Final Project Report

Predicting the Success of an Indie Game

Authors:
Isaac Kinane
Samy Mancer

Supervisor:
Yoann Poupart

Master MIND – Machine Learning, Artificial Intelligence and Data

Academic Year 2025–2026

Sorbonne University
Department of Computer Science

Contents

1 Introduction and Objectives

In accordance with the DALAS course methodological framework, this project was conducted following the standard stages of the Data Science process: problem framing, data acquisition, data preparation and exploration (EDA), modeling, and the visualization and interpretation of results.

1.1 Context: The Indiepocalypse and Market Saturation

The video game industry has evolved from an oligopoly dominated by large studios into a highly fragmented market. In 2024, indie games accounted for nearly **99% of Steam releases**, with more than **14,000 new titles** published annually. This oversupply, often referred to as the *"Indiepocalypse"*, has made visibility the scarcest resource for independent developers. Our analysis focuses specifically on Steam, as the platform holds a near-monopoly over the independent gaming market.

Paradoxically, the indie market now generates **substantial revenues**, contributing approximately **48% of Steam's total sales revenue** in 2024. However, this growth is largely concentrated among so-called *Triple-I* studios, leaving small independent teams competing for a limited share of the market. While landmark successes such as *Hollow Knight* or *Elden Ring* demonstrate the potential of indie productions, commercial failure remains the norm. This project therefore aims to **objectively identify the determinants of success** through a *data-driven* approach.

1.2 Objectives and Target Definition (SMART Framework)

Our research problem is as follows:

> *Can we predict the commercial performance of a game even before its full launch, based on its metadata and semantic positioning?*

- **Specific:** The scope is restricted to paid (*premium*) *indie* games on Steam, excluding Free-to-Play titles (due to their distinct business model).
- **Measurable:** Lacking public sales data, we construct a robust proxy based on the literature (*Boxleiter method*):

$$\text{Estimated Sales} \approx \text{Review Count} \times \alpha, \quad \text{where } \alpha \in [30, 50] \tag{1}$$

  We define two target variables:

  - **Regression ($Y_{\log}$):** $\log(1 + \text{Reviews})$.
  - **Classification (Y_{class}):** Binary Success (Reviews > 500 **and** Positive Ratio $> 85\%$).

- **Achievable:** Via the creation of a dataset (scraping).
- **Realistic:** Using state-of-the-art NLP (Transformers) and tabular learning.
- **Time-bound:** Analysis of post-Covid trends (2020–2025).

2 Acquisition Architecture & Robustness

2.1 Ingestion Strategy: Asynchronous Concurrency

To compile a dataset of over 80,000 games, we designed a high-performance architecture for *IO-bound* ingestion based on `asyncio` and `aiohttp`. Unlike a blocking sequential approach, our `SteamScraper.py` module maximizes throughput via fine-grained parallelization:

- **Simultaneous Triple-Fetch:** For each unique identifier (`app_id`), we simultaneously trigger three asynchronous calls (`asyncio.gather`) targeting the Details API, the Reviews API, and the Store HTML page respectively. This approach reduces network latency threefold per processed item.
- **Batch Orchestration:** Processing is performed in configurable batches (here `chunk_size=100`), ensuring optimal save granularity and error recovery without data loss.

2.2 Dynamic Throughput Management

The major challenge in massive data acquisition lies in responsible traffic management to avoid server overload and respect the platform's access policies (HTTP 429 and 403 codes). Rather than relying on inefficient static delays, we developed an advanced rate-limiting mechanism within the **SteamScraper** tool. This mechanism is based on an **AIMD (Additive Increase Multiplicative Decrease)** adaptive control algorithm, inspired by network protocol congestion control strategies.

This system regulates collection speed in real-time via a Finite State Machine (FSM) with three regimes:

1. **OPTIMIZING:** The system progressively increases throughput (delay reduction, controlled increase in concurrency) as long as the error rate over the sliding window (`history_size=100`) remains below the critical threshold of 7.5%.
2. **THROTTLED:** Upon detection of a server constraint, concurrency is multiplicatively reduced (factor 0.9) to stabilize the load and return below tolerance thresholds.
3. **RECOVERING:** In the event of temporary access denial, the script voluntarily pauses before resuming at a minimal safe rate.

It is important to emphasize that this approach in no way aims to bypass protection mechanisms, but rather to **adapt to them cooperatively**. Steam's `robots.txt` file was explicitly consulted prior to the project, and only unrestricted routes were used. The *SteamScraper* thus acts as a **self-throttling** mechanism, ensuring behavior that respects server infrastructure and complies with web scraping best practices.

This strategy maximizes collection efficiency while maintaining a stable and sustainable throughput. Thanks to this adaptive approach, we were able to reduce the total scraping time from approximately 130 hours to 45 hours, whereas a fixed-rate scraper would have either underperformed or incurred a high risk of premature blocking.

2.3 Data Quality: "Schema-on-Write"

To ensure data integrity prior to modeling (the *Garbage In, Garbage Out* principle), we apply strict validation at the point of ingestion via the **JSON Schema** standard. The `schema.json` file acts as an inviolable *Data Contract*:

- **Strong Typing and Constraints:** Immediate type conversion (e.g., `app_id` to integers) and rejection of incomplete data (e.g., missing user reviews).
- **Stream Segregation:** Invalid payloads are automatically routed to an audit file (`steam_games_errors.jsonl`), ensuring the main dataset is "Model-Ready" without requiring heavy subsequent cleaning.

2.4 Data Processing Flow

The process is orchestrated by a series of scripts with clearly defined responsibilities, as illustrated below.

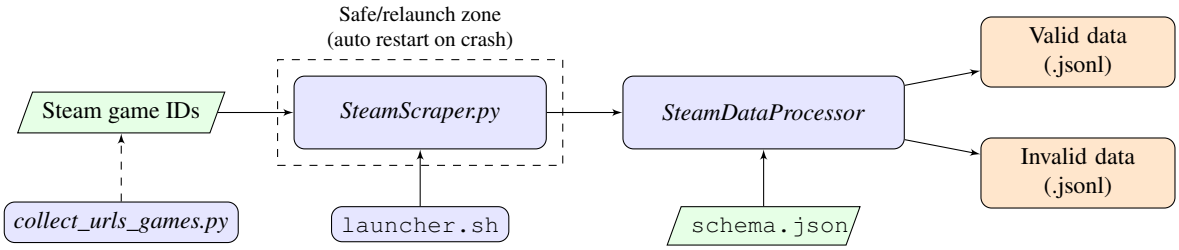


Figure 1: Complete pipeline for collecting and processing Steam data.

3 Exploratory Data Analysis (EDA)

An **Exploratory Data Analysis (EDA)** phase was conducted prior to any modeling steps. The objective of this phase is to **understand the dataset structure**, identify imbalances, trends, and potential sources of bias, and to **motivate the preprocessing, feature engineering, and modeling choices** presented subsequently.

3.1 Distribution of Review Counts

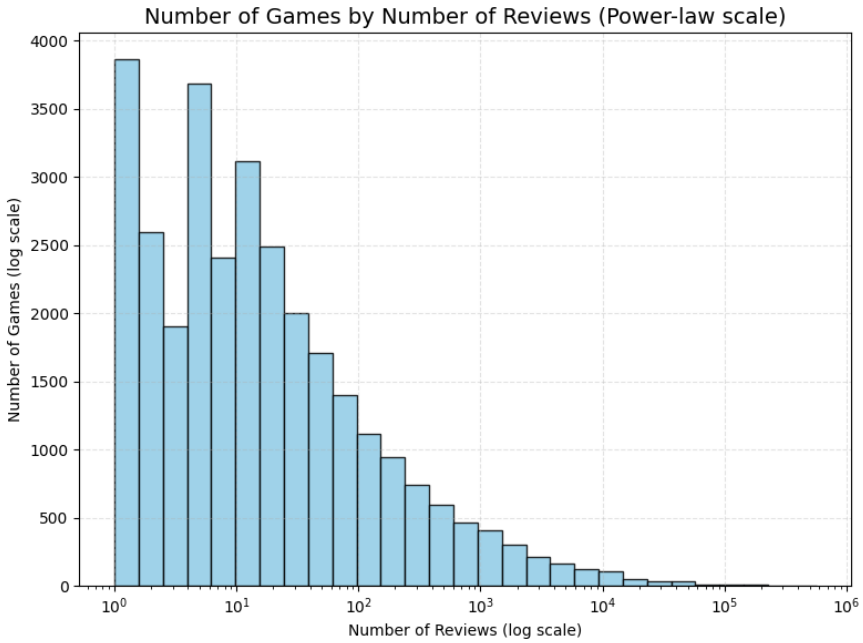


Figure 2: Distribution of review counts (logarithmic scale).

The distribution of review counts exhibits **pronounced skewness**, characteristic of a **power law**. A minority of games accounts for a very significant share of reviews, while the vast majority of titles do not exceed a few dozen, or at most a few hundred, reviews.

This observation highlights a highly unequal market, dominated by a few major hits, and justifies the use of a **logarithmic transformation of the target variable** to stabilize variance and facilitate model training.

3.2 Relationship between Price and Review Counts

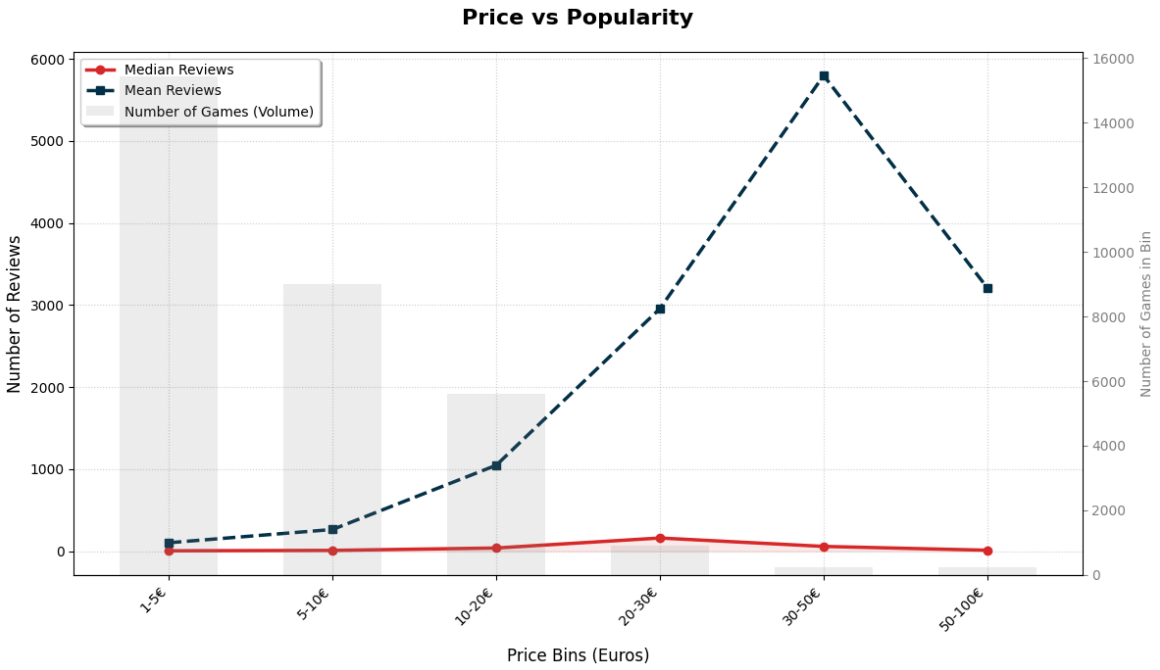


Figure 3: Relationship between price and review counts (logarithmic scale).

The exploratory analysis reveals a **high concentration of games in the 1–5 euro price range**, as well as a **long-tail** distribution for higher prices. Across all price ranges, the median number of reviews remains relatively low, indicating that price, taken in isolation, is not a sufficient determinant of commercial success.

However, we observe an increase in the average number of reviews up to the 30–50 euro bracket, followed by a decline. This trend can be explained by several factors, notably:

- the fact that the most ambitious games require more resources, leading to higher selling prices;
- a **survivorship bias**, as games that met with initial success are more likely to increase their price over time to reach market standards over the 2020–2025 period.

These elements suggest that price should be interpreted in interaction with other variables (categories, platforms, languages, etc.), rather than as a standalone explanatory factor.

3.3 Game Platform Distribution

Next, we analyze the distribution of games according to the **targeted platforms** (Windows, macOS, Linux).

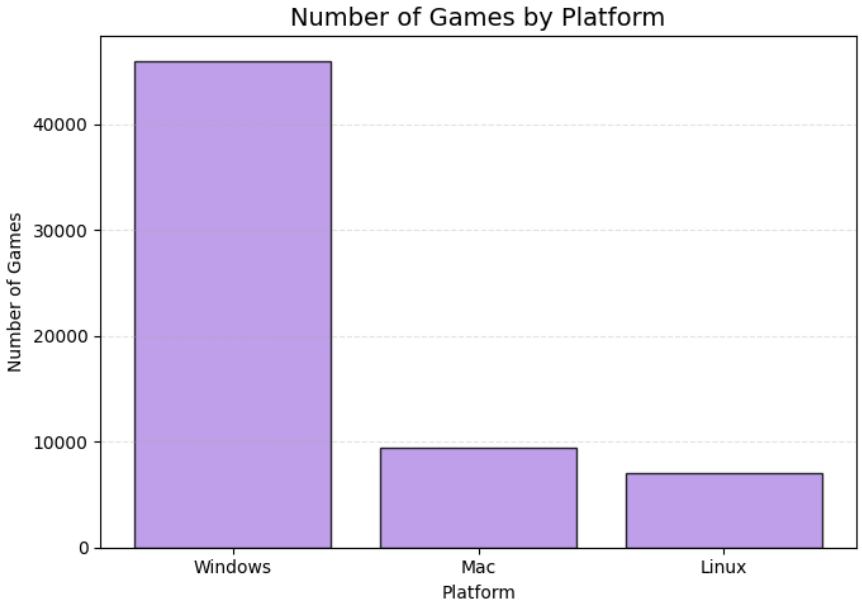


Figure 4: Distribution of games by platform.

The analysis highlights a **very clear dominance of the Windows platform**, while macOS and Linux are significantly less represented. This asymmetry reflects the historical structure of the Steam market and introduces a **structural imbalance** in the *platform* variable.

These observations suggest that:

- simple presence on Windows provides little discriminatory power, given the strong competition;
- on the other hand, multi-platform support may serve as an **indirect indicator of technical maturity or investment**.

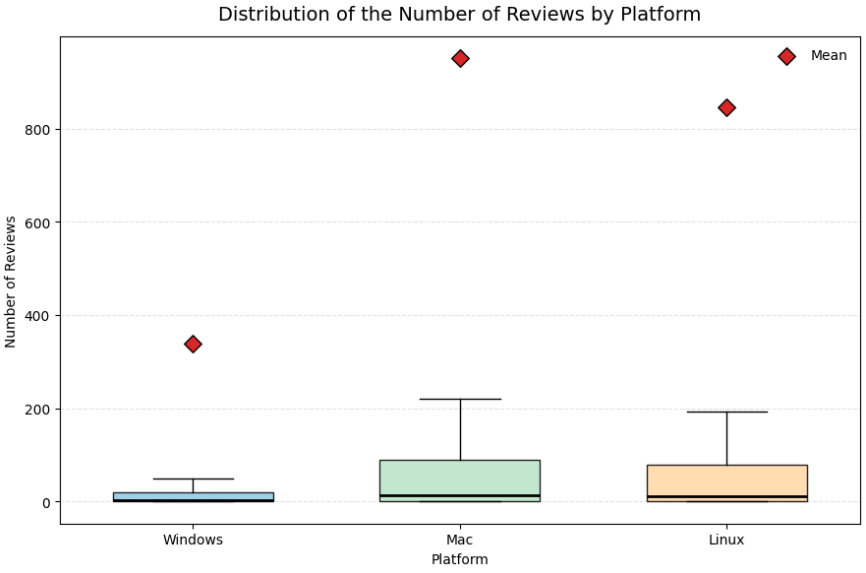


Figure 5: Distribution of review counts by platform.

The boxplot reveals that Windows games show, on average, a lower number of reviews than those available on macOS or Linux, mainly due to the **very high competitive density** on Windows. The

macOS and Linux platforms, although more niche, seem to benefit from a more concentrated and potentially more engaged audience.

To validate these observations, a **Tukey HSD post-hoc test** was applied. The results indicate that the differences between Windows and the other two platforms are statistically significant, while the distributions of macOS and Linux do not differ significantly from each other.

These findings motivated a **specific processing of platform variables** during the feature engineering phase.

3.4 EDA Summary and Methodological Implications

This exploratory analysis phase highlights:

- the existence of **structural imbalances** in several key variables;
- the necessity of **tailored transformations** to mitigate biases related to cardinality or the dominance of certain categories;
- the importance of a **multifactorial analysis**, as variables taken in isolation rarely explain commercial success.

The insights derived from this EDA directly guided the **data preprocessing**, **dimensionality reduction**, and **feature engineering** choices detailed in the following sections.

4 Feature Engineering and Dimensionality Reduction

Feature Engineering formed the cornerstone of our approach, necessitated by the heterogeneity and high dimensionality of the data. The coexistence of unstructured data (textual descriptions), semi-structured data (lists of tags, genres), and structured data (numerical metrics) required a dual strategy: specific data adaptation for "Baseline" architectures (linear regression, MLP) and the exploitation of the CatBoost model's native capabilities.

4.1 NLP: Semantic Vectorization via Transformer

Unlike frequency-based approaches (TF-IDF), which are limited by their contextual blindness and high redundancy with explicit user tags, we deploy a Dense Embedding strategy to capture the deep and implicit semantics of the descriptions.

- **SOTA Model:** Use of `e5-base-v2` (Embeddings from Encoders), a pre-trained Transformer specifically optimized for asymmetric semantic similarity tasks.
- **Inference and Preprocessing:** Each raw description d_i is normalized and prefixed with the technical token `"passage: "` (required by the model architecture) before being converted into a dense vector $v_i \in \mathbb{R}^{768}$.
- **Compression (PCA):** To mitigate the *Curse of Dimensionality*, we apply Principal Component Analysis (PCA) calibrated to retain 95% of the explained variance. This operation projects the semantic space from 768 to ≈ 400 orthogonal latent dimensions, eliminating residual noise. This 400-dimensional representation is used for *clustering* tasks. For *baseline* models (linear/MLP), a more aggressive reduction is applied by retaining 80% of the explained variance, corresponding to ≈ 200 dimensions.

4.2 Semantic Clustering and Model Selection

The selection of the optimal model was conducted in two distinct phases: an initial screening based on a business metric to filter out incoherent segmentations, followed by statistical validation of the top 5 retained approaches.

1. Business-Oriented Filtering: Creating the "Business Score" (S_{adj})

Given the continuous nature of the embedding space, standard geometric metrics (e.g., *Silhouette Score*) proved ineffective, as they artificially penalize fluid transitions between sub-genres (e.g., *Roguelite* vs. *Roguelike*).

To identify the most economically relevant segmentation, we designed a metric based on *Steam Tag* purity:

- **Specificity (S_g):** Measures the inverse entropy of tags, representing a cluster's capacity to capture a specific thematic niche. To ensure reliability, ubiquitous generic tags (e.g., *"Single-player"*, 71% of the corpus) are excluded.
- **Separation (M_{inter}):** The mean cosine similarity between centroids, serving as a proxy for theme uniqueness.
- **Adjusted Final Score:**

$$S_{adj} = \frac{S_g}{M_{inter}} \times \frac{1}{\sqrt{k}}$$

  This ratio incorporates a complexity penalty (inspired by BIC) to discourage excessive fragmentation and favor operationally viable groupings.

This step allowed us to pre-select the 5 highest-performing models. As these models displayed very similar S_{adj} scores, a finer tie-breaking method was required.

2. Final Selection: Statistical Validation of Popularity

To differentiate between these candidates, we sought to determine which one best captured the market's economic reality. We used *Review Volume* (`review`) as a proxy for sales volume.

Hypotheses:

- H_0 : Popularity is identically distributed across all themes.
- H_1: There exists a significant disparity in popularity across the identified themes.

Test Selection (Bartlett → Kruskal-Wallis): Prior to applying ANOVA, Bartlett's test revealed strong heteroscedasticity (p<0.001) for all models, inherent to the *Power Law* distribution of Steam sales (presence of extreme blockbusters). Since parametric ANOVA is biased in such cases, we opted for the **Kruskal-Wallis H Test** (robust to outliers).

4.2.1 Comparative Results and Decision

The table below benchmarks the 5 finalists. The *H Statistic* serves as the decisive metric: a higher value indicates a superior distinction between "viral" and "niche" games.

Table 1: Robustness Benchmark of the Top 5 Models (Kruskal-Wallis Test)

Clustering Model	Bartlett (p)	Kruskal (H)	Significance (p)	Decision
BERTopic (K-Means $K = 310$)	$< 10^{-3}$	**87.66**	$\mathbf{3.73 \times 10^{-16}}$	**Selected**
K-Means ($K = 175$)	$< 10^{-3}$	70.20	1.34×10^{-12}	Rejected
K-Means ($K = 280$)	$< 10^{-3}$	63.63	2.83×10^{-11}	Rejected
BERTopic (K-Means $K = 135$)	$< 10^{-3}$	60.17	1.40×10^{-10}	Rejected
BERTopic (Default: HDBSCAN)	$< 10^{-3}$	50.71	1.05×10^{-08}	Rejected

Conclusion: Table 1 confirms the superiority of the **BERTopic model combined with K-Means** ($K = 310$). With $H = 87.66$ and an extremely low p-value (3.73×10^{-16}), this model proves to be

the most effective at distinguishing genres with high commercial potential from niche genres, thereby validating our hybrid approach (Semantic + Bayesian Optimization of K).

4.2.2 Visualization and Economic Profiling of Genres

The previous statistical analysis confirmed that textual descriptions significantly differentiate popularity. To translate this mathematical finding into market intelligence, we mapped the distribution of reviews onto our semantic clusters.

This visualization employs three complementary graphs to analyze the structure of the Steam market.

1. Overview of Dominant Themes Figure 6 displays the review distribution for the 8 largest clusters in the corpus.

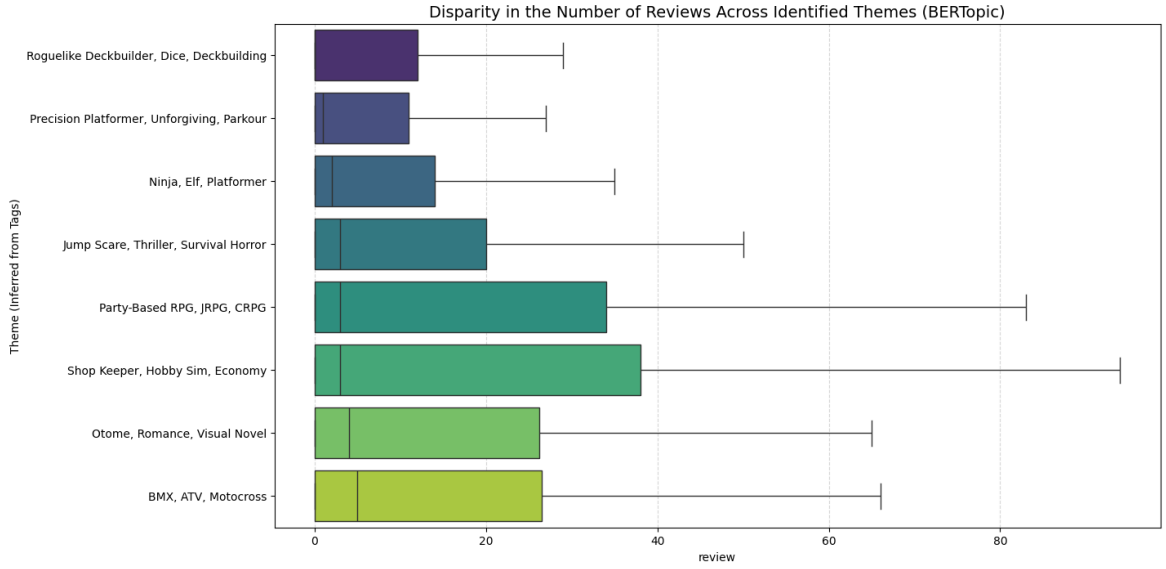


Figure 6: Popularity distribution across the 8 dominant themes. Labels (Y-axis) are automatically generated using the cluster's 3 most frequent tags, validating the semantic interpretation.

We observe strong heterogeneity: certain genres (e.g., *Shop Keeper, Hobby Sim* or *JRPG/CRPG*) exhibit distributions heavily skewed to the right, indicating a capacity to generate massive "hits." Conversely, more "niche" genres (e.g., *Platformer* or *Roguelike Deckbuilder*) show a more compact distribution.

2. Risk/Reward Analysis (Mean vs. Median) The disparity between the mean and the median is crucial in a market governed by a *Power Law* like Steam. We extracted the "Top 10" based on these two metrics to establish risk profiles.

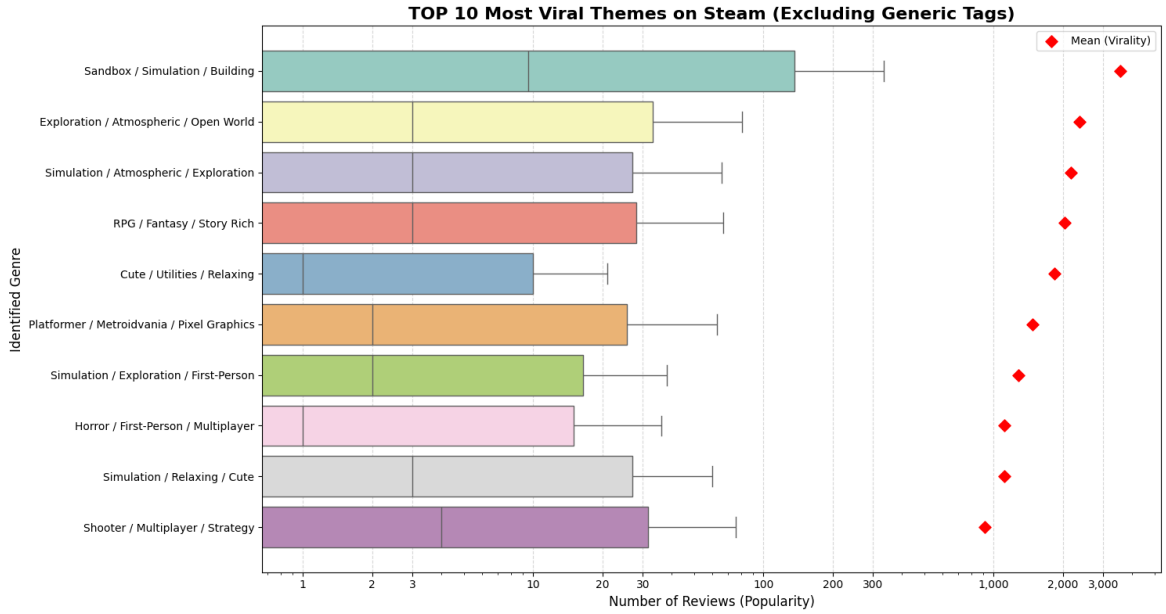


Figure 7: Top 10 by **Mean**: Virality Indicator.

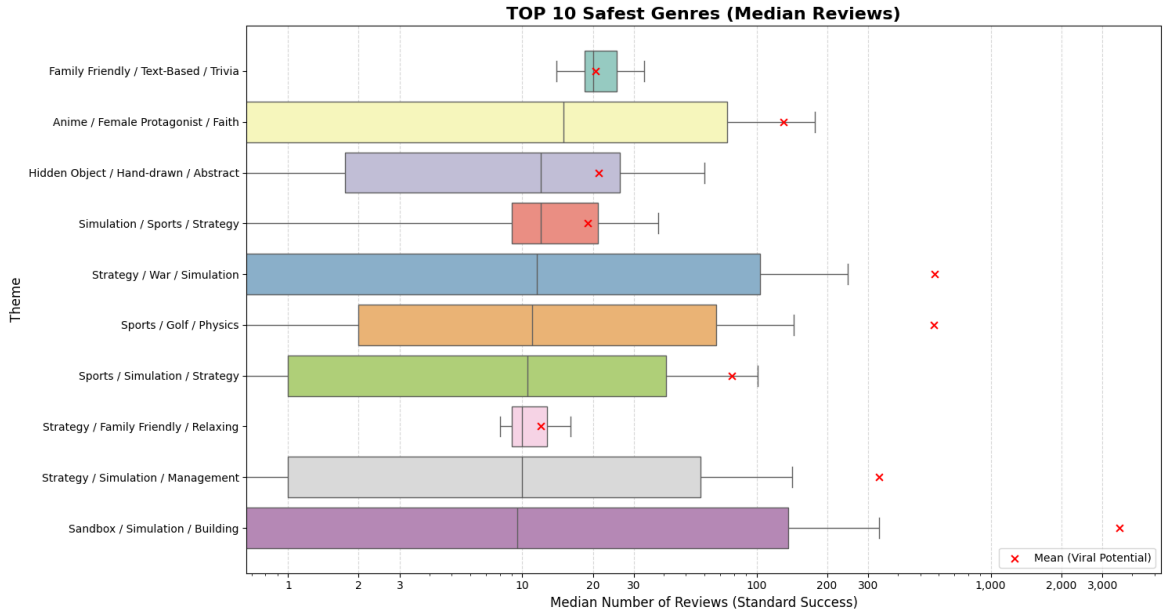


Figure 8: Top 10 by **Median**: Stability Indicator.

The comparative analysis of Figures 7 and 8 reveals two distinct development strategies:

- **"Jackpot" Strategy (Figure 7):** Genres ranked by high mean (*Multiplayer, Open world*) are driven by a few exceptional *outliers*. However, their median is often low. These are saturated markets where the risk of failure is high, but the ceiling for success is unlimited.
- **"Safe Bet" Strategy (Figure 8):** Genres ranked by high median (*Relaxing/Strategy or Sport*) guarantee a baseline audience. The "average game" in this category finds its audience. This is the most rational segment for an indie studio seeking to minimize financial risk.

Visualization Conclusion: Our clustering model does not merely group texts; it acts as a high-resolution *Market Sizing* tool.

Where classic families (e.g., *"Action"* or *"RPG"*) obscure market reality under overly generic labels, our approach captures the nuance of **sub-genres** (e.g., clear distinction between *"Action-Roguelite"* and *"Action-Platformer"*). This granularity allows for identifying, as early as the concept stage, whether a game targets a specific and stable niche or competes in a volatile mass market.

4.3 Numerical and Categorical Feature Engineering

4.3.1 Seasonality Treatment (Cyclical Encoding)

By default, machine learning models interpret months (1-12) as a linear scale, introducing an artificial discontinuity between December (12) and January (1). To preserve the cyclic continuity—crucial for analyzing seasonal trends (e.g., the "Christmas effect")—we projected the temporal variable onto the unit circle:

$$x_{sin} = \sin\left(\frac{2\pi \cdot m}{12}\right), \quad x_{cos} = \cos\left(\frac{2\pi \cdot m}{12}\right)$$

This transformation ensures that the vector distance between month 12 and month 1 is minimized, thereby facilitating the convergence of gradient-based algorithms.

4.3.2 Cardinality Reduction (Languages)

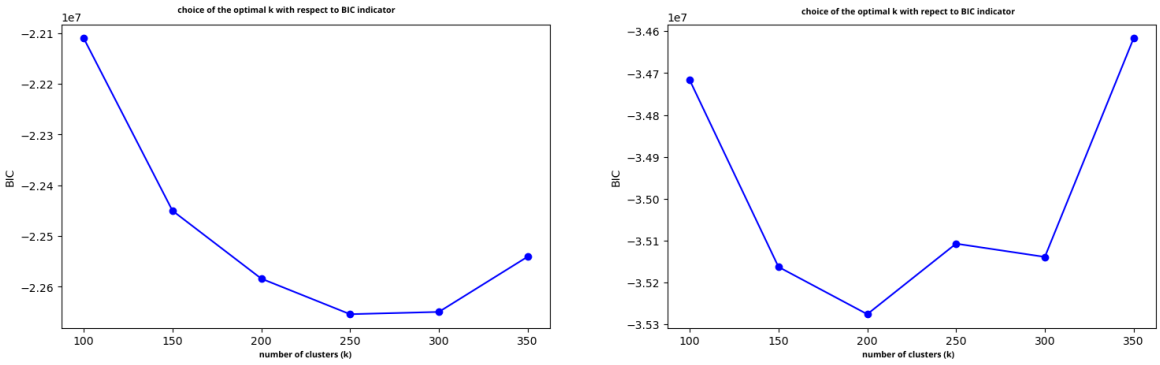
The *"Supported Languages"* variable exhibits a "Long Tail" distribution (over 200 unique values). Our reduction strategy focuses on two axes:

- **Quantitative Abstraction:** Summarization into two budget indicators: `N_langues_partiel` (text) and `N_langues_complet` (audio). Hypothesis (H1) posits that investment in dubbing acts as a strong proxy for a studio's commercial ambition.
- **Rare Label Pruning:** Application of a strict frequency threshold ($< 1\%$). Rare languages are aggregated under a generic *"Other"* label, compressing the feature space from 200 to 30 dimensions while retaining key markets (EN, CN, FR, DE, JP).

4.4 Multi-Label Categorical Feature Engineering

The high dimensionality and multi-label nature of the *Genre*, *Category*, and *User Tag* variables necessitated an information compression strategy to extract a robust predictive signal. We applied a three-step approach:

1. **Reduction and Normalization:** The binarized columns are normalized using `StandardScaler` and then subjected to PCA, retaining 95% of the variance.
2. **GMM Segmentation:** We employ a Gaussian Mixture Model (GMM) to identify game archetypes. The optimal number of clusters k is determined by minimizing the Bayesian Information Criterion (BIC).
3. **Feature Space:** The resulting clusters are reintegrated as *One-Hot Encoded* vectors and concatenated with the base variables (initial price, achievements, controller support).



(a) BIC Optimization for Genres ($k_{opt} \approx 250$). (b) BIC Optimization for Categories ($k_{opt} \approx 200$).

Figure 9: Determination of the optimal number of clusters using the BIC criterion.

4.5 Modeling and Performance Analysis

In this report, all reported R^2 scores are evaluated in log-space due to the power-law nature of the target variable (price), ensuring consistency and interpretability across all experiments.

4.5.1 Regression Architecture Selection

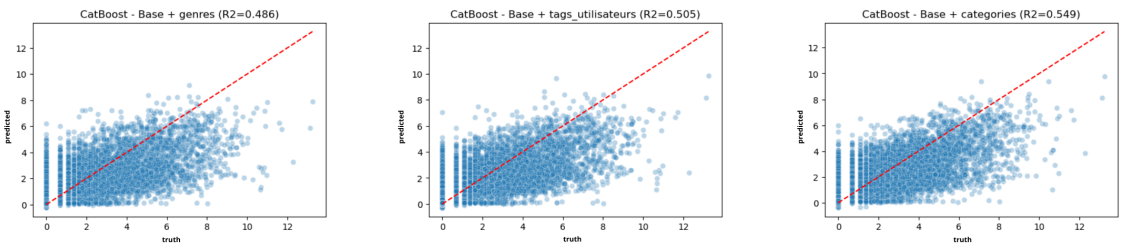
To identify the optimal model, we benchmarked three distinct approaches across each feature group: *Linear Regression*, *Multi-Layer Perceptron*, and *CatBoost Regressor*. The tests revealed a consistent performance hierarchy:

- **Linear Regression** yielded limited results (R^2 ranging from 0.08 to 0.28), failing to capture the non-linear relationships between latent clusters.
- The **Multi-Layer Perceptron** provided a notable improvement ($R^2 \approx 0.46 - 0.48$).
- **CatBoost** consistently outperformed the other models, achieving a maximum R^2 of 0.549 with the categories.

Given this technical superiority, only the results derived from the CatBoost model are detailed graphically below.

4.5.2 CatBoost Results Analysis

Given the skewness of the target variable, we applied a $\log(1 + y)$ transformation. For evaluation, a robust function performs the inverse transformation (`expm1`) with clipping at $L = 100$ to prevent numerical instability.



(a) Base + Genres ($R^2 = 0.486$). (b) Base + Tags ($R^2 = 0.505$). (c) Base + Categories ($R^2 = 0.549$).

Figure 10: CatBoost Regression Analysis: Comparison of the predictive contribution of different feature groups.

4.5.3 Discussion of Results

A comparative analysis of the plots in Figure 10 yields several key insights:

- **Information Hierarchy:** Incorporating *Categories* ($R^2 = 0.549$) outperforms *Genres* ($R^2 = 0.486$). This suggests that technical features (Cloud, Multi-player, etc.) are more reliable predictors of commercial success.
- **Model Stability:** The scatter of points around the identity line (red line) demonstrates that CatBoost effectively captures the underlying trend, despite underestimating extreme successes.

4.6 Hybrid Strategy for Multi-Valued Attributes

Handling multi-valued attributes (`tags`, `genres`, `categories`) necessitated splitting the preprocessing pipeline to adapt to each model's architecture.

- **Baseline Pipeline (MLP/Linear) - "Sparse Expansion & Compression":** Since these models require fixed-size vector inputs, we first applied comprehensive binarization (`MultiLabelBinarizer`), generating a highly sparse space with dimension $D > 800$. *Optimization:* To mitigate the *curse of dimensionality*, two reduction methods were tested:
  1. **Global PCA** retaining 95% of the variance.
  2. An experimental **Feature Clustering (PCA $\rightarrow$ GMM)** approach: applied specifically to genre, category, and tag vectors, this method uses Gaussian Mixture Models (optimized via BIC) to compress these binary vectors into a compact set of membership probabilities.
- **CatBoost Pipeline - "Native Text Ingestion":** We opted for raw text concatenation (e.g., *"RPG | Strategy"*). These fields are directly injected via CatBoost's `text_features` parameter. **Rationale:** This approach delegates tokenization to the model, enabling it to learn its own internal representation (BPE/BoW) during training. This captures high-order interactions (e.g., the correlation between *Genre=Visual Novel + Language=Chinese*) without incurring the combinatorial explosion associated with static One-Hot Encoding.

5 Model Selection Strategy

To address the bias-variance trade-off, we benchmarked three major algorithm families, ranging from simple statistical models to state-of-the-art methods. Specifically, we selected:

(i) **Linear Models** – including penalized linear regression (LASSO);

(ii) **Multilayer Perceptrons (MLP)** – capable of modeling complex non-linear relationships;

(iii) **Tree-based Gradient Boosting Models** – specifically CatBoost.

5.1 Validation and Testing

Upon data acquisition, the entire dataset was split into training and test sets using a 90:10 ratio. All feature selection, hyperparameter optimization, and cross-validation steps were performed exclusively on the training data to ensure an unbiased final performance evaluation on the test set.

5.2 Challenger Configuration (MLP)

For our predictive tasks, we designed two Multi-Layer Perceptrons (MLP) tailored for each objective:

- **Classification – Success Prediction:** The MLPClassifier is trained to minimize *log-loss* because it is a differentiable loss function essential for gradient descent unlike AUC, which is non-differentiable.

  Architecture and Key Hyperparameters: Two hidden layers of size $(128, 64)$, `ReLU` activation function.

- **Regression – Sales Estimation (Review Count):** The model minimizes `RMSE` to heavily penalize significant deviations between predictions and actual values.

  Architecture and Key Hyperparameters: Two hidden layers of size $(128, 64)$, `ReLU` activation function, `RMSE` loss function.

5.3 Champion Configuration (CatBoost)

For our predictive tasks, we deployed two optimized instances of CatBoost, tailored for each objective:

- **Classification – Success Prediction:** The model is trained to maximize AUC, ensuring strong discrimination capability and precise probability calibration. *Key Hyperparameters:* Tree depth (`depth=10`), enabling the capture of complex feature interactions.
- **Regression – Sales Estimation (Review Count):** The model minimizes `RMSE` to heavily penalize significant deviations between predictions and actual values. *Key Hyperparameters:* Tree depth (`depth=10`), `RMSE` loss function.

5.4 Comparative Benchmark

Algorithm	Role	Structural Advantage	Weakness
Logistic / Linear *(Scikit-Learn)*	**Baseline**	**Interpretability:** Coefficients β directly indicate the weight of each feature (e.g., Price impact on success probability).	**Underfitting:** Inability to model non-linear relationships (e.g., price threshold effects).
MLP *(Scikit-Learn)*	**Challenger**	**Multimodal Fusion:** Ideal for concatenating dense vectors (NLP Embeddings) with numerical variables.	**Instability:** Demands precise scaling and substantial data volumes to converge without overfitting.
CatBoost *(Yandex)*	**Champion**	**Native Handling:** Processes categories and text without high-dimensional One-Hot Encoding. *Ordered Boosting* algorithm reduces overfitting.	**Inference:** The primary bottleneck lies in memory consumption for deep trees (depth > 8), which grows quasi-exponentially.

Table 2: Benchmark Summary: GLM vs. Deep Learning vs. Gradient Boosting.

6 Experimental Feature Selection

The input space was not defined arbitrarily; instead, we adopted an empirical approach aiming to maximize the Signal-to-Noise Ratio across all models.

6.1 Ablation Protocol and Mix-Testing

We employed a *Greedy Forward Selection* strategy to construct the optimal feature vector:

1. **Univariate Validation (One-by-One):** Each candidate variable (particularly semi-structured data such as *Genres*, *Tags*, and *Categories*) was evaluated in isolation. This step allowed us to discard overly sparse technical metrics (pure noise).
2. **Combinatorial Search (Mix Match):** We tested for synergies through successive additions. For instance, the following configurations were compared:
   - *Tags* + *Genres*
   - *Tags* + *Categories* + *Languages*

   Result: We observed that certain feature combinations were redundant or uninformative. For example, the number of supported languages (audio and text), embedding clusters, the embeddings themselves, or the simple indication of controller compatibility provided negligible additional information to the model.

7 Optimal Combination and Interpretability (Golden Feature Set)

The final set selected to train all models (CatBoost, MLP, Logistic Regression) consists of the following variables, ranked by their predictive impact based on CatBoost feature importance ($R^2 = 0.75$, with $\sigma = 0.0037$ across K-fold cross-validation)

Selected Feature	Importance	Selection Rationale
Categories	**25.72%**	**Pillar #1.** Functionalities (Co-op, PvP) are more discriminative than the artistic theme.
Price	**21.35%**	**Pillar #2.** Essential economic variable (H2 validation).
User Tags	17.65%	Preferred over *Genres* for their fine granularity (e.g., *Roguelite* vs. *RPG*).
Year	11.59%	Contextual variable necessary to correct recency bias.
Detailed Description	7.87%	Better interpretation of the game's artistic direction.
Nb. Achievements	6.49%	Engagement proxy (Gamification).
Languages (partial)	4.01%	Proxy for localization budget and commercial ambition.

Table 3: Hierarchy of success factors resulting from the optimal combination.

7.1 Interpretability and Economic Validation: SHAP Analysis

In order to validate the economic coherence of our *Champion* model (CatBoost) and move beyond a *black-box* approach, a *SHapley Additive exPlanations* (SHAP) analysis was conducted. This method decomposes the marginal contribution of each variable to the final sales prediction, providing a fine-grained understanding of the key drivers of success.

The analysis confirms the hierarchy of factors identified during the experimental feature selection phase, while also clarifying the direction of their impact:

- **Feature Premium (Categories – 25.72%)**: SHAP values reveal that a game incorporating technical social features (notably *Co-op* and *Multiplayer*) experiences a significant increase in its predicted success relative to the average. This suggests that the current market places greater value on shared gameplay experiences and technical replayability than on purely artistic or thematic elements.
- **Price Signal (21.35%)**: Contrary to standard price elasticity intuition (where higher prices would reduce sales volume), SHAP analysis uncovers a positive, non-linear relationship within mid-range segments. A price positioned in the *Double-A* bracket (€20–€30) acts as a quality and ambition *proxy* within the model. This finding supports our hypothesis that players interpret excessively low prices as a signal of low quality for complex games.
- **Tag Granularity (17.65%)**: The analysis demonstrates that highly specific niche tags (e.g., *Roguelite*, *City Builder*) exert a substantially greater positive marginal impact than generic genres (*Action*, *Indie*). This confirms that specialization and precise audience targeting ("Niche Marketing") constitute an effective risk-mitigation strategy on Steam.

Overall, the SHAP analysis validates the model's alignment with economic reality: structural success depends more on tangible production choices (multiplayer architecture, production budget reflected through pricing) than on narrative concepts alone.

8 Regression Performance: Volume Estimation

The primary task was to predict the specific review volume (a proxy for sales) via regression on the logarithmic target $Y = \log(1 + \text{Reviews})$.

8.1 Error Analysis (Log-Space vs. Real-Space)

The evaluation reveals an interesting dichotomy between absolute and relative precision, typical of power law distributions:

- **Scale Accuracy (RMSE Log $\approx$ 1.00):** This is the most relevant metric for a publisher. A logarithmic error of 1.03 indicates that the model predicts the order of magnitude of success with an average multiplicative factor of $\approx 2.8\times$. *Business Impact:* The model acts as a "Tier Predictor". It reliably classifies projects into their economic tier (Niche, AA, Indie Blockbuster), enabling the alignment of production budgets with realistic market potential.
- **Sensitivity to Outliers (Real RMSE $\approx$ 4650):** The average error in terms of raw review counts appears high, but it is structural. In a dataset where a few "Black Swans" accumulate hundreds of thousands of reviews, a few large deviations on viral hits are sufficient to disproportionately inflate this quadratic metric, without invalidating the quality of the median prediction on the "core market."

9 Classification Results: The Quest for the "Golden Trade-off"

Predicting "Hit" status (> 500 reviews, $> 85\%$ positive) is a major challenge due to extreme class imbalance: only **3.2%** of games achieve this status. In this context, the Accuracy metric is misleading (a naive model that systematically predicts "Failure" would obtain 96.8% accuracy). Our evaluation therefore focuses on the ability to isolate the positive minority class, **benchmarked by** the **F1-Score**.

Model	AUC	Precision (Hit)	Recall (Hit)	F1-Score (Hit)
Logistic Regression	**0.83**	0.15	**0.79**	0.26
MLP (Neural Net)	0.64	**0.61**	0.28	0.38
CatBoost	0.77	0.38	0.57	**0.46**

Table 4: Performance comparison on the minority class "Hit".

9.1 Benchmark: The Precision-Recall Dilemma

The analysis reveals three distinct behaviors:

- **Logistic Regression:** With a record Recall of **79%**, this model misses almost no hits. However, its catastrophic Precision of **15%** makes it **operationally unviable** (6 false positives for 1 true hit).
- **MLP:** Conversely, the neural network maximizes Precision (**61%**) but collapses on Recall (28%). It is too conservative and **overlooks** 72% of market opportunities.
- **CatBoost (The "Rational Investor" Approach):** Although its AUC (0.77) is lower than the linear Baseline, CatBoost dominates on the **F1-Score (0.46)**. It achieves the optimal synthesis: it captures the majority of hits (57%) while keeping the **false positive rate under control**.

9.2 Champion Model Confusion Matrix (CatBoost)

The normalized confusion matrix confirms the model's discriminative power:

$$\begin{pmatrix} \text{TN (True Failures)} & \text{FP (False Hopes)} \\ \text{FN (Missed Opportunities)} & \text{TP (True Hits)} \end{pmatrix} \approx \begin{pmatrix} 0.98 & 0.02 \\ 0.48 & \mathbf{0.52} \end{pmatrix}$$

- **Zone of Strength:** The model filters out **98%** of non-viable games. It acts as an extremely effective "filter" to eliminate market noise.
- **Zone of Uncertainty:** On actual hits, the model detects **52%** of cases. This suggests that half of commercial successes rely on structural factors (our features), while the other half depends on unavailable exogenous factors (Marketing, Streamer Hype, Art Direction).

9.3 Selection Conclusion

We validate **CatBoost** as the final model. Adopting an ROI (Return on Investment) perspective, it is strategically preferable to operate a model that "errs twice for every success" (CatBoost) rather than one that dilutes investment (LogReg) or remains overly risk-averse (MLP).

10 Model Evaluation on the Test Set

10.1 Regression

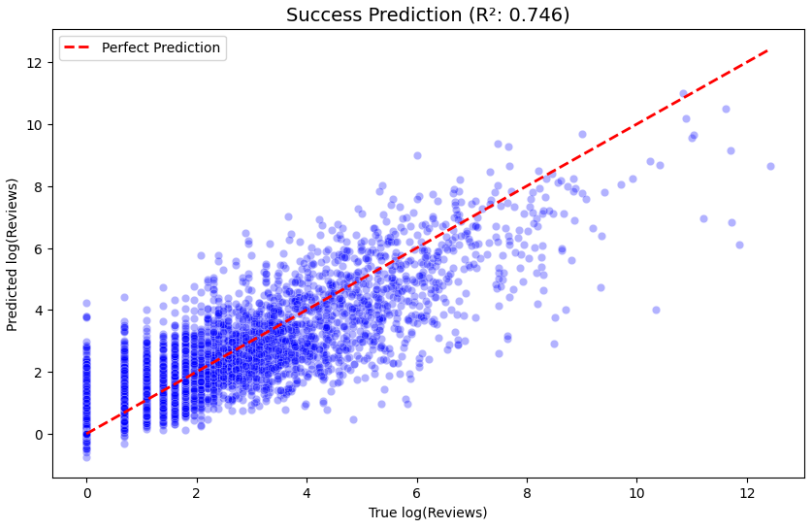


Figure 11: Prediction visualization (log-space)

The model achieved excellent performance metrics, securing an $R^2 = 0.746$, a Mean Absolute Error MAE $= 2.56$, and a Root Mean Square Error RMSE $= 1.02$ (in log-space). The model demonstrated robust generalization with no signs of overfitting.

10.2 Classification

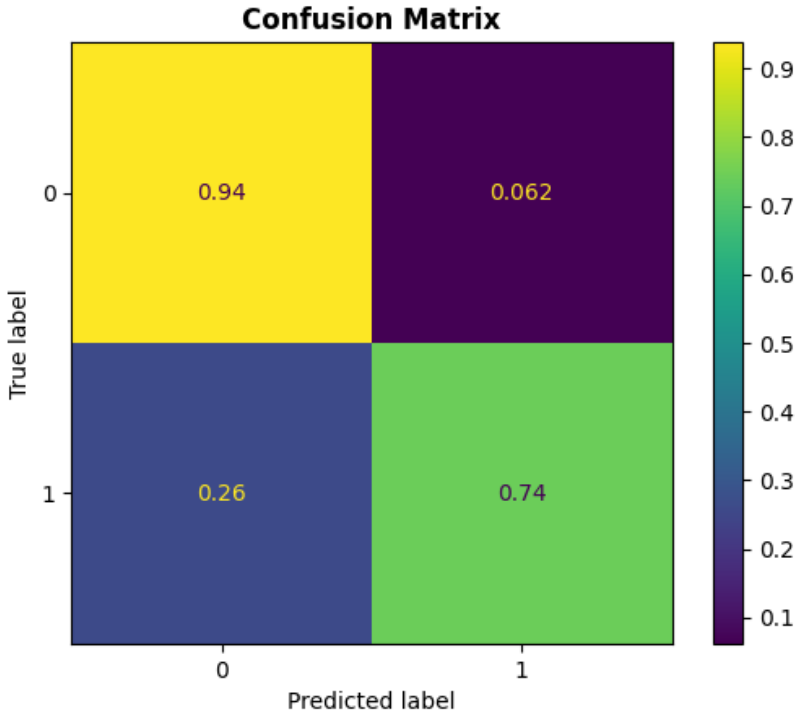


Figure 12: Prediction visualization (log-space)

The model also delivered strong classification performance. As previously discussed, it acts as an excellent filter; notably, we observed a higher True Positive (TP) rate on the Test set than on the Validation set. This confirms the model's ability to generalize correctly without overfitting.

11 Modeling Conclusion: Towards a Decision Support System

Combining a **CatBoost Regressor** with a logarithmic target allowed us to sidestep the mathematical unpredictability of "Viral Sales" (Black Swans) and focus on the fundamentals. We do not predict randomness; rather, we quantify the **structural potential** of a project. This approach transforms the model into a genuine strategic asset for two key stakeholders:

- **For the Studio (Production Steering):** The model serves as a "Greenlighting" compass. By simulating the impact of costly features (e.g., adding *Multiplayer* or full voice acting) on projected sales, the studio can align its production budget (CAPEX) with a realistic revenue forecast, avoiding the trap of unprofitable over-engineering.
- **For the Investor (Rationalized Due Diligence):** Faced with pitches often driven by emotion or "Hype," our tool offers a neutral counter-analysis based on market fundamentals. If the model predicts "Niche" success (Tier 3) for a game demanding a "Blockbuster" budget (Tier 1), a red flag is raised. Data Science does not replace artistic intuition, but it **secures the Business Case** by validating the consistency between the product promise and the statistical reality of the Steam market.

11.1 Future Work: Towards Multimodal Intelligence

Despite its performance, our model remains "blind": it analyzes the game via metadata and text, but ignores its visual dimension. Yet, in video games, **Art Direction** is a primary emotional purchase driver.

A major evolution would involve integrating a **Computer Vision** pipeline to analyze game screenshots. Using the SigLIP model to generate embeddings or fine-tuning multimodal models (e.g., *LlaVA*) would allow us to vectorize the game's aesthetics. This fusion of Text, Image, and Tabular data would constitute the ultimate step in capturing the full commercial signal.

12 Reproducibility and Code Availability

For all experiments and data processing steps, we set the random seed to 42 to ensure reproducibility.

Some models were trained using Google Colab on a T4 GPU to accelerate training.

The full code and additional resources (including figures and graphs) are available on our GitHub repository: `https://github.com/Isaac-KD/SteamStore-ANALYSE`.

For the reviewer's convenience, a description of the dataset features is provided in Appendix A.

A Appendix A — Dataset Features

Feature Name	Type	Description
game_id	Categorical	Unique identifier for each game
name	Text	Name of the game
image	URL	Link to the game image
short_description	Text	Short description of the game
detailed_description	Text	Detailed description of the game
developers	List of strings	Developer(s) of the game
publishers	List of strings	Publisher(s) of the game
genres	List of strings	Genres of the game
categories	List of strings	Categories associated with the game
user_tags	List of strings	User-generated tags
platforms	Categorical	Supported platforms (Windows, Mac, Linux)
controller_support	Boolean	Whether a controller is supported
achievements_count	Numerical	Number of achievements in the game
price	Numerical	Price of the game
reviews_count	Numerical	Number of reviews
positive_percentage	Numerical	Percentage of positive reviews
metacritic_score	Numerical	Metacritic score
total_recommendations	Numerical	Total number of recommendations
partial_languages	List of strings	Partially supported languages (text only)
full_languages	List of strings	Fully supported languages (audio + text)
num_partial_languages	Numerical	Number of partially supported languages
num_full_languages	Numerical	Number of fully supported languages
release_year	Numerical	Year of release
release_month	Numerical	Month of release
success	Boolean	Success flag (commercial or critical)
emb_i	Numerical	i^{th} dimension of the game embedding vector
cluster_BERTopic_K	Categorical	Cluster assignment from BERTopic (310 clusters)

Table 5: Complete list of dataset features for the game dataset