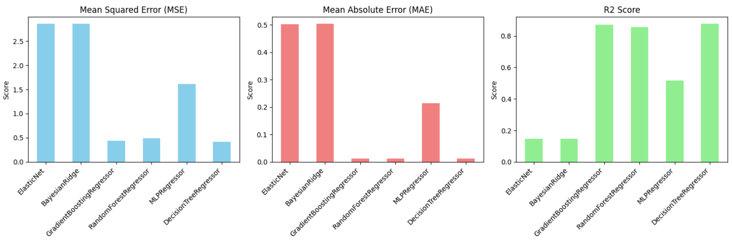

Figure 7: Bar chart of scores across each metric.

We looked further into the feature importance for the decision trees and found that among all three, the rank was the most important variable, followed by a few columns representing publishers mixed with platforms. This is a limitation of our model, as the final rankings are determined in part by the sales output, which might limit the utility of our model or explain the degraded performance when including text embeddings.

7 Discussion

We have generated an ensemble of sophisticated machine learning models in order to optimize forecasts on video game sales. In conducting this experiment, we found that our dataset was sparse and might make it difficult to learn an effective decision boundary. In this context, model selection became pivotal, as we saw night-and-day changes in performance depending on the sort of decision boundary modelled. This research demonstrates that tree-based models might be superior in categorizing forecasts which use a variety of categorical data as opposed to inputs. Future investigations into optimizing the linear-based forecasts would involve intervention at the data-generation level, involving processes which mimic the work done by Li et al. 2021 to scrape for more weekly and platform-based information. This data would be helpful in improving the performance of all of our models because it would engage rich, numerical data.

This investigation also probed into the possibiltiy of using text embeddings in order to extract useful information from titles. However, the method that we engaged was perhaps too simplistic, relying on a highly distorted representation of the sentences in order to provide numerical data on their use. As such, we cannot eliminate the possibility of text-based embeddings being useful. A different analysis might also attempt to use rule-based mechanisms for feature engineering in titles through trying to see if a game is part of a franchise. We know intuitively that franchises offer more sales volume than one-off games because they tend to succeed previous commercially successful games. Other approaches might attempt to scrape text from the synopsis of the game, engage in social media sentiment allocation à la Malvankar et al. 2023, or other strategies for more information. Other limitations in this model perhaps include the inclusion of variables that are difficult to forecast in advance, like game ranking and relative sales proportion. These variables are ultimately accumulated with a factor of number of copies sold – a game cannot rank highly if it is ultimately unknown. These factors also contribute to the general unpredictability of the video games industry, since many successful titles come from once-unknown publishers through strategies of proliferation that take advantage of decentralized spaces like social media.

However, our model performs to the standard of other precedent work in this field, and as such it satisfies the ambitions sought out in the introduction of this problem space. This work demonstrates the utility of tree-based methods in incorporating diverse, categorical information for inference. The model can be scaled into an industry context, provided some estimation mechanisms for understanding sales outcome in proportion to region and total ranking.

8 Code and Contributions

Code:

Code and data can be found here. Note that the vgsales.csv data will need to be manually updated.

Contributions:

Michael and Dani contributed in equal quantities for the production of this project. Michael was in charge of the project vision and goals, the selection of models, and preprocessing steps. Dani was in charge of model implementation, creation of charts, evaluation, formatting of the report, and the text embedding feature extraction. Writing the report was contributed equally with each member detailing the portions that they worked on.

References

Keerthana, B. and K. Rao (2019). “Issue 6 www.jetir.org (ISSN-2349-5162)”. In: *Journal of Emerging Technologies and Innovative Research* 6. URL: `https://www.jetir.org/papers/JETIR1907H50.pdf?fbclid=IwAR2SSF1LLfho_xN463ZgChm_KKjL0Qw188ZE7EtkjDg0WkV-WmGiePjCRRY`.

Li, J. et al. (2021). “Predicting Video Game Sales Based on Machine Learning and Hybrid Feature Selection Method”. In: *2021 16th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*. DOI: `10.1109/iske54062.2021.9755343`.

Malvankar, Kshitij et al. (2023). *A Case Study to Analyze the Impact of Social Media on Video Game Sales*. DOI: `10.1109/icct56969.2023.10076200`.