Machine Learning Model for Predicting Video Game Success

Ajay Sodhi , Yehonatan Katz

2023-04-06

Table of contents

List of Figures

List of Tables

Preface

The purpose of this project is to determine which model is the most accurate at predicting how many individuals own a game on Steam based on various characteristics such as genre, tags, categories, reviews, platform support, english support, average and median play time, developer and publisher. The following pages will outline the process of exploring the dataset, feature selection, model selection, model training, model interpretation, and the final results.

Abstract

Today the video game market offers thousands of products of many different genres, budgets, and prices. It is difficult for developers of games to get noticed on online platforms when they lack marketing budget due to the sheer size of these platforms, and the time it takes for users to determine if they want to purchase a game. The objective of this project is to provide a tool that can determine the success of a PC game based on data that is available before the game is released, such as categories and platforms. A dataset that is extracted from Steam and SteamSpy will be used to get information about games for training and testing. Data cleanup and dimensionality reduction techniques will be necessary in order to get good data to train the models with. Since we are only concerned with the current market, the database will be reduced to only include relevant titles that have been released in the last couple of years. Determining which model to use for the dataset can be a complex process as many different factors such as performance, complexity, explainability and the size of the dataset need to be taken into account. Machine learning models including Gaussian Naive Bayes, Lasso regression, KNN, and Decision Trees will be compared to determine which model best suits the dataset. the best suited model will generally be decided based upon the results of metrics like r2 score, f1-score, recall and precision as well how explainable the results are.

1 Introduction to Project

1.1 Overview

In this project, machine learning models were trained on data gathered from SteamSpy: a service that hosts information about games from the Steam platform. From this dataset, information including genres, tags, categories, reviews, platform support, english support, average and median play time, developer and publisher were used. On Steam, a tag is a descriptor of a genre, feature or content of a video game that can be added to help customers find games they might enjoy. This information will be important when training machine learning models on, because it will help determine the most popular features that well-selling games provide.

Appropriate data analysis and pre-processing were performed to get appropriate data for training with various machine learning models. The code of this project was organized in a way to make training different models efficient and without requiring change to previous data or code.

1.2 Objectives of Project

The objective of this project is to train an appropriate machine learning model to be able to predict the relative success of a video game based on its genre, categories, and features. This tool will be able to help people who make video games by exploring the popular genres and features that popular games offer. Using available data, the trained machine learning models will predict the number of owners of a game based on the categories mentioned above.

2 Pre-Processing and Exploratory Data Analysis

Once you have your dataset, you need to preprocess it to make it suitable for machine learning. This includes cleaning the data, handling missing values, encoding categorical variables, and scaling numerical features. This step is crucial as the accuracy of your model will depend on the quality of your data.

2.1 Dataset Collection

The Steam dataset was retrieved from Kaggle.com. See references for a direct link.

2.2 Data Pre-processing

Some issues to consider to be included: - There were no missing values in the steam.csv dataset. Since the dataset did not have any missing values, there was no observable pattern of missing values and they did not have to be explicitly dealt with.

- To interpret the data, one-hot encoding was utilized. To keep the performance efficient and accurate. We decided to filter out some data. For example, after the dataset was cleaned up and formatted correctly, there was a column corresponding to every tag and category on Steam. Many of these columns were extremely niche so they were removed entirely. Only the best fitting categories and tags were considered and selected using SelectKBest to ensure proper performance and to remove any unwanted bias from one-hot encoding.
- Scaling variables in this dataset were considered and tested, however, it did not yield favourable results as the performance metrics on various models were less than what they previously were without any scaling. In particular, attempts were made to scale the positive ratings and negative rating columns, as these values were quite different from other numerical columns.
- There were relatively few outliers in the dataset and they were dealt with by removing them during processing. These outliers were either columns that represented very niche categories or game titles that did not have any ASCII values within the title. For example, Some titles were listed in Mandarin instead of English which caused problems so they were ultimately removed.
- Does transforming the data simplify any analysis? Transforming the dataset using One-hot encoding played a vital role in simplifying our analysis. This is because our dataset had either values of 1 or 0 depending on if they fall under a certain category which helps as some models may provide unwanted results just using labels. One-hot encoding essentially puts all of these binary values in an easily understood array which allows it to be accurately tested on a variety of models.

2.3 Exploratory Data Analysis and Visualisations

Some of the steps, included here:

- Data visualization: Data visualization is the process of creating graphs and charts to help you understand the patterns and trends in the data. This step can help you identify outliers and anomalies in the data. Several attempts were made to better visualize which categories seemed to correlate to a higher number of owners. A bar graph was made to visualize the difference between how many single-player games and multiplayer games make up the total number of games owned. The results were that the majority of games owned on Steam were single-player games which insinuates that individuals may be more likely to purchase a single-player game. Another bar graph was made to show the number of owners for various price ranges. This graph was made to observe if individuals were more likely to purchase a game if it was on the cheaper side of the spectrum rather than a premium price. This likely turned out to be true as the majority of games owned were in the price ranges of free-to-play or less than $15. Another bar graph was created to show if certain developers tend to be more successful when it comes to selling their games. This graph took the total amount of owners from each developer and compared them against each other to find the 5 most successful developers. This graph was disproportionate, as Valve had the largest number of owners for their games. For a different perspective, another graph was created that excluded any free-to-play games. The result is different and the developers are much closer to each other in terms of owners of their games. A scatterplot graph was created to find a correlation between the median playtime of a game and its price as player retention can be a valuable metric of a game's success. The results of the graph suggest that individuals are more likely to put more time into free play and games on the cheaper side.
- Statistical analysis: Statistical analysis involves applying statistical methods to the data to identify patterns, trends, and relationships. This step can help you identify correlations between variables and understand the distribution of the data. This was covered under data visualization.
- Feature selection: Feature selection is the process of selecting the most important variables that will be used in the machine learning model. This step can help you identify which variables are most predictive and which variables can be ignored. It was decided that the most important variables in this dataset would be category/tag, the number of positive ratings, the number of negative ratings, and the price of the game. A column representing the number of achievements of a game was ultimately dropped as it held little signficance. Each tag, genre, and category was represented as a column and this turned out to be an issue because there was a significant amount of them. Many of these columns also only corresponded to an extremely small percentage of the total games on the database. It was decided that these columns were to be dropped as it could skew results and these games would still be represented by other more frequently occuring tags and categories. The final processed dataset contained a sample of tags and categories according to SelectKBest. The decision to filter out entries that were before a certain date was to ensure that the data being analyzed is relevant to the current market.
- Dimensionality reduction: Dimensionality reduction is the process of reducing the number of variables in the data set. This step can help you reduce the complexity of the model and improve its performance. When the raw dataset was imported, categories, genres and tags were in a different

format that likely would have causes problems later on. These values were essentially a string with the different categories/genres/tags separated by semicolons. In an effort to simplify the dataset, each string was split by semicolons, and each result was given its own column to make things more linear.

3 Methodology

3.1 Platform and Machine Configurations Used

A Visual Studio Code environment was used with the Anaconda distribution. Every python script was run on a local machine using the environment.

3.2 Data Split

Data splitting was performed on the final processed dataset, with a test size of 0.2 and a random state 42.

3.3 Model Planning

Different machine learning regression models were used when training on the dataset. The small size of the data allowed us to test multiple models efficiently. The models used in our testing were:

- Logistic regression
- Linear discriminant analysis
- K-Nearest Neighbour
- Decision trees
- Lasso regression
- Gaussian Naive Bayes

All models were tested to find the most accurate model.

3.4 Model Training:

The models were trained on the data with appropriate parameters. Different feature selection techniques using sklearn were used to find the best data for use with the models: variance threshold selection and univariate feature selection. Models were fitted and trained to predict for the "owners" metric, an estimated number of owners of the game on the platform.

3.5 Model Evaluation

Model evaluation was done using four statistics:

- r2 score
- f1 score
- recall
- precision

After training all of the models, they were all evaluated using those statistics for comparison.

3.6 Model Optimization

Optimization was performed during the data processing and feature selection phases.

3.7 Final Model Building

Comparison graphs were plotted after all model evaluation was done:

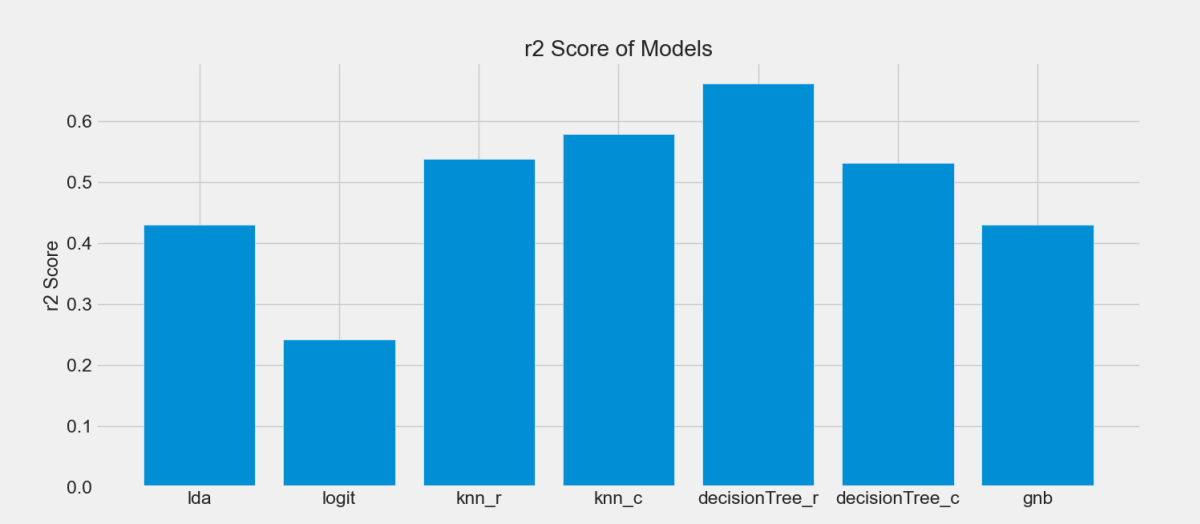

Figure 3.1: r2 / accuracy

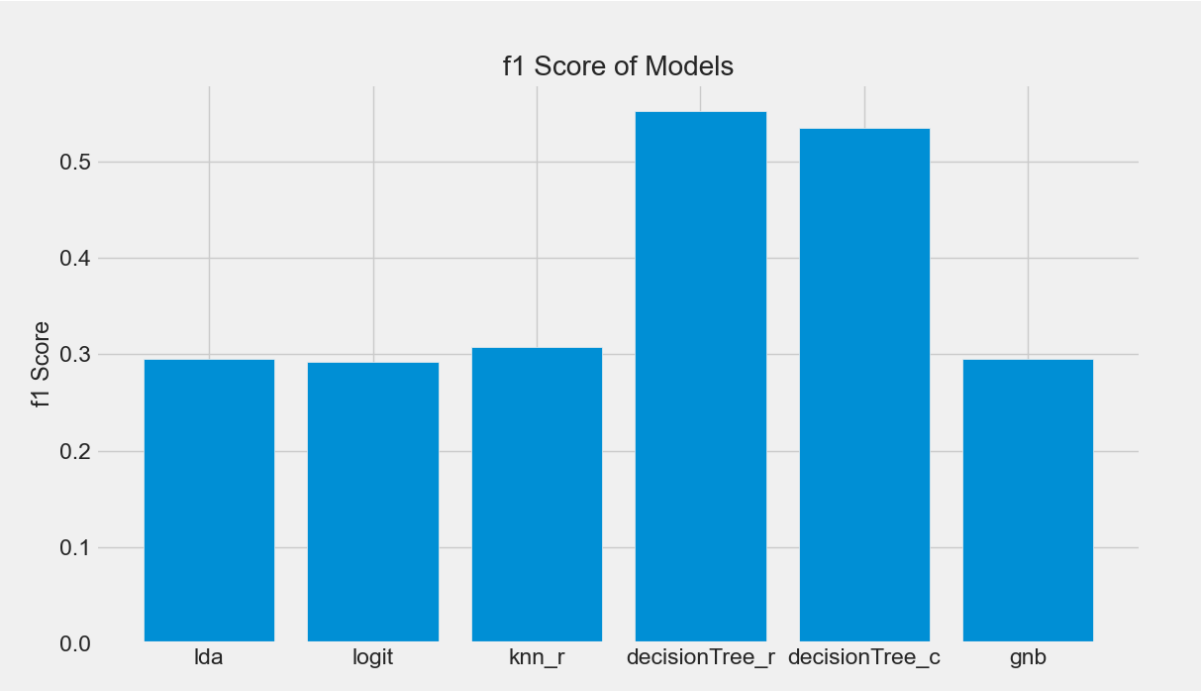

Figure 3.2: f1 scores

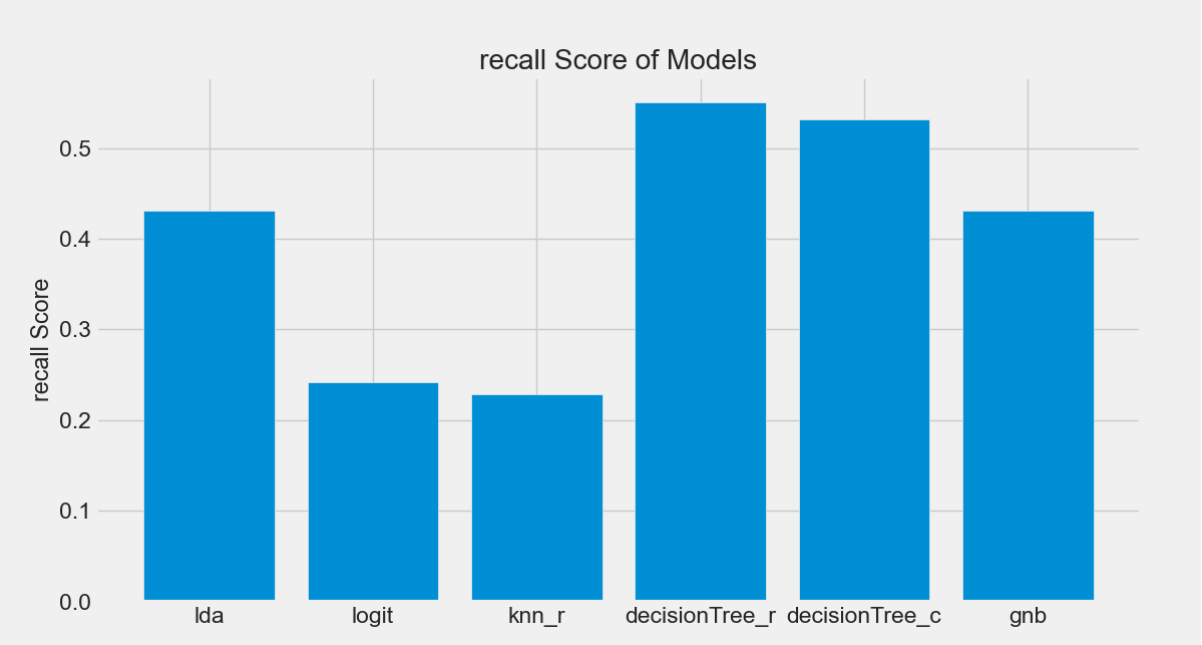

Figure 3.3: recall

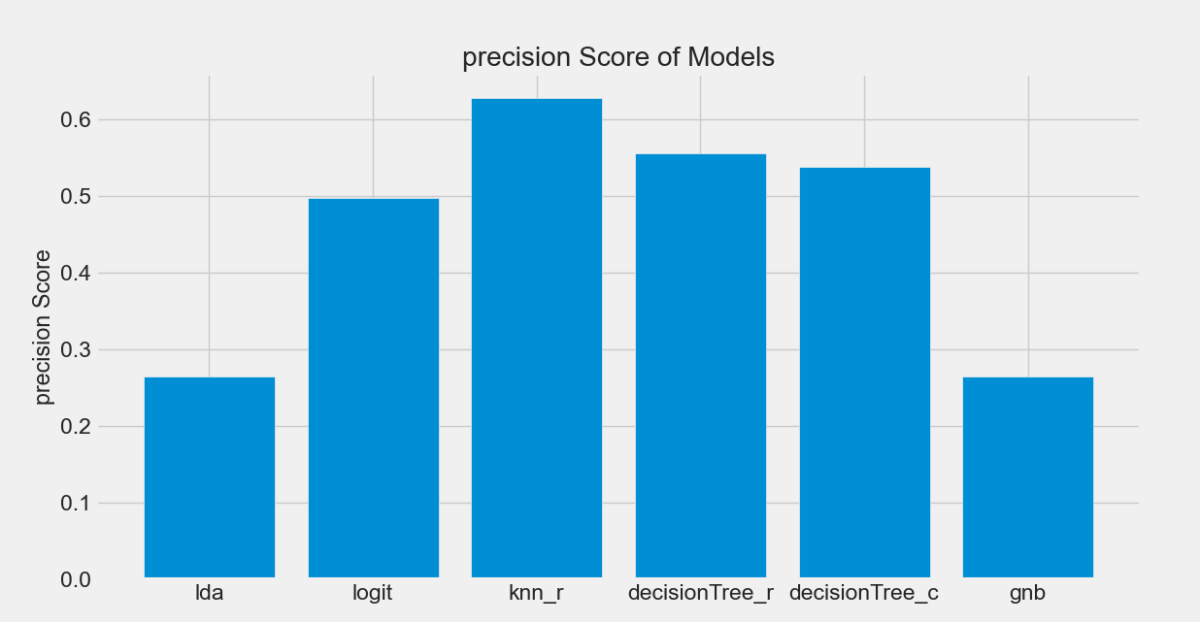

Figure 3.4: precision

4 Results

To compare different machine learning models in the results section, here are some steps to consider:

- Accuracy vs r2 score: In the case of our data the variable we are predicting comes in set sizes, so both regression and classification models were appropriate to use. Therefore, r2 score was required for regression models and accuracy was required for classification models. This difference needs to be kept in mind with the comparisons.
- Computation time: With the data used in this project, pre-processing took relatively significant computation resources, however model fitting was efficient and could be done in a few minutes. For the purpose of this project, all models can be considered as similar in computation time.
- Precision and recall: The rate of positive indications and the rate of true positives are not as significant as accuracy because they are not critical for the task. Therefore accuracy or r2 score must be considered more important than precision and recall.

4.1 Description of the Models

Models used in this project:

- Logistic Regression - a regression model that estimated the parameters of a logistic model according to the trained data. This model was included for the sake of comparison with more appropriate models.
- Linear Discriminant Analysis - A machine learning model that is used to find a linear combination of parameters that separate between data. It is similar to principal component analysis.
- K-Nearest Neighbour - A machine learning model that determines distinctions between classes by using the distance between their "neighbours" of the same type.
- Decision Trees - A machine learning method where a custom tree structure is mathematically made in a top-down, greedy approach. Important parts of this structure are: root node, decision nodes, terminal nodes, splitting policy.
- Lasso Regression - A regulairzation technique that is used with regression methods to make a more accurate prediction. LASSO stands for Least Absolute Shrinkage and Selection Operator.
- Gaussian Naive Bayes - Also known as GNB, it is a machine learning technique that uses the Gaussian distribution. It assumes that class densities are normally distributed and uses that information to predict.

4.2 Performance Metrics

Describe the performance metrics used to evaluate the models, such as RMSE, MSE, accuracy, precision, recall, F1 score, or area under the receiver operating characteristic curve (AUC-ROC), etc based on the model class.

- R2 or accuracy: Used to determine the proportion of correct predictions the model made.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$R2 = 1 - \frac{SumOfSquaresOfResiduals}{TotalSumOfSquares}$$

- f1 score: A measure of accuracy that combines the precision and recall scores of a model.

$$F1 = \frac{2 * precision * recall}{precision + recall}$$

- Recall: The proportion of positives that were identified correctly. It is important when identifying positives is critical.

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives}$$

- Precision: The proportion of true positives (accurate, positive guesses) over all positives that the model predicted. The proportion of It is significant when positive predictions have to be accurate, and avoiding false positives is necessary.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

4.3 Results Table

Model	R2 / Accuracy	F1	Recall	Precision
Logistic Regression	0.2421	0.2918	0.2421	0.4976
LDA	0.4307	0.2949	0.4307	0.2642
KNN Regressor	0.5386	0.3071	0.2282	0.6281
KNN Classifier	0.5786	0.5603	0.5786	0.5520
DT Regressor	0.6623	0.5525	0.5501	0.5562
DT Classifier	0.5320	0.5345	0.5320	0.5383
Gaussian NB	0.4307	0.2949	0.4307	0.2642
———————	—————	———	———	————
Mean	0.4879	0.4051	0.4275	0.4715
Std	0.1254	0.1249	0.1325	0.1359

4.4 Interpretation of the Results

The Decision Tree model appears to have the best performance out of all models that were tested. When running the tests multiple times, the KNN model would sometimes achieve a greater accuracy than the Decision Tree model. For the end result, Decision Tree was chosen as the best model for the objective.

The final model did not achieve the specified accuracy, this could be due to multiple reasons. One potential source of error was insufficient parameter tuning and cross validation. Basic hyper-parameter tuning was experimented with in the early stages of the project, but did not yield significant results so it was not included in the final models. Another potential source of error is the data that was used. Number of owners was the metric that was predicted by the machine learning models. Because Steam sales for video games are not publicly avaialable, SteamSpy provides an estimated metric of owners as a range between two numbers (e.g. 20,000-50,000). As this metric is just an estimate and not an accurate report, the accuracy of the models is lower than if real sales numbers could be used. The majority of the games in the dataset are niche, small, or otherwise not popular, public sales data could not be found for them, but only for the larger games. This would not have helped for the purpose of our project, so data for larger games was not included. Another potential source of error is how the "owners" metric was calculated. Different predictions of the number of owners such as minimum, mean, etc. could have potentially provided greater accuracy.

4.5 Visualization

see (**chapter-3.7?**) for visualization of the model reports.

4.6 Sensitivity Analysis

Various attempts were made at changing certain aspects of the process in order to obtain the best performance metrics. For example, entries in the data set were filtered based on their release date and the result was chosen based on performance and relevance to the current market. Numerous attempts were made at changing the value of SelectKBest however, the current value was found to yield the best performance metrics.

5 Conclusion

In conclusion,the decision tree regressor model was chosen as it yielded the best performance metrics on the dataset. When comparing accuracy, r2_scores, precision and f1_scores of various models, the decision tree model regressor consistently performed the best. It seems the data set benefits from the various strengths of decision trees such as being able to handle non-linear relationships as well as handling both numerical and categorical data. This implies that the dataset contains non-linear relationships between variables and a mixture of both categorical and numerical data which provides a possible explanation for the poor performance of the other models aside from KNN. Future research on this topic should put more emphasis on obtaining accurate values for the number of owners since the current estimated value is an estimation between a broad range. An alternative solution could possibly be to replace the number of owners column with a sales column as it may be easier to track.

References

https://www.kaggle.com/datasets/nikdavis/steam-store-games/code James, G., Witten, D., Hastie, T., Tibshirani, R. (2022). An introduction to statistical learning: With applications in R. Springer. ::: {#refs} :::