Predicting the Success of Indie Games on Steam Using Metadata and Machine Learning Models

Derek W. Graves

Northwest Missouri State University, Maryville MO 64468, USA
S573443@nwmissouri.edu and derek.graves4@outlook.com

Abstract. This study aims to predict the commercial success of indie games on the Steam platform by analyzing game metadata. Utilizing data collected from the Steam API, machine learning techniques, including Random Forest and Logistic Regression, were implemented to identify significant attributes contributing to game popularity. By exploring factors such as gameplay features, pricing strategies, and developer information, this research aims to offer practical insights for indie developers looking to enhance player engagement and maximize their games' market success.

Keywords: machine learning · data analytics · Steam · indie games

1 Introduction

The purpose of this research is to analyze and predict the success potential of indie games on Steam based on available metadata. By utilizing the Steam Web API to gather data, machine learning models—such as Random Forest and Logistic Regression—were applied to explore the attributes that contribute most significantly to a game's popularity. The ultimate aim is to provide developers with actionable insights to optimize their games for improved audience engagement.

1.1 Research Goals

The primary goals of this study include:

- Identifying which game metadata features are correlated with the success of an indie game.
- Applying predictive machine learning techniques to forecast game success based on these features.
- Providing meaningful insights that indie game developers can use to enhance the reception of their games.

2 Data Collection

The dataset for this study was collected using the Steam Web API, which offers comprehensive metadata for all games on the platform, including gameplay features, pricing, developer details, and user engagement metrics. Documentation from the Steam API Documentation was referenced to understand the API's various parameters and response formats.

2.1 Source of Data

The dataset consists of data from indie games released between 2010 and 2024. Data was gathered from various regions, including North America, Europe, and Asia, capturing a diverse set of game genres such as action, role-playing, puzzle, and adventure. This broad spectrum helps ensure that the analysis captures different aspects of indie game success.

2.2 Data Extraction Procedure

The dataset was extracted using Python, leveraging the Requests library to interact with the Steam API. The specific procedure involved:

- **API Integration:** Metadata was retrieved by calling the Steam Web API through the `Requests` library in Python, which provided access to detailed information about each game.
- **Rate Limiting and Retry Logic:** To avoid exceeding API limits, pauses were added after every 10 requests. A retry mechanism was implemented using the `Retry` feature from the `urllib3` package, which helped mitigate server timeouts and manage transient errors effectively.
- **Data Filtering:** The dataset was filtered to focus only on completed indie games, explicitly excluding adult content, demos, downloadable content (DLC), and games still in early access.
- **Data Storage:** The collected data was saved in CSV format, containing both the full dataset and a balanced subset that focused on games with varying levels of popularity for further analysis.

2.3 Data Format and Volume

The initial collected balanced dataset comprises 300 game records, each containing seven attributes: `AppID`, `Game Name`, `Release Date`, `Developer`, `Genres`, `Price ($)`, and `Recommendations`. Games were classified into three popularity tiers—low ($\leq$50 recommendations), moderate (50–500 recommendations), and high (more than 500 recommendations). Any entries with incomplete metadata were excluded to maintain data quality.

2.4 Dataset Attributes

Table 1 provides an overview of the dataset attributes:

Column Name	Description	Data Type	Example Value
AppID	Unique identifier for each game	Integer	440
Game Name	Title of the game	String	Team Fortress 2
Release Date	Date when the game was released	String	October 10, 2007
Developer	Developer(s) of the game	String	Valve
Genres	Genres associated with the game	String	Action, Free-to-Play
Price ($)	Price of the game in USD	Float	19.99
Recommendations	Number of recommendations received	Integer	50000

Table 1. Attributes of the Indie Games Dataset

2.5 Other Considerations

Challenges during data collection included handling incomplete metadata for certain games and managing the rate limits imposed by the Steam API. Future work could consider integrating data from other platforms, such as the Epic Games Store, to broaden the analysis and improve model performance.

3 Data Cleaning and Curation Process

The data cleaning and curation process involved multiple stages, from filtering raw data obtained via the Steam API to balancing and final cleaning. Figure **??** illustrates the overall workflow.

3.1 Tools and Techniques for Data Cleaning

Python was chosen for its versatility in handling large datasets and for its robust libraries, including `pandas` for data manipulation and `requests` for API calls. During data collection, we integrated error-handling mechanisms with the `urllib3` library, which allowed us to implement rate limiting and automatic retries, ensuring stable data retrieval. Techniques such as one-hot encoding were used to transform genre information into a format suitable for machine learning models, as outlined by Bellavista et al. [1].

In addition to the initial Python script used to extract and store data, we utilized Jupyter Notebook to further clean and curate the data. Several additional Python libraries, such as `numpy`, `matplotlib`, and `seaborn`, were employed for data visualization and exploration. These tools provided visual insights into the distribution of various attributes, helping us identify and handle inconsistencies or outliers effectively.

Jupyter Notebook was particularly suitable for iterative data cleaning, as it allowed for an interactive approach to transforming and analyzing the data.

The use of visualizations made it easier to ensure data quality, reproducibility, and transparency throughout the cleaning process. The `pandas` library was instrumental in efficiently handling missing data, data aggregation, and applying transformations, which significantly streamlined the data preparation phase. The `seaborn` library was particularly useful for visual insights, allowing us to quickly identify patterns and anomalies in the data.

3.2 Handling Missing Values

Missing values were encountered primarily in the `Metacritic Score` attribute. To ensure that incomplete entries did not bias the analysis, we filled missing values with the median score. This approach minimized potential skew, as the median is less sensitive to extreme values than the mean. Median imputation was chosen over other strategies, such as mean imputation, due to its robustness against outliers, which are common in game ratings.

For other critical attributes, such as `Recommendations`, any games with missing data were removed from the dataset during the data collection phase to maintain dataset integrity. A total of 12 records were excluded due to missing values, which helped in preserving the overall quality and reliability of the dataset.

We used a custom Python script and the `pandas` library in Jupyter Notebook to handle missing values efficiently. Visualizations, such as histograms and box plots, were generated to identify missing or inconsistent data, which enabled us to make informed decisions during data cleaning.

3.3 Attribute and Record Definitions Post-Cleaning

After the cleaning process, the final dataset contained 288 records and 29 attributes. Initially, 31 attributes were extracted, but two columns—`Nudity` and `Sexual Content`—were removed to suit an academic context. The cleaned dataset included attributes that were essential for predicting game success, such as `Recommendations`, `Price`, and `Genres`. These attributes were selected based on their expected impact on game popularity:

- **Recommendations:** This attribute was crucial for predicting popularity, as user recommendations are a strong indicator of player satisfaction and engagement. Ensuring complete data for `Recommendations` helped maintain the quality of our model predictions.
- **Price:** The `Price` attribute was normalized to reduce variance and ensure it could be effectively utilized by the machine learning models. Pricing strategies can significantly influence game success, making it a key factor in our analysis.
- **Genres:** The `Genres` attribute was one-hot encoded to transform categorical data into a numerical format suitable for machine learning algorithms. This allowed us to capture the influence of game genres on popularity.

3.4 Usefulness to Research Goals

The cleaned dataset was tied to the research questions initially posed by ensuring that key attributes relevant to game success were retained and transformed appropriately. For example:

- Ensuring complete data for `Recommendations` allowed for accurate predictions of popularity, as it served as a key independent variable in our models.
- The median imputation of the `Metacritic Score` was chosen to maintain continuity for games with otherwise complete metadata, reducing the impact of missing values on model accuracy.
- The one-hot encoding of `Genres` ensured that genre diversity was effectively represented, allowing the models to capture any correlations between game genres and success metrics.

3.5 Independent and Dependent Variables

To provide clarity on the analysis, the following independent and dependent variables were explicitly identified:

- **Independent Variables:**
  - `Price`
  - `Genres` (one-hot encoded)
  - `Developer`
  - `Release Date`
- **Dependent Variable:**
  - `Recommendations` (used as a proxy for game success/popularity)

4 Model Development and Performance Assessment

The dataset was split into training and testing subsets in an 80:20 ratio. Random Forest and Logistic Regression models were then used for predictive analysis:

- **Random Forest:** Selected for its robustness and ability to manage datasets with high variance. It was also suitable for capturing non-linear relationships in the data.
- **Logistic Regression:** Chosen for its interpretability, especially useful for understanding the relationship between features and game success. This model worked well for lower-dimensional datasets, as discussed by Kirasich et al. [2] and Lounela [3].

4.1 Training Details

The training data was used to fit both models. Model performance was evaluated using metrics such as accuracy, precision, recall, and F1-score. Hyperparameters for both models were optimized using grid search, which helped fine-tune the models and enhance prediction accuracy.

5 Results and Discussion

Upon training the models, insights into feature importance were gained. For instance, the Random Forest model showed that the number of recommendations and the price were critical in predicting game success. Visual representations, such as feature importance plots, will be incorporated to further illustrate model performance and findings.

6 Limitations and Future Work

The study faced limitations, including potential biases in the metadata due to the exclusion of certain game types (e.g., adult content and early access). Furthermore, constraints on niche game availability may limit the generalizability of the findings. Future research could involve incorporating data from additional platforms (e.g., Epic Games Store) and utilizing deep learning models to improve predictive accuracy.

7 Conclusion

This study explored the use of machine learning in predicting the success of indie games on Steam. By identifying which features most significantly influence game popularity, actionable insights are provided to indie developers to help them optimize player engagement and commercial outcomes.

Additional Resources

For more details, please refer to the project resources below:

- Overleaf Report
- GitHub Repository
- GitHub Data Directory
- Steam API Documentation

References

1. Bellavista, P., Corradi, A., Stefanelli, C.: Mobile agent middleware for mobile computing. Computer **34**(3), 73–81 (2001). https://doi.org/10.1109/2.910896
2. Kirasich, K., Smith, T., Sadler, B.: Random forest vs logistic regression: Binary classification for heterogeneous datasets. SMU Data Science Review **1**(3), Article 9 (2018), bluehttps://scholar.smu.edu/datasciencereview/vol1/iss3/9, creative Commons License
3. Lounela, K.: On identifying relevant features for a successful indie video game release on steam. Master's Programme in Department of Information and Service Management (2024), bluehttps://aaltodoc.aalto.fi/items/d578980e-71fa-4618-b500-dff30bbac490