Executive Summary
For Dev10 Capstone Project:
Video Game Sales Data Analysis

Project by Group 6:

Members:

TJ Fugelseth, Hugh McMurray, Ryan Shell, & Jason Singh

Submitted on:

November 9, 2021

Introduction

The video game industry has been a large part of the entertainment industry, both globally and in the United States, since the mid-80s. As a result, there are large amounts of data available regarding how a game sells, how well it is received critically, and a number of other factors. As a result, we chose to examine these factors to determine what impact sales the most, and then use them in a machine learning model in an attempt to predict the sales of a game.

Exploratory Questions:

1. How much does a game's content (ESRB) rating impact its sales?
2. How much does a game's critic review score impact sales?
3. Do different consoles sell more games than others?
4. How do different regions impact game sales?
5. Do different genres sell better or worse than other genres?
6. How are global sales trending over time?
7. What factors have the most influence on game sales?

Our Data:

Our data primarily came from a Kaggle dataset. The dataset contained info for around 50,000 video games released between 1985 and 2019, including sales both globally and for the three major regions of release: North America, Europe, and Japan. While we simply had to download a csv from Kaggle to get this dataset, originally it was scrapped from VGchartz.com, which tracks sales history for a number of titles.

ETL Process:

Next, we will give a brief, high-level overview of the ETL process we used to implement and create our data platform. A more detailed description of this process can be found in the ETL Report in the GitHub repository.

Sales data of this nature is typically private due to competition, or would have to be compiled from multiple sources based on financial disclosures from the company, so we had to use the aforementioned Kaggle dataset. We simulated a data stream using Azure Databricks and Kafka, and used this data stream to set up a data factory to save the data into a SQL database. This flowchart visualizes our data platform (next page):

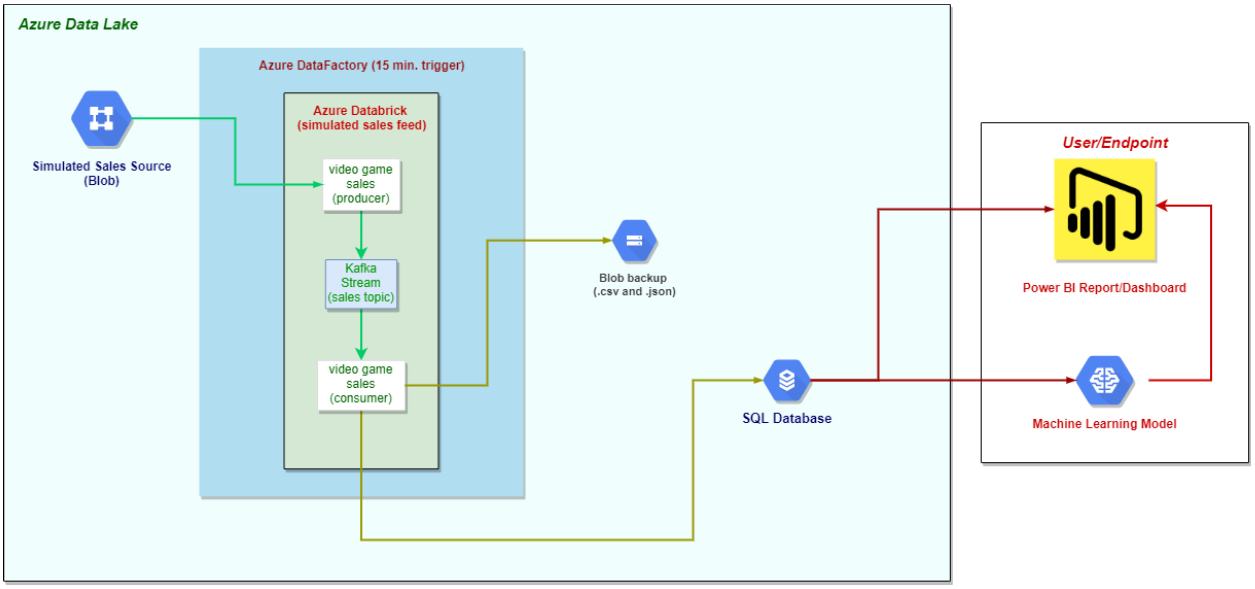
Azure Data Lake
Azure DataFactory (15 min. trigger)
Simulated Sales Source (Blob)
Azure Databrick (simulated sales feed)
video game sales (producer)
Kafka Stream (sales topic)
video game sales (consumer)
Blob backup (.csv and .json)
SQL Database
User/Endpoint
Power BI Report/Dashboard
Machine Learning Model

Figure 1: Data flow visualization

Machine Learning Modeling

Next, we will detail our main machine learning model.

Gradient Boost Regressor

The main model we chose to use was a Gradient Boost Regressor. This is an additive model, allowing for the optimization of arbitrary differentiable loss functions. In each stage of the model, a regression tree is fit onto the negative gradient of the given loss function. We chose to use a 70/30 training/test split. We used the following parameters after performing a GridSearchCV optimization:

- learning_rate: 0.001 (It shrinks the contribution of each tree by the fraction given)
- max_depth: 5 (Maximum depth of the individual regression estimators. The maximum depth limits the number of nodes in the tree.)
- max_features: sqrt (The number of features to consider when looking for the best split; sqrt means (number of features)^0.5
- n_estimators: 15000 (The number of boosting stages to perform.)

This model gave us a testing score of 0.60. The attached plots on the next page summarize the spread of the residuals from the model (right) and the goodness of fit of the model (left):

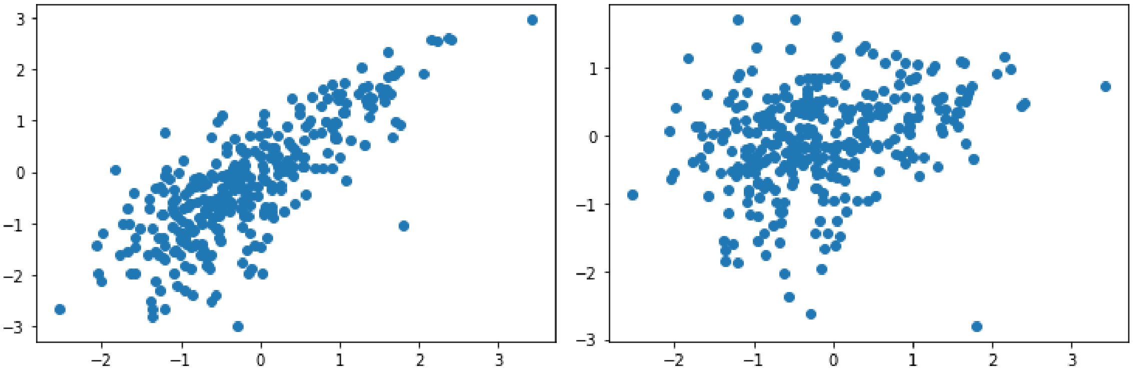

Figure 2: Goodness-of-fit (left) and Residuals Spread (right) of the Gradient Boost Regressor Model

Census Data

Another part of this project required us to incorporate census data. We chose to look at month by month sales for businesses under the electronics store classification code in the NAICS. The following graph shows the monthly sales (in millions) of the industry for every month since January 2010.

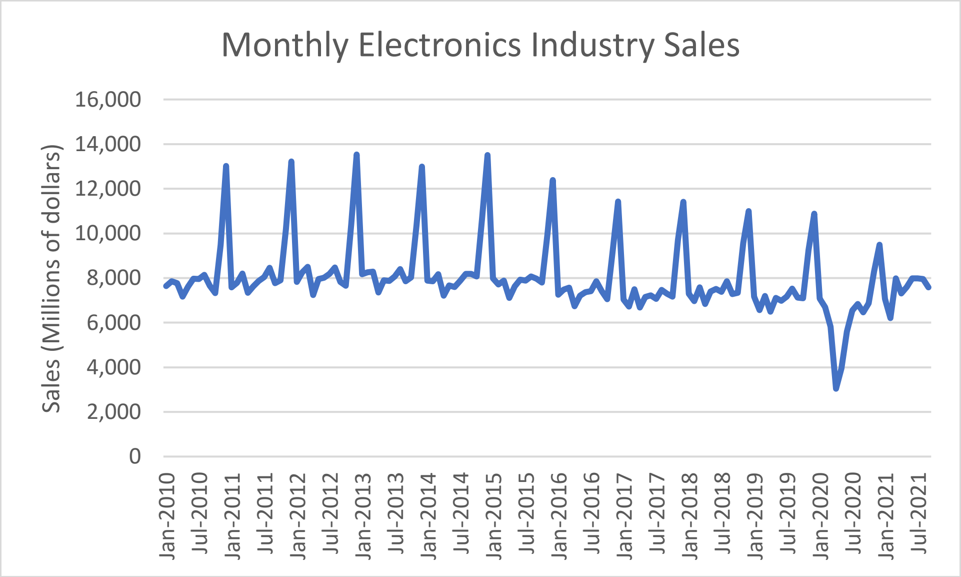

Figure 3: Electronics Industry Sales by Month, with respect to Sales (in millions)

Not surprisingly, there are massive jumps in sale each holiday season, particularly in November. Also, we can see when sales dipped due to the Covid pandemic in March 2020, as retail sales came to a halt as a result of the pandemic not allowing for in person sales, which this data accounts for.

Results & Recommendations:

Results

The first question we wanted to answer after running our model was what factors most impact game sales? To answer this, we looked at the fit parameters.

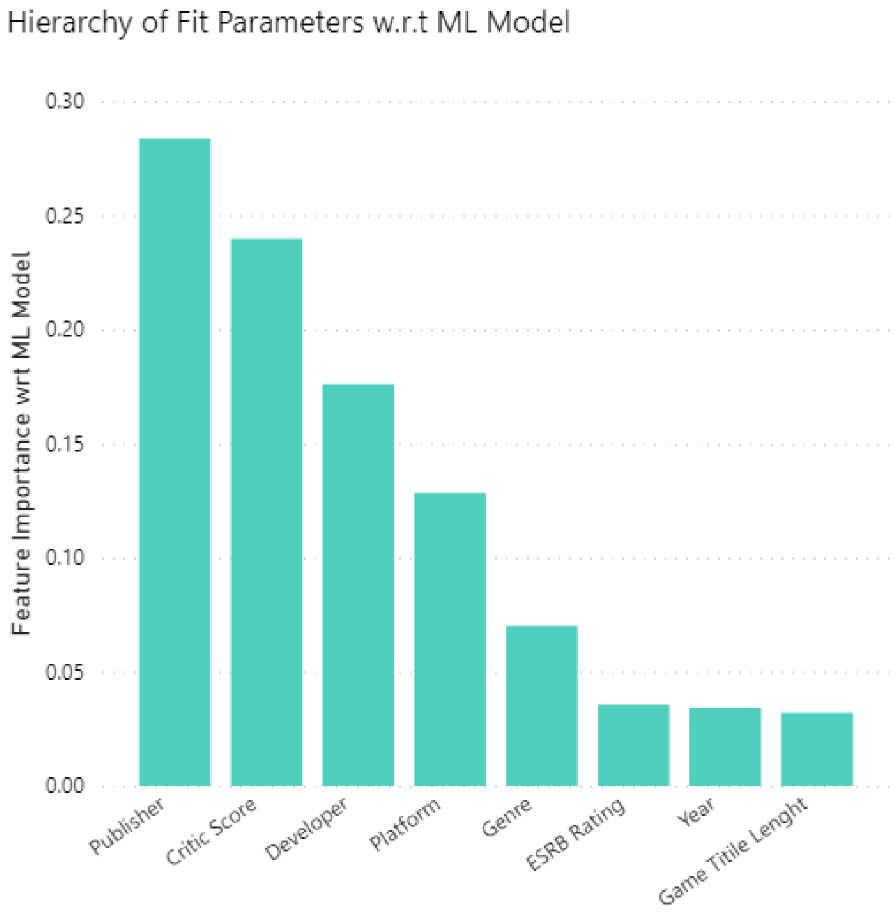

Figure 4: Fit Parameters for ML model

From this, we can see that the most important features with respect to a game’s sales are the publisher, critic score, developer, and platform.

After this, we wanted to examine the relationship between critic score, and the global sales of a game:

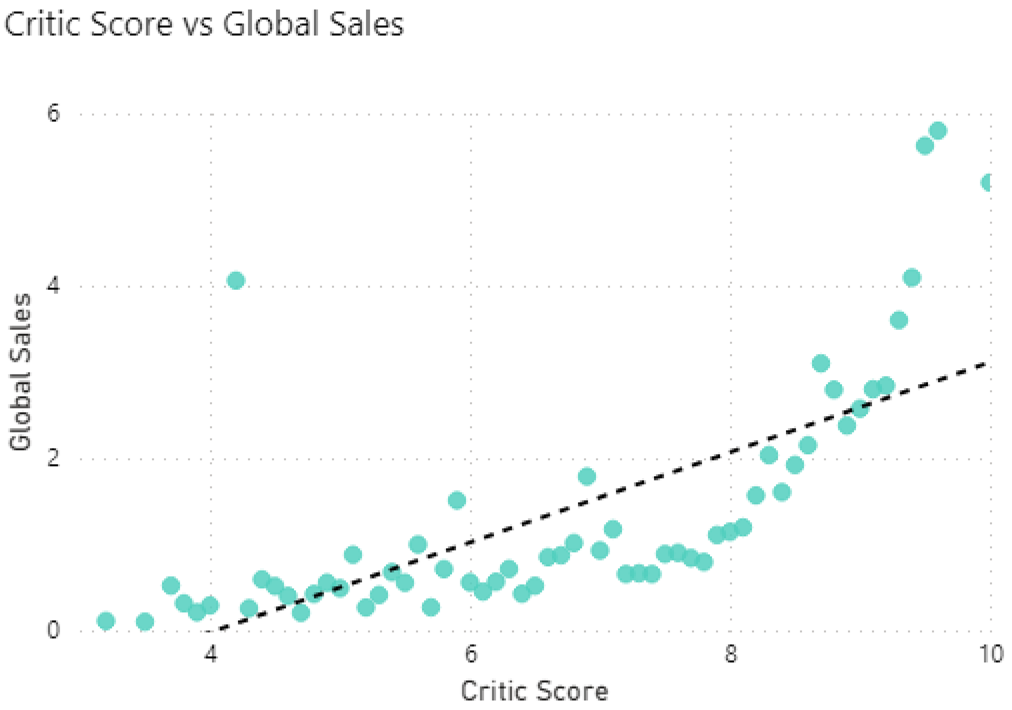

Figure 5: Relationship between Critic scores and Global sales (in millions)

Besides the obvious outlier at around 4 million sales and a critic score of around 4, higher critic scores correlate to higher sales. The next question we wanted to answer was if a game's genre impacts the sales. We have a bar chart showing the sales by genre:

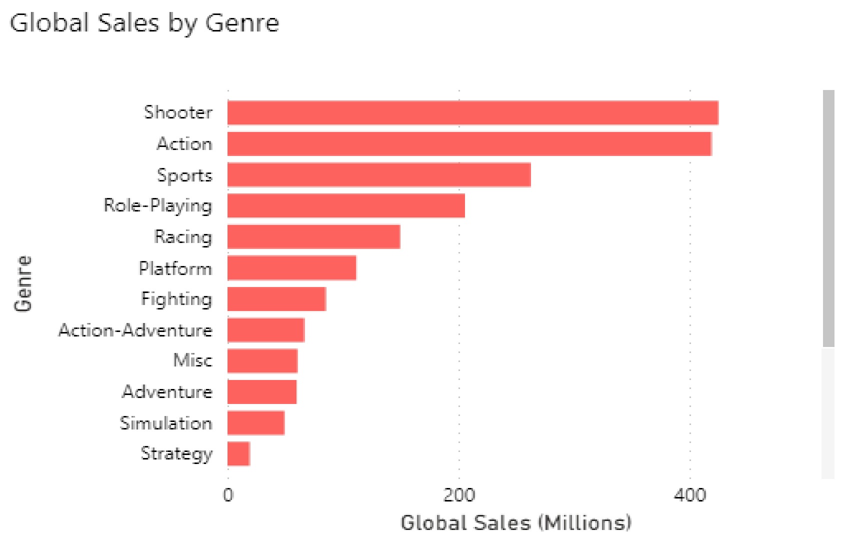

Figure 6: Global sales with respect to the game's genre

The most sold genres are shooter, action, sports, and role-playing. Other questions we wanted to answer included game sales by platform, sales by rating, and lastly, sales by region:

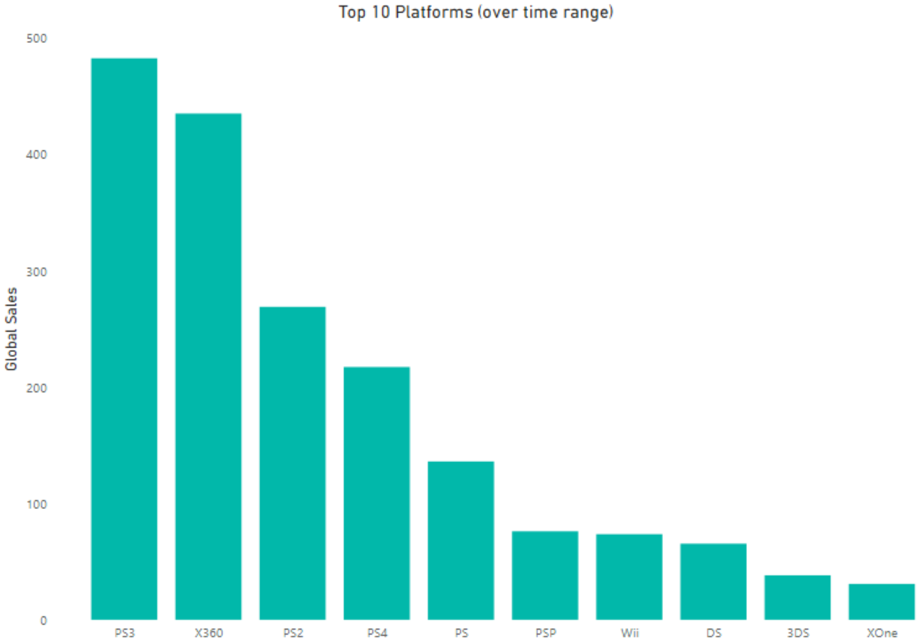

Figure 7: Top 10 platforms by total global sales for all software

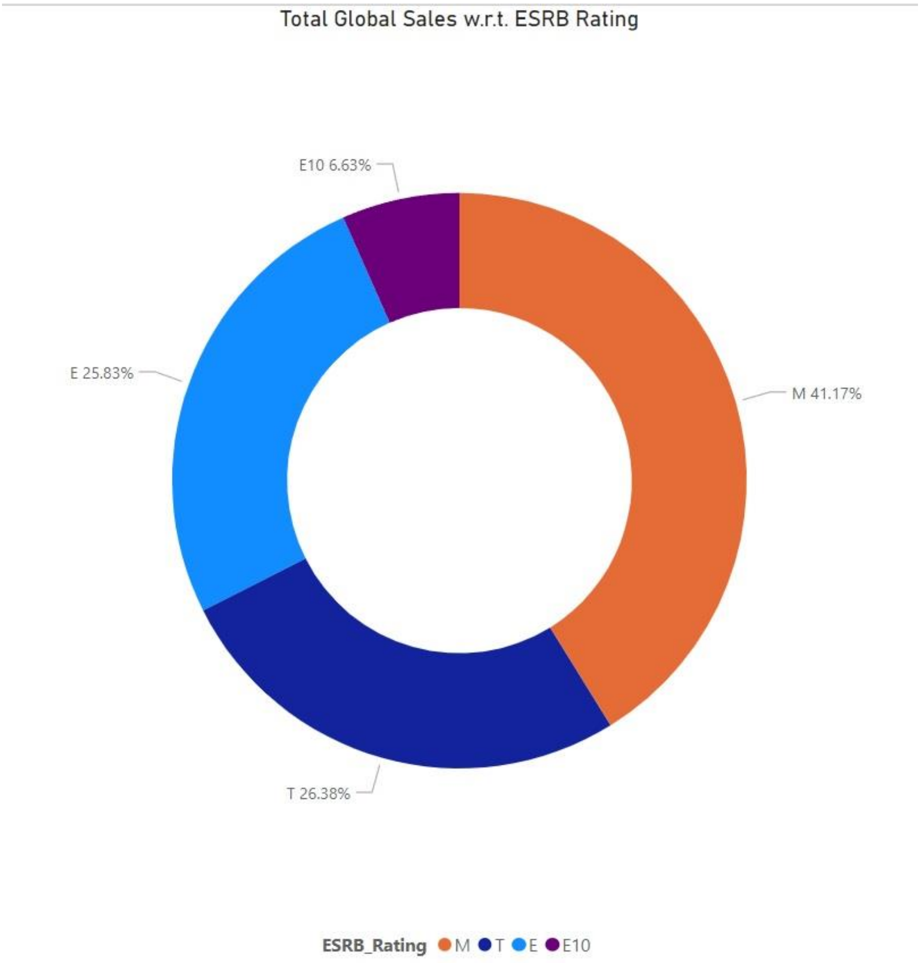

Figure 8: Global sales with respect to ESRB rating

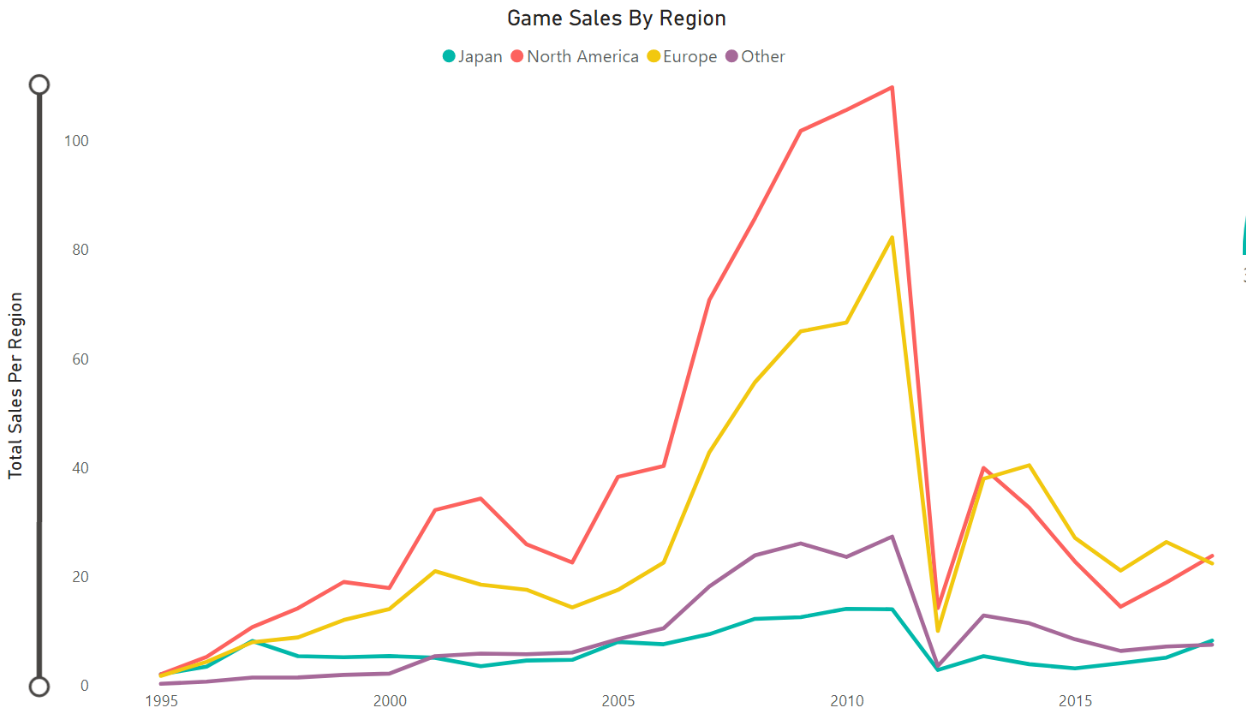

Figure 9: Regional game sales since 1995 by region

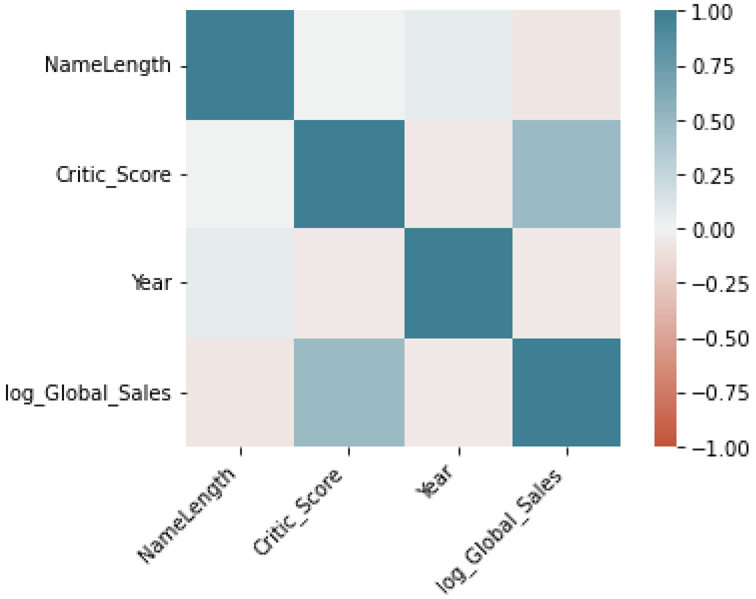

Figure 10: Correlation Matrix, non-categorical data

Recommendations:

Based on our research, we can give recommendations on how to make a top selling game. In order to sell well, developing a shooter or action game and published on the PlayStation is the key to success. Looking at figure 7, four of the five most top-selling platforms are PlayStation consoles, so selling your game on those systems is a must. We recommend the shooter or action genres because, via Figure 6, they have the highest global sales over any other genre by a significant margin.

Finally, we would recommend having your game published by a well-known and large publisher, as, while looking at Figure 4, publisher ended up being the most important feature to the model, with a score of 0.28. It goes without saying to make sure that the game is critically received, as there is a positive relationship between sales and the average critic review score. Lastly, looking at the Census data, we would recommend publishing during the holiday season, as sales for the electronics industry peak significantly during these months, and this would allow for optimal sales while the game is still at full price.

Critic score is positively correlated with global sales, via Figure 10, albeit mildly, so having a better critic reception will definitely boost sales as mentioned before, but it won’t help much. The string-length and release year are negatively correlated, but to a very small extent, so having a longer title could potentially adversely affect sales but since it’s such a small correlation, it’s highly speculative (correlation is not causation), doesn’t make sense to release your game in the past, but games seemed to have performed a bit better back then, though very slightly.

Works Cited

1.) ***Video Games Sales 2019***, "*Sales and Scores for more than 55,000 games*"**.** Retrieved from Kaggle

2.) US Census Bureau, Retail Trade: Summary Statistics for the U.S., States, and Selected Geographies: 2017. Survey/Program: Economic Census, TableID: EC1744BASIC, Dataset: ECNBASIC2017. (directlink)

Appendix A: Machine Learning Models

Here are brief summaries of the other machine learning models that we tried implementing, and their results.

Lasso/LassoCV

This was one of the first models we tried. It is a linear regressor that uses alphas as a tuning hyperparameter. To find these alphas, we set the variable in a logspace and then test to see which one performs best. The most optimal alpha was 0.001. To get more repeatable results, we specified a random state as well, and we also chose to increase the maximum number of iterations the model would run to improve performance. After running the model (and removing the publisher from the regression to increase performance), we came to a final R-squared value of 0.54, meaning that our model could predict 54% of the variability in sales based on the predictors we fed it.

Here is a scatterplot of the actual sales of a game, versus the predicted sales that our model made, with actual sales being on the x-axis and predicted on the y.

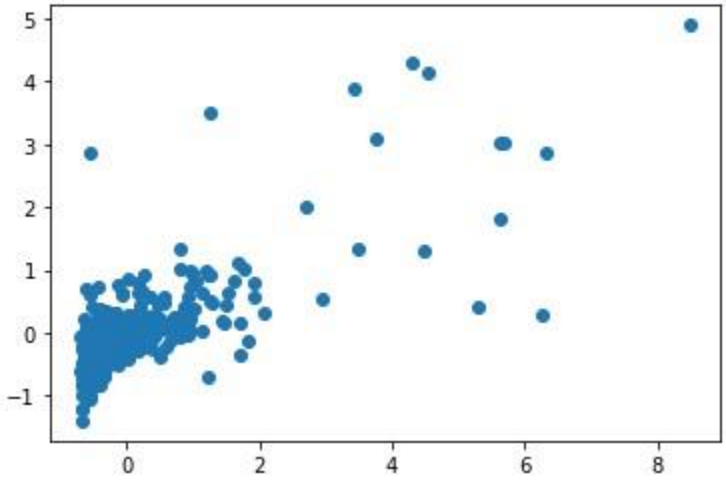

Figure 10: LassoCV Model Scatterplot, Predicted (y) v. Actual (x) Sales

Random Forest

Next, we tried a random forest model. In this model, we take random samples of our split and use averaging to improve the accuracy of the model. We used the same test/train split of the lasso cv model. The only other parameter we changed was our estimator's column, which was changed from the default value of 100 to 10, as that would result in a better score. After running the model, we got a final R-squared of 0.59, which is slightly better than what we got in the LassoCV model.

Attached is the scatterplot for the model:

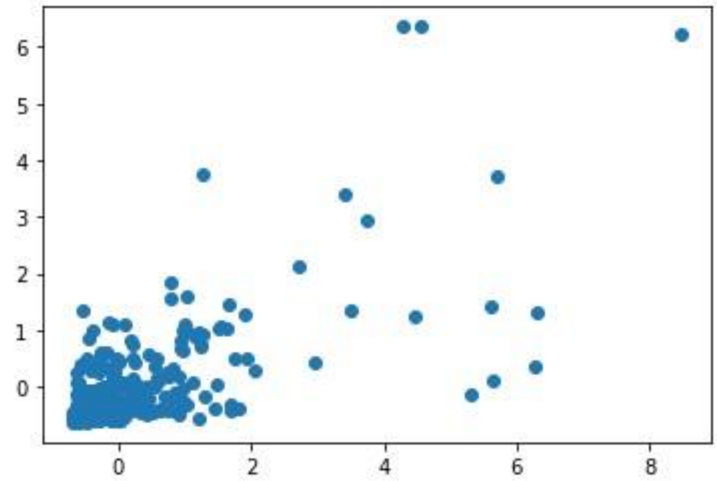

Figure 11: Random Forest Model Scatterplot, Predicted (y) v. Actual (x) Sales

Neural Network

Next, we tried a neural network regressor, again from the sklearn library. This regressor tries to optimize the squared error of the regression line. Again, we defined a random state, and this time we found that instead of messing with tuning our parameters, leaving most of them alone led to the best results. The only ones we changed were increasing maximum iterations to 500, and changing the Beta 1 variable. Our final score was 0.56, placing it in the middle of the three models we have tried running so far.

Here is the scatterplot, again with actual sales on the x, and predicted sales on the y.

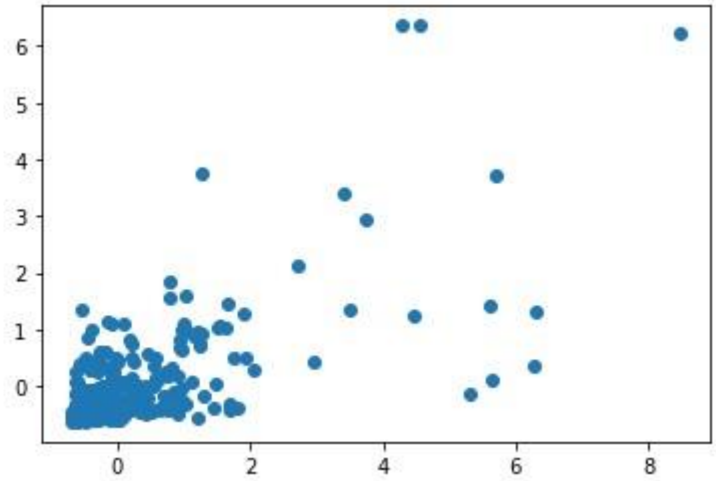

Figure 12: Neural Network Model Scatterplot, Predicted (y) v. Actual (x) Sales