What Makes a Well-Received Game? Predicting Player Reception on Steam

Aryan Raj Dhawan, Alvis Lee, Amanda Jen, Rainah Allen
University of Rochester
April 2026

***Abstract*—We investigate whether game metadata available at or before a Steam release can predict community reception, framed as both a three-class rating classification task and a continuous Wilson-score regression. Using a dataset of 26,821 games with 499 raw features, we apply Random Forest importance thresholding and correlation pruning to reduce to 38 predictive features, target-encode the high-cardinality developer column within cross-validation folds, and evaluate five classifiers and three regressors under stratified 5-fold cross-validation. Random Forest and XGBoost achieve the best 3-class test accuracy of approximately 69.4% and the later obtained a regression $R^2 = 0.31$ on the Wilson score. K-Medoids clustering on a 5,000-game stratified sample using Gower distance reveals that unsupervised groupings reflect commercial scale rather than quality, motivating the supervised framing. An interactive Streamlit application is deployed publicly for exploration and prediction.**

***Index Terms*—game success prediction, Steam, Wilson score, gradient boosting, K-Medoids, Gower distance, target encoding, stratified cross-validation**

I. Problem Formulation

A. Industry Context and the Steam Ecosystem

Since 2000, the video game industry has seen seismic growth, transforming the hobby from niche to mainstream. A major contributing factor to this expansion has been the rise of digital distribution platforms, which allow players instant, worldwide access to games, removing the need for physical copies. Within this broad landscape, the PC gaming market has emerged as a significant segment, driven by platform flexibility, diverse game libraries, and the ability for players to customize their experience through social communities.

Integral to this ecosystem is Steam, the leading digital distribution platform for PC games, which controls approximately 75% of the global PC digital distribution market and 130 million monthly users (8). With over 8,000 games released on the platform in 2019 alone (5), Steam reflects just how competitive and saturated the market has become. Beyond functioning as a distribution platform, Steam itself generates a uniquely rich database covering a wide range of game-related characteristics, as well as player-motivated activities such as reviews and player-applied tags. This makes it an exceptionally well-suited environment for studying the factors that shape player attitude and reception, a consequential question for developers and publishers seeking to understand what distinguishes well-received games in an increasingly crowded market.

B. Data Challenges and The Research Gap

With the wealth of data available through API tools such as SteamSpy, identifying relevant and investigating unexplored factors driving player reception remains a difficult task. Games vary widely across many attributes ranging from genre, price, platform support, technical requirements. These attributes interact in complex ways, as games with similar characteristics and features can often exhibit starkly different reception and success outcomes. Additionally, the high dimensionality of the data, combined with the mix of structured features and player-generated data such as review tags, makes it challenging to develop models that fully capture player preferences. Although player reception does not entirely address a game's success (commercial or otherwise), prior research on digital marketplaces demonstrates that online consumer reviews are significantly influential on games and internet experienced players (9).

Preexisting research has made an attempt to identify what contributes to a game's success, but many lack coverage on games with a smaller player base. Several factors like player retention, price, early access, revenue, reviews, etc. have been researched in search of finding what factors make a game popular (5). Hu et al. (6) developed machine learning models with an accuracy rate of 88.17% to predict positive game reviews based on game-tags. Another study focused on how to improve post-released games (7). Although much research has been done, many lack coverage on smaller games and tend to adopt a narrower view on what is success. In contrast, our investigation takes on a more holistic view on a game's success. In this investigation we account for genre tags as well as pc requirements in addition to other common features (player count, release date, etc.)

Thus we seek to address the following question: *can game metadata available at or before launch predict how a Steam game is rated by the community?* To this end, we formalize two complementary sub-problems:

1) **Multi-class classification**: predict the rating category (*Positive*, *Mixed*, etc.).
2) **Regression**: predict a continuous measure of reception quality, the Wilson lower bound score.

We additionally apply unsupervised **K-Medoids clustering** to discover latent groupings of games that share commercial and design characteristics, independent of their rating labels.

II. Data Understanding and Preprocessing

A. Dataset Sourcing and Integration

The foundational data for this study originate from the **Steam Store API** and **SteamSpy** compiled by Nik Davis for their Steam Store Games Dataset on Kaggle. The Steam Store API provides metadata from Valve, while SteamSpy contributes estimations for ownership and user-applied tags. Davis' dataset consists of six-structured tabular files of which we used the following:

1) **`steam.csv` ($27,075$ rows, 18 columns):** The primary metadata repository containing core attributes such as appid, name, release date, price, raw rating counts, etc.
2) **`steam_requirements_data.csv` ($27,319$ rows, 6 columns):** A collection of hardware specifications including minimum and recommended requirements for PC, Mac, and Linux.
3) **`steamspy_tag_data.csv` ($29,022$ rows, 372 columns):** A dataset consisting of an appid and 371 individual community-voted tags representing genres and mechanics.
4) **`steam_support_info.csv` ($27,136$ rows, 4 columns):** Administrative data including developer websites, support URLs, and contact emails.

B. Data Cleaning and Pre-processing

To prepare the dataset for classification and clustering, we utilized several pre-processing steps to ensure consistency and usability. Firstly, we merged the above datasets using the shared `appid` column. This resulted in a pre-cleaned dataset of around 27,000 rows and 400 columns.

Secondly, we expanded on the categorical metadata found in `steam.csv` by isolating developer-defined categories and genres from the "categories" variables. Each unique genre and category was represented as a binary feature indicating its presence for a given row. This would allow our models to capture the influence of specific game characteristics while preserving the multi-label nature of these attributes.

Additionally, we transformed the HTML string text found in `steam_requirements_data.csv` into usable numerical formats. Using the semi-structured HTML text, we parsed to extract structured variables such as processor speed (GHz), RAM (MB), storage requirements (MB), GPU (MB) and binary indicators for internet connectivity. In order to best capture technical specs for most games, we scraped for "minimum requirements" first over "recommended requirements" as older games and indie games typically only indicated the former. This transformation enabled the inclusion and use of technical specifications as quantitative predictors.

Lastly, due to the highly skewed distribution of several numerical variables, logarithmic transformations were applied to features such as ownership counts and playtime. This was done to reduce the impact of extreme values and improve model stability as a few outliers had tens of millions of owners while others had disproportionately higher playtime.

C. Final Dataset

After data pre-processing, our final dataset covered **26,821 games** with **499 raw features** such as: game metadata (price, release date, required hardware specs), developer and publisher identity, 400+ binary Steam tag columns (genre, theme, and mechanic indicators), log-transformed owner count (`owners_log`), average and median playtime, and platform support flags. After feature selection, **40 features** were retained for modeling (38 after remove highly correlated features).

D. Target Variables

1) Classification Target: Steam games accumulate reviews aggregated into five sentiment labels: *Negative*, *Mostly Negative*, *Mixed*, *Mostly Positive*, and *Positive*. Because adjacent categories are inherently fuzzy (54% vs. 56% positive share the same boundary), we evaluate both the native **5-class** formulation and a collapsed **3-class** formulation:

$$\text{Bad} = \{\text{Neg.},\ \text{M.Neg.}\},\quad \text{Mixed} = \{\text{Mixed}\},$$

$$\text{Good} = \{\text{M.Pos.},\ \text{Pos.}\}$$

The class distributions for both formulations are shown in Fig. 1. These were obtained by discretizations of the continuous Wilson score W, roughly based on the quantiles in our dataset. Collapsing to three classes yields a more balanced distribution (Bad 33%, Mixed 29%, Good 37%), reducing the class-imbalance penalty during training.

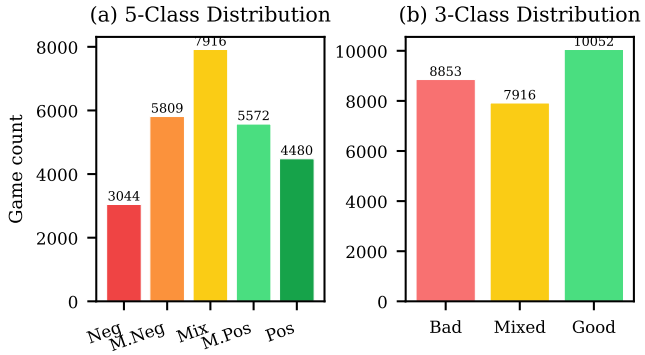

Fig. 1: Rating class distributions for the full dataset (N=26,821). The 5-class distribution (left) is moderately imbalanced; the 3-class collapse (right) yields a near-uniform split.

2) Regression Target — Wilson Score: Rather than predicting the raw positive-review percentage, we introduce the **Wilson lower bound score** as our regression target. For observed proportion $\hat{p}$ over n reviews at $z = 1.96$:

$$W = \frac{\hat{p} + \frac{z^2}{2n} - z\sqrt{\frac{\hat{p}(1-\hat{p})}{n} + \frac{z^2}{4n^2}}}{1 + \frac{z^2}{n}} \tag{1}$$

This metric as apt as it penalizes games with few reviews even if all are positive, providing a statistically principled measure of confidence in a game's reception. In our dataset $W \in [0.00,\ 0.99]$ with $\mu = 0.531$ and $\sigma = 0.254$ (Fig. 2).

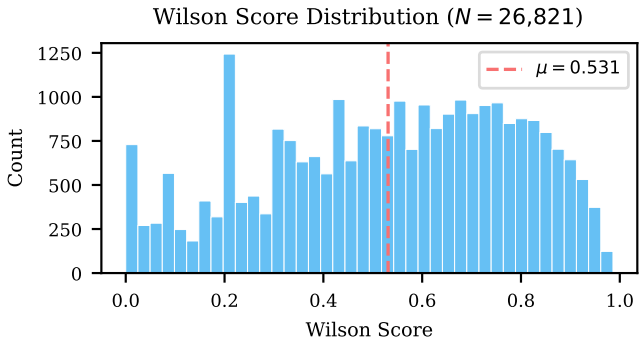

Fig. 2: Distribution of Wilson scores across all 26,821 games. The dashed red line marks the dataset mean ($\mu = 0.531$). The near-symmetric spread confirms that Wilson scoring produces a well-behaved regression target without the extreme spikes at 0 and 1 seen in raw review ratios.

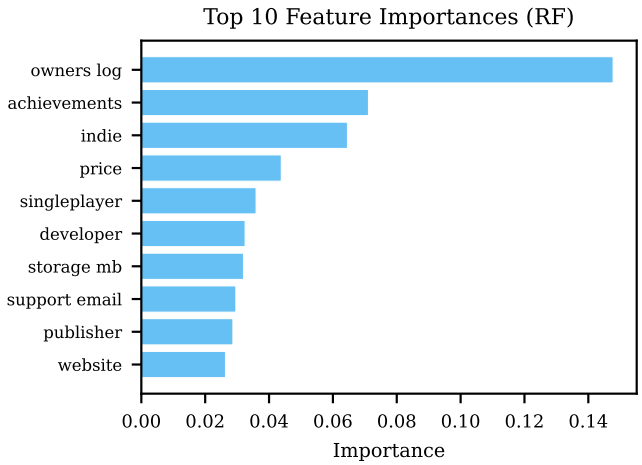

Fig. 3: Top 10 features by Random Forest importance. Log-transformed owner count is the single strongest predictor. Beyond ownership, game design choices (achievements, indie tag, singleplayer tag) and price are the most informative pre-launch signals.

E. Feature Selection

Feature selection proceeded in two stages:

1) **RF importance threshold.** A Random Forest was trained on all 499 features; those with importance < 0.003 were discarded. Two playtime columns were added manually, yielding 40 features. Note that importance was calculated as reduction in Gini impurity (1):

$$G = \sum_{i=1}^{C} p_i(1 - p_i)$$

2) **Correlation pruning.** For any feature pair with Pearson $|r| \geq 0.90$, the lower-importance feature was removed, leaving **38 features** for evaluation model training. Naturally, including highly correlated pairs would violate the avoiding multicollinearity assumption that governs many of our prediction techniques (2). The correlation matrix obtained is provided in the appendix for completeness.

Fig. 3 shows the top 10 features by importance. `owners_log` dominates by a wide margin (0.148), followed by `achievements` (0.071) and the `indie` tag (0.065).

F. Developer Target Encoding

The `developer` column is a high-cardinality categorical variable (thousands of distinct values). One-hot encoding would generate thousands of sparse columns; plain integer encoding implies a false ordinal relationship (3). We apply **within-fold target encoding**: in each CV fold, each developer is replaced by the mean target value of their games computed exclusively on the training portion, preventing label leakage. This is implemented as a custom `TargetEncoderCV` scikit-learn transformer at the head of every pipeline.

III. Clustering

A. Methodology

K-medoids clustering was applied to discover natural groupings of games within the dataset without reference to rating labels. K-Medoids was preferred over traditional K-Means because the feature space is mixed-type (numeric, binary, and categorical), necessitating the use of the Gower distance metric. The Gower distance is defined and bounded in the range $[0, 1]$, allowing for a mathematically sound distance calculation across heterogeneous data. Unlike K-Means centroids, which represent synthetic averages, K-Medoids cluster representatives are actual data points from the set.

Due to the O(n²) memory cost associated with constructing a Gower distance matrix, initial clustering was performed on a stratified sample of 5,000 games to preserve the class proportions of the rating categories. Once the algorithm was refined, the process was scaled to the full dataset. Our analysis consisted of three distinct clustering iterations, utilizing progressive dimensionality reduction to isolate signal from noise:

1) High-Dimensional Baseline: Included all 135 features (120 general features and the top 15 most frequent tags).
2) Refined Feature Set: Restricted to the top 10 features most indicative of rating categories.
3) Targeted Feature Set: Restricted further to the top 5 most important features.

To assess the clustering tendency, we implemented a modified Hopkins statistic adjusted for Gower distances:

$$H_{Gower} = \frac{\sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i},$$

where x_i is the Gower distance from a point in a real data sample D_s to its nearest neighbor in D, and y_i is the Gower distance from a point in a randomly generated uniform sample R to its nearest neighbor in D_s.

This statistic measures how close one point is to another versus how far a random point is to either, indicating whether clustering tendency exists within the dataset on a scale of $[0.5, 1.0]$.

The optimal number of clusters k was determined using the Elbow Method: plotting the Within-Cluster Sum of Pairwise

Distances (WCPD) and identifying the "elbow" where marginal gains in variance explanation diminish.

B. Results and Challenges

Using the full feature set resulted in poorly defined clusters. The Silhouette Score peaked at 0.17, and the ARI was 0.0055, indicating that cluster membership was essentially random relative to the game's actual rating. No distinct patterns emerged; the distribution of rating categories in clusters followed the true distribution of the dataset closely.

Iteration 2: Top 10 Features

Dimensionality reduction provided immediate improvements. The Silhouette Score rose to 0.3284 at $k = 4$, and the Hopkins score of 0.8053 suggested a much stronger clustering tendency. While the ARI remained low (0.0213), one specific cluster began to show an overrepresentation of positively rated games, providing the first sign of meaningful structural alignment.

Optimal Performance: Top 5 Features

The best results were achieved by limiting the Gower distance calculation to only the top 5 features. The Silhouette score rose to 0.7626, indicating much stronger, well-separated clusters. The Hopkins score of 0.9076 confirmed an even higher clustering tendency. Cluster 2 and Cluster 4 showed significant overrepresentation of positive and mostly positive categories, successfully isolating high-quality titles. Unlike previous attempts, this iteration produced highly varied sizes, with the largest containing 18,000 games and the smallest containing only 133.

From a data perspective, we found that the majority of columns in the Steam dataset act as noise when attempting to predict user reception through unsupervised means. The resulting clusters primarily separated games along the commercial scale axis, distinguishing large-audience commercial titles from niche games. While we successfully identified structural groupings, the Adjusted Rand Index (ARI) remained near zero across all tests. This leads to a critical conclusion: unsupervised methods like K-Medoids are excellent for identifying structural groupings but are not the optimal tool for recognizing patterns that help predict specific reception scores like the Wilson score. Success prediction likely requires a signal not recoverable by unsupervised methods on this feature set alone.

IV. Methodology

A. Evaluation Framework

All models were embedded in scikit-learn `Pipeline` objects applying identical preprocessing across folds (target encoding, optional standard scaling). We used:

- **Stratified 5-Fold CV** for classifiers (preserving class proportions, critical for the imbalanced 5-class setting).
- **5-Fold CV** for regressors.
- An **80/20 stratified train/test split**: CV ran on the 80% training partition; test metrics are unbiased generalization estimates.
- **`class_weight='balanced'`** on all classifiers to prevent majority-class dominance.

B. Classification Models

Five classifiers were evaluated on both the 3-class and 5-class problems (Table I).

TABLE I: Classifier configurations.

Model	Key configuration
Naive Bayes	Gaussian; feature-scaled; balanced.
Logistic Reg.	`max_iter`=1000; scaled; balanced.
Random Forest	200 trees; balanced; no scaling.
XGBoost	500 trees; η=0.05; depth 6; `min_child_wt`=3.
KNN	k=10; feature-scaled.

Naive Bayes provides an important baseline for our classification. Further hyperparameter tuning is beyond the scope of our research, but these were determined based on well established heuristics and software defaults known to provide stable performance across a diverse dataset (4).

C. Regression Models

Three regression models were evaluated to predict the continuous Wilson score (Table II).

TABLE II: Regression model configurations.

Model	Key configuration
Linear Regression	Baseline; feature-scaled.
Random Forest	200 trees.
XGBoost	500 trees; η=0.05; depth 6.

V. Evaluation of Results

A. Classification — 3-Class

Table III reports 3-class results. XGBoost achieves the best test accuracy of 69.4% with weighted F1 of 0.680. The narrow gap between CV and test accuracy across all models confirms stable, non-overfit results. A random baseline on this near-balanced 3-class problem scores ≈33% (33% = 1/number of classes, in this case 3), so all models substantially outperform chance. Logistic regression also had the weakest performance at 56.1% test accuracy and a 56.4% average cross-validation accuracy.. The nonlinear models Random Forest and XGBoost outperformed the linear model (logistic regression). A reason for this is because the non-linear models were better able to capture a more complex relationship between selected features that a linear hyperplane can not express. On the other hand KNN, another nonlinear model, performed relatively poorly similar to to logistic regression; KNN treats features uniformly which is a caveat potentially as it can oversimplify relationships between features disregarding the possibility that features are not all equally important. While important features were found and used instead of the entire dataset additional noise and the curse of dimensionality could have affected results leading to a subpar performance.

Fig. 4(a) visualizes test accuracy and F1 across models. Misclassifications concentrate at class boundaries: *Bad* games are most often misclassified as *Mixed*, and *Good* as *Mixed*, reflecting genuine ambiguity in mid-range games. ROC AUC

TABLE III: 3-Class classification results (5-fold stratified CV + hold-out test).

Model	CV Acc	CV F1	Test Acc	Test F1
Logistic Reg.	56.4±1.7%	0.592	56.1%	0.591
Random Forest	69.0±0.9%	0.653	68.8%	0.650
XGBoost	**68.7±0.6%**	**0.670**	**69.4%**	**0.680**
KNN	61.6±0.6%	0.584	62.3%	0.590

values are well above 0.5 for all three classes, with *Good* and *Bad* being more discernible than *Mixed*.

Table IV displays the binarized AUC scores of each model in tabular form. The higher scores for the *Good* and *Bad* AUC scores (up to 0.87) suggest that the metadata contains stronger signals for opposite ends of the reception spectrum. In contrast, the lower AUC scores were for *mixed* games. This finding is consistent with the 5-class model as well. Mixed classes are middle range which indicates an overlap such that there may not be defining characteristics or metadata of mixed games. Despite a slightly lower AUC score for mixed games, the lowest score of 0.62 still is higher than an AUC score of 0.50 from a random guessing classifier. Thus all of the models were significantly better at identifying what games were *Good*, *Bad*, or *Mixed* than a random guesser classifier.

TABLE IV: 3-Class classification binarized ROC AUC scores

Model	Bad AUC	Good AUC	Mixed AUC
Logistic Reg.	0.82	0.80	0.63
Random Forest	0.86	0.85	0.73
XGBoost	**0.87**	**0.85**	**0.74**
KNN	0.72	0.72	0.62

B. Classification — 5-Class

Table V reports 5-class results. XGBoost again leads at 50.8% test accuracy, but the overall performance is considerably lower than the 3-class setting. A random baseline achieves ≈20%; the 17 percentage point gap to the 3-class best demonstrates that *Mostly Positive/Positive* and *Mostly Negative/Negative* distinctions are not meaningfully recoverable from metadata alone. Differentiating mostly positive vs. positive becomes increasingly difficult compared to the 3-class classification. The lowest performing model: Naive Bayes performs similarly to random guessing and has the lowest Test F1 score of 0.202. These results were somewhat expected as the Naive Bayes classifier relies on the assumption that all features are not related to each other and thus are independent. From earlier data cleaning, features were found to be correlated with each other. For the other models the order of general model performance was the same as the 3-class classification. While test and CV accuracy was substantially lower than its 3-class counterpart XGBoost and Random Forest still performed better due to its ability in handling non-linear relationships between features. Most models did better than a random 5-class classifier, though.

C. Regression — Wilson Score

Table VI reports regression results. Based on the test data, XGBoost explains approximately 31% of the variance

TABLE V: 5-Class classification results.

Model	CV Acc	Test Acc	Test F1
Naive Bayes	17.8±1.2%	15.9%	0.202
Logistic Reg.	38.2±1.6%	37.9%	0.413
Random Forest	47.7±1.5%	50.8%	0.497
XGBoost	**49.4±1.2%**	**50.8%**	**0.499**
KNN	38.7±0.8%	36.7%	0.366

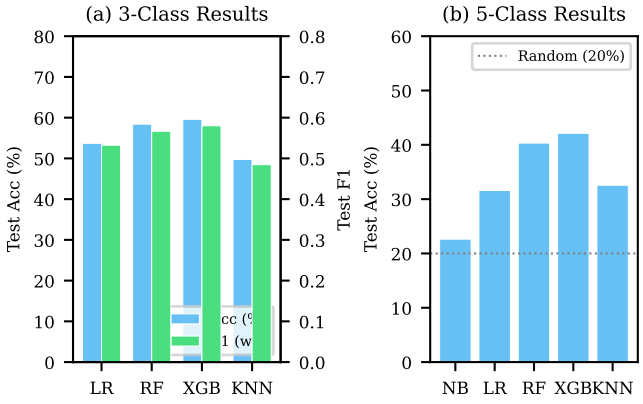

Fig. 4: Test accuracy and weighted F1 for the 3-class problem (a) and test accuracy for the 5-class problem (b). The dashed line in (b) marks the 20% random-baseline. XGBoost leads in both settings; the accuracy drop from 3-class to 5-class confirms that adjacent Steam labels are not separable from metadata.

(R^2=0.31, RMSE = 0.167 on a target in $[0, 1]$). The large gap between Linear Regression ($R^2 = 0.21$) and tree-based models ($R^2 \approx 0.24 - 0.31$) evidences important non-linear relationships, particularly between `owners_log` and Wilson score (Fig. 6). It is important to note, however, that even with a relatively modest R^2, linear regression obtained an F score of 13.02 which is significantly greater than 1. Under a significant level of $\alpha = 0.05$, this yielded a p-value of $3.28 \times 10^{-69} << \alpha$. Thus, we rejected the null hypothesis H_0 and concluded that even the 40 most important predictors chosen by the RF importance threshold (10) explain a significant amount of variability in Wilson scores.

TABLE VI: Regression results predicting continuous Wilson score.

Model	CV R^2	CV RMSE	Test R^2	Test RMSE
Linear Reg.	0.193±0.012	0.181	0.211	0.178
Random Forest	0.221±0.017	0.178	0.242	0.175
XGBoost	**0.282±0.015**	**0.171**	**0.304**	**0.167**

Fig. 5 shows the XGBoost residual plot. Residuals are approximately centered at zero, with mild heteroskedasticity: the model slightly underestimates very high Wilson scores ($W > 0.85$), corresponding to games with extremely large, uniformly positive review counts that are difficult to predict from metadata alone.

One also observes the variance of the residuals does not remain constant across different values, and this violation of homoscedasticity (which violates a regression assumption) does not bode well.

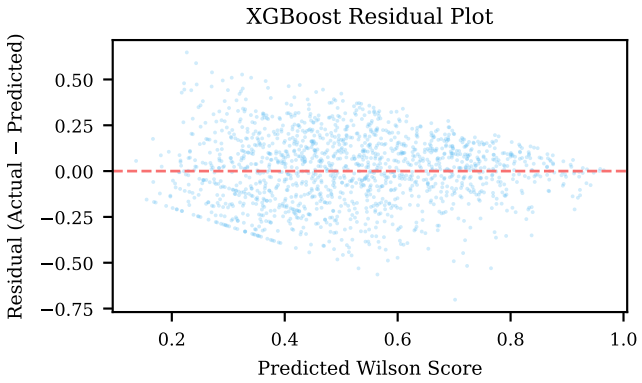

Fig. 5: XGBoost residuals vs. predicted Wilson score (1,500-point held-out sample). Residuals are centered near zero across the prediction range, with slight underestimation at high scores ($W > 0.85$) where very popular, well-reviewed games cluster.

Fig. 6: Scatter plot of log-estimated owner count vs. Wilson score for a random sample of 2,000 games, colored by rating category.

VI. The Wilson Score

The Wilson lower bound score (Eq. 1) is the central innovation in our target variable design. Standard Steam-scraped datasets use the raw positive-review percentage as the label, which is statistically unstable for games with few reviews — a title with three reviews, all positive, records 100% approval but with negligible confidence. The Wilson score corrects this by providing a lower confidence bound on the true approval rate, penalizing sparsely-reviewed games.

Commercial alignment. The Wilson score priorities games that are both well-reviewed *and* widely reviewed: precisely the definition of commercial success. It is also better-behaved statistically: the histogram in Fig. 2 shows fewer extreme values near 0 and 1 compared to a raw-ratio target, yielding a more uniform regression target.

Owners–score relationship. Fig. 6 illustrates the relationship between log-transformed owner count and Wilson score. Games with few owners span the full score range, while games with many owners converge toward scores of 0.6–0.9. This reflects survivorship: games attracting large audiences do so partly because they are genuinely good. The relationship is non-linear and varies by game type (free-to-play titles naturally accumulate owners more easily), which explains why linear regression performs substantially worse than tree-based models.

As the figure shows, games with higher owner counts converge toward mid-to-high Wilson scores, but dispersion is large at low counts — unsurprisingly, games succeed or fail independent of scale at the low-ownership end.

We also note that our data is left-skewed with respect to the positive ratings ratio, which is given by

$$P = \frac{\text{pos\_ratings}}{\text{pos\_ratings+neg\_ratings}}$$

This would hold if our sample itself were approximately normal, the observed sample distribution of P is heavily right-skewed. Thus, it is more akin to the blue curve in Fig. 7 rather than the pink one. Even if one posits a notional symmetric population centered, the *observed* positive-ratio distribution does not resemble it. Consequently, a naive normal-approximation confidence interval $\hat{p} \pm z\sqrt{\hat{p}(1-\hat{p})/n}$ can yield bounds outside $[0, 1]$ for extreme $\hat{p}$, and is unreliable for games with small n. The Wilson interval corrects precisely for this: producing asymmetric bounds $[w^-, w^+]$ that remain within $[0, 1]$ and shrink appropriately when review counts are low, regardless of the shape of the empirical P distribution. This asymmetry is visible in the figure — the gap $w^+ - \hat{p}$ is wider than $\hat{p} - w^-$, reflecting the skewed uncertainty around a high observed proportion. Thus, it proves to be a theoretically reasonable choice in our case.

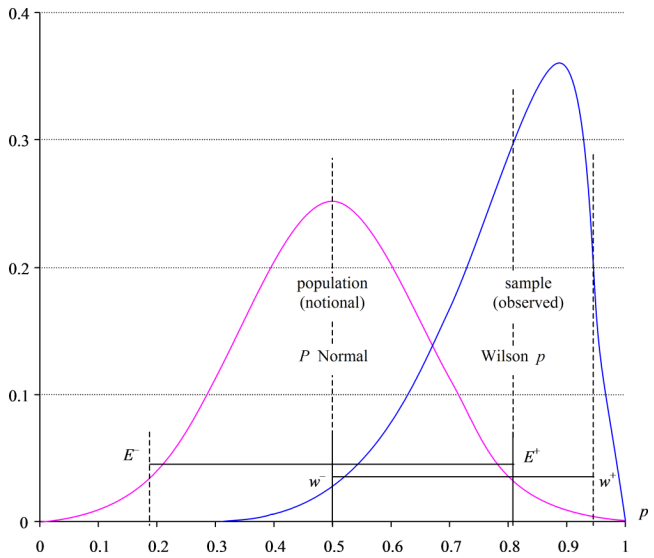

Fig. 7: Normal Vs. Wilson distributions. (a) The pink curve assumes $p \sim N(\mu, \sigma^2)$ (b) The blue curve represents our sample distribution, which is left skewed. The Wilson interval $[w^-, w^+]$ more accurately captures the true value of p as compared to the normal confidence interval $[E^-, E^+]$. This image was taken from **??**.

VII. Interpretation and Discussion

Accuracy ceiling at ∼69%. Game success is fundamentally influenced by factors absent from metadata: be it factors like

marketing spend, Twitch/YouTube exposure, review bombing, seasonal timing, as well as the subjective quality of gameplay. The available features explain a meaningful but incomplete share of variance; 69% represents a realistic ceiling for this feature set.

Most predictive features. `owners_log` dominates all models (importance 0.148), creating a mild circularity (popular games tend to be well-reviewed) that is nonetheless genuinely informative for regression. Excluding ownership, `achievements` (0.071), the `indie` tag (0.065), and `price` (0.044) are the next strongest signals, suggesting that measurable game design choices correlate with community reception.

3-class vs. 5-class. The 17-point accuracy gap between the best 5-class (50.8%) and 3-class (69.4%) test accuracies demonstrates that the five Steam labels do not represent equally meaningful distinctions in the metadata space. *Mostly Positive* vs. *Positive* is driven by review volume thresholds instead of fundamentally different game characteristics.

Clustering vs. classification. Unsupervised groupings primarily reflect commercial scale and price tier rather than rating quality, confirming that rating prediction indeed requires the supervised label signal.

VIII. Conclusion

We demonstrated that Steam game success, formally quantified by rating category and Wilson score, can be predicted to a meaningful degree from game metadata. XGBoost consistently outperformed all other models, achieving **69.4% 3-class test accuracy** and $\mathbf{R^2 = 0.31}$ for Wilson score regression with cross-validated stability. Our key findings are

1) Owner count is the dominant predictor (with the highest importance obtained from Random Forest Regression).
2) Tree models capture non-linearities that linear regression misses.
3) Collapsing to three rating classes is necessary for meaningful classification — adjacent Steam labels are not separable from metadata.
4) The Wilson score is a statistically superior regression target to raw review percentage.
5) K-Medoids clustering recovers commercial-scale tiers, not quality tiers, confirming the need for supervised signal.
6) A public **Streamlit application** provides interactive EDA, model evaluation, single-game prediction, and batch inference.

Therefore, our analysis shows that game reception is hard to predict purely from metadata alone. Indeed, the inherent subjectivity of player taste likely caps how far any feature-based model can go.

IX. Future work

Future directions include more advanced feature engineering and model selection, as well as analysis of social-media-mention signals at launch. Having a larger dataset with more balanced classes in terms of prices and other key features would be valuable to our analysis as well.

X. References

1) L. Breiman, "Random forests machine learning," SpringerLink, https://link.springer.com/article/10.1023/A:1010933404324 (accessed Apr. 2026).
2) Farrar, Donald E., and Robert R. Glauber. "Multicollinearity in Regression Analysis: The Problem Revisited." The Review of Economics and Statistics, vol. 49, no. 1, 1967, pp. 92–107. JSTOR, https://doi.org/10.2307/1937887. (accessed 23 Apr. 2026).
3) C. Guo and F. Berkhahn, "Entity embeddings of categorical variables," arXiv.org, https://doi.org/10.48550/arXiv.1604.06737 (accessed Apr. 2026).
4) F. Pedregosa, G. Varoquaux, and Al., "Scikit-Learn: Machine learning in Python," arXiv.org, https://doi.org/10.48550/arXiv.1201.0490 (accessed Apr. 2026).
5) De Luisa, A., Hartman, J., Nabergoj, D., Pahor, S., Rus, M., Stevanoski, B., Demsar, J., and Strumbelj, E. "Predicting the Popularity of Games on Steam," University of Ljubljana, 2021. https://ev.fe.uni-lj.si/4-2021/Luisa.pdf (accessed Apr. 2026).
6) Hu, W., Wang, Y., and Xia, R. "Machine Learning-Based Steam Platform Game's Popularity Analysis," SCITEPRESS, 2024. https://www.scitepress.org/Papers/2024/128538/128538.pdf (accessed Apr. 2026).
7) Lin, D., Bezemer, C.-P., Zou, Y., and Hassan, A. E. "An Empirical Study of Game Reviews on the Steam Platform," Empirical Software Engineering, 24(1):170–207, 2019. https://sealqueensu.github.io/publications/pdf/EMSE-Dayi-2019.pdf (acessed APr. 2026)
8) "Steam game user statistics 2026," *Icon Era*, Dec. 13, 2025. https://icon-era.com/statistics/steam-game-statistics/. (acessed APr. 2026)
9) F. Zhu and X. Zhang, "Impact of online consumer reviews on sales: The moderating role of product and consumer characteristics," *Journal of Marketing*, vol. 74, no. 2, pp. 133–148, Mar. 2010, doi: 10.1509/jm.74.2.133. (acessed APr. 2026)
10) Y. N. D. of Biostatistics, "Random Forest Variable Importance-based Selection Algorithm in Class Imbalance Problem", https://arxiv.org/html/2312.10573v1 (accessed Apr. 2026).
11) S. Wallis, "Binomial distributions," *Corpus Linguistics Blog*, Mar. 31, 2012. https://corplingstats.wordpress.com/2012/03/31/binomial-distributions/ (accessed Apr. 2026).

XI. Appendix

Code: https://github.com/LuminousFirefly/game_success_prediction

Streamlit website: https://steam-game-success-prediction.streamlit.app