Final Report: Steam Sales Price Analysis

Gavin Sorensen, Parker Seehafer, Noah Sutherland

Project GitHub

https://github.com/pseehafer/DS150Final

Motivation

Our interest in analyzing Steam prices comes from our shared interest in online games. With the extensive library Steam has to offer, we wondered how developers, whether independent or corporate, can accurately price their respective games/software on release. Furthermore, if one of us were looking to publish our own game, how could we examine preexisting Steam data to set our own price?

Problems

- The Steam Database houses tens of thousands of games, each with its own price, budget, and player data. This means that there is a lot of data for a developer to examine when analyzing how they want to create and market their game.
- With the massive amount of data varying in utility and relevance, how can the information be better organized for a small indie developer, for example, to use when predicting the price, release, and popularity of their game?

Project Description

We will be analyzing Steam game data from March 2025. The dataset contains a list of nearly 90,000 game titles from as early as 1997, with information on price, popularity, release date, reviews, and more. To aid in the release of new games, we will break down this dataset by the attributes: genre, release date, and player count. We will build linear regression models to predict the price of a future game by its attributes. The goal is to better equip new game developers with the information they need to categorize and price their future projects without needing to search through the raw, daunting Steam Database.

Data Source

Our data comes from a Kaggle CSV file on Steam Games from March 2025.
https://www.kaggle.com/datasets/artermiloff/steam-games-dataset

Methodology

- To analyze the data and collaborate with each member in the group, we used Google Colab, hosting a Jupyter Notebook with a Python environment. The Notebook is separated into sections to help us organize our analysis. The sections include: setup, explorations, functions, cleaning, and individual examination sections of different attributes. Refer to the README file in the GitHub repository for more information on the structure of the Notebook.
- The raw data was read from the CSV file into a Pandas DataFrame. Before any filtering or analysis was done, the data was examined to determine the overall structure of the DataFrame and the best attributes to compare in relation to game sales.

- Due to the large amount and variety of data, price was heavily skewed to the right. In order to achieve a more accurate prediction model, we performed several transformations on the raw data using functions created in the Notebook, including a log transformation function, a price filtering function creating a grouping of pricing, and an outlier filter function, removing data outside the upper and lower bounds of the data, respectively.
- After the raw data had been transformed/filtered to enable more accurate predictions, we chose attributes that we felt would most likely help to predict a game's price, being: release date, genre, and public opinion. These attributes were then separated into subsets of the filtered DataFrame for each member to analyze. The initial examination consisted of exploring the data in relation to price and our given attribute, developing a hypothesis based on the observed behavior, cleaning the subset if necessary, and then plotting the relationship between price and the selected attribute on a graph using Pyplot imported into the Notebook from Matplotlib.
- The final step in our analysis consisted of developing linear models to predict price based on the selected attribute and answer our respective hypotheses. The linear models were created from functions we built into the notebook, which include: a single linear regression function, a multi-linear regression function, and a linear regression function for non-numeric predictor variables. All functions returned a created model (using Sklearn imports) and predicted values based on the function used and the variables passed. These models were used to compare with our respective original data, test the accuracy of the developed model, and answer the hypothesis.

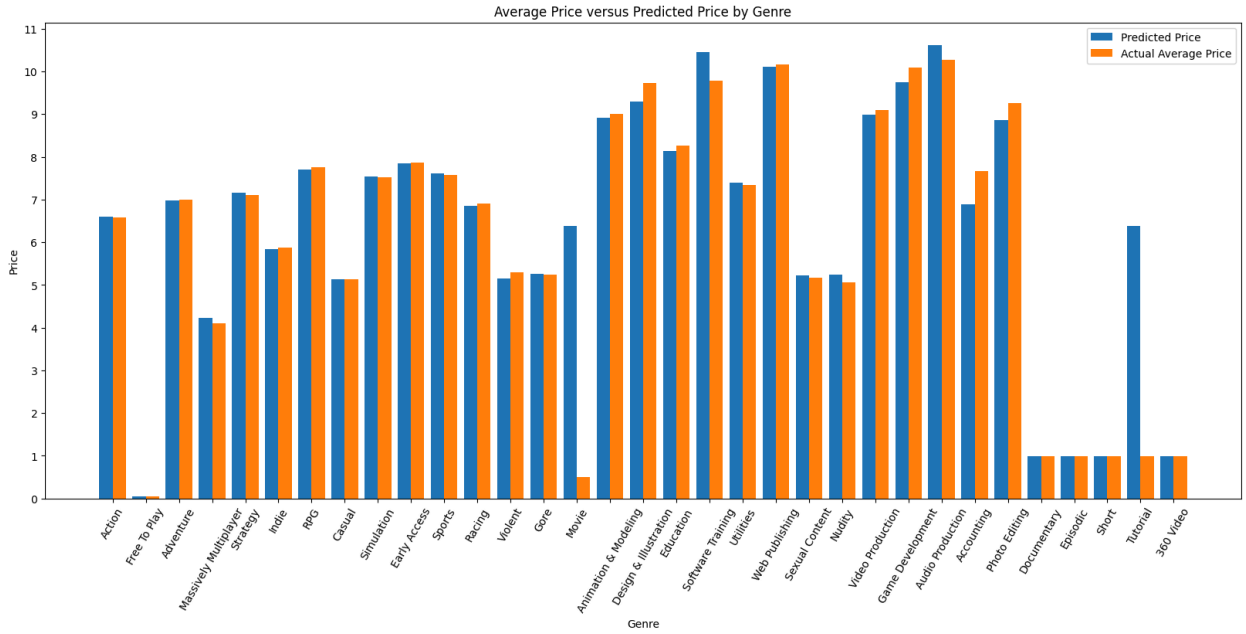
Results and Interpretations

The first analysis targeted the release data information provided in the data set. The goal was to determine the difference in release price given a month. The price data was categorized by the month of release and weighed against other months. To avoid the skew caused by free games, only paid games were filtered into this subset. When plotted together, there was no statistical difference in the price of a game and the month of its release. To better predict the price of a future game given a release month, this subset was passed into our categorical regression function to create a linear model capable of predicting a price given a month. The results of the model are displayed in the left figure. At first glance, the model is fairly accurate in predicting the price for each month. Upon further gathering the model's statistics, the discovered R^2 score and P-value were 0.001, meaning that the variance in the

release_month_name	predicted_paid_games	actual_paid_games
January	4.98	5.01
February	5.24	5.23
March	5.24	5.23
April	5.18	5.24
May	5.34	5.40
June	5.34	5.36
July	5.41	5.37
August	5.42	5.39
September	5.63	5.68
October	5.68	5.62
November	5.55	5.57
December	5.28	5.23

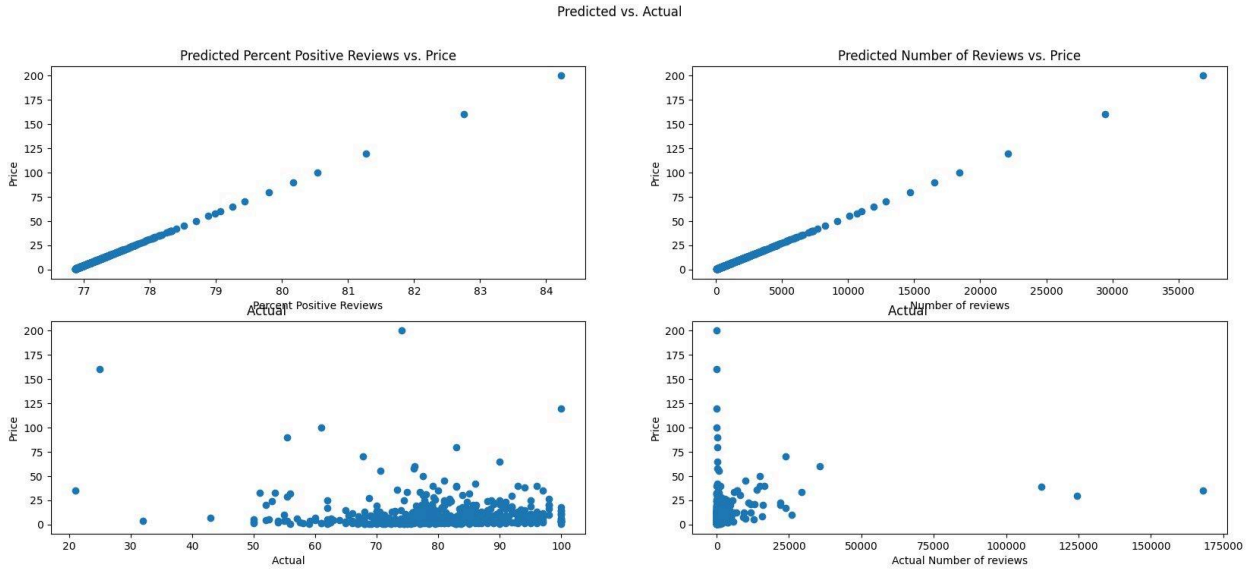
predictions can not be explained by this model, and the month is statistically unlikely that the release month can confidently predict the price of a future game.

The dataset was then broken down by genre to analyze and predict the price of a game based on its categorization. After the dataset was cleaned and grouped by respective genre, it was plotted to examine the difference between genres. The average price point for games by genre ranges around 5 to 9 US Dollars. This new subset was also passed into the categorical regression model function. Like the model created to predict price by release month, the new regression model trained predicted the price given genre fairly accurately, as seen in the figure below. This,



of course, is to be expected, as the data was trained all the entire DataFrame when filtered by individual genre. Subsequently, also like the release month model, the R^2 score and P-value were extremely low, 0.052 and 0.001, respectively. From the analysis of this regression model, we concluded that genre cannot accurately predict the price of a future game.

The last predictor examined was public opinion. The data was cleaned and transformed in the same way as the previous subsets were. Once outliers were filtered, the DataFrame was grouped into two subsets: by positive reviews and by total reviews. Those subsets were plotted, revealing heavy skew towards lower-priced games. This can be explained by the high number of games in the data set with lower prices. The subsets were then passed to a linear regression model function to predict the price of the game based on the number of positive reviews and total reviews, respectively. The predictions were plotted and compared to the actual price given reviews



subsets, seen in the figure above. The model was returned with a R^2 score of only 0.009, meaning this model cannot explain the variance between the predicted and actual prices. Consequently, we determined that the price of a game cannot be determined by reviews alone.

Limitations and Challenges

- The dataset analyzed had a very large amount of data in many categories that lacked correlation. This resulted in a large amount of skewed data, especially in the price attribute. Our main challenge at the beginning of the analysis was transforming a filtering the data to find the most effective way to predict price based on our selected attribute.
- Our regression models for each attribute had low R^2scores and struggled to confidently predict the price. The attributes passed into the models were transformed in hopes of improving accuracy, but no significant improvements were observed. Training each respective model on multiple categories may yield more effective models when predicting price.
- The dataset only contained information relating to a game post-release (reviews, playtime, etc.). The information lacking in the dataset was anything relating to the development process of the respective game (budget, development studio, etc.). With more information on game development rather than release, it may yield more effective models when predicting price points specific to development companies and game categories.

Conclusion and Future Work

Throughout this analysis, it became clear that predicting a release price of a game becomes increasingly difficult with only a single predictor category. In game creation, there are seemingly an infinite number of design possibilities. While our regression models were accurate in their predictions based on a single predictor, in relation to the entire dataset, they become obsolete. In terms of being able to accurately predict a game's price, many more than one

predictor variable will need to be accounted for. For future work on this project, multivariable correlation would enhance the effectiveness of our models. Once variables with high correlation with price and each other are selected, a larger multi-linear regression model can be built to further train our data. Furthermore, a large talking point of our research and limitations centered around a lack of predevelopment data, that is, a game's development budget, team size, resources, and development time. Only gathering the data post-release limits our ability to predict the price and success of a game, whereas having both post and pre-release information can strengthen our predictions. Of course, this presents another challenge; much of the development process is kept from the public. With this information, however, in combination with the research and analysis done here, we hope that a larger prediction model can be built, catering to specific requests in game genre, popularity, and release window, to aid future developers who want to make games of their own.