Lisbon School of Economics & Management
Universidade de Lisboa

# MASTER

## DATA ANALYTICS FOR BUSINESS

## MASTER'S FINAL WORK

THESIS

# PREDICTING SUCCESS IN THE INDIE VIDEO GAME MARKET

FILIPA BACHAREL IVENS FERRAZ PORTELA

JUNE - 2026

Lisbon School of Economics & Management
Universidade de Lisboa

# MASTER
## DATA ANALYTICS FOR BUSINESS

## MASTER'S FINAL WORK
THESIS

# PREDICTING SUCCESS IN THE INDIE VIDEO GAME MARKET

FILIPA BACHAREL IVENS FERRAZ PORTELA

**SUPERVISION:**
PROFESSOR DOUTOR JOÃO AFONSO BASTOS

JUNE - 2026


ISEG
Lisbon School
of Economics
& Management
Universidade de Lisboa

Lisbon School of Economics & Management
Universidade de Lisboa

# GLOSSARY

FN – False Negative

FP – False Positive

Indie Games – Video games made by a small team or single individuals

GB – Gradient Boosting

MMO – Massive Multiplayer Online

OHE – One-Hot Encoding

RPG – Role-playing game

TN – True Negative

TP – True Positive

ABSTRACT

This dissertation provides insights on the main characteristics of successful and independently made video games, commonly known as “indie games”. Using data extracted from Steam, a popular video game distributor, this study aims to find relevant characteristics and help these creators by giving them industry insights. This study compares models like the Logistic Regression, XGBoost, LightGBM and CatBoost to find the best at predicting the success of games, which is CatBoost, in this case. We then apply TreeSHAP to gather the most important features of a successful game. The results suggest that the dummy variable which decides whether a game is free or not has a positive relationship with the target variable “Success”. If the developers desire to publish the game on Steam, adding Steam-related features, such as Steam Achievements, Cloud and Family Sharing, could also benefit the model’s prediction, as these are highly ranked in the feature importance scale, with positive relationships with the likelihood of success. The genre of the game had less impact than expected on the model’s classifications, although the presence of some genres, like Action, Adventure, and Strategy, was shown by the model to have a negative relationship with the likelihood of success of an “indie” game.

KEYWORDS: Success; Indie; Video Games; Machine Learning; Statistical Models; Steam

Lisbon School of Economics & Management
Universidade de Lisboa

# TABLE OF CONTENTS

## TABLE OF FIGURES



## TABLE INDEX

Lisbon School of Economics & Management
Universidade de Lisboa

## ACKNOWLEDGMENTS

This dissertation marks the end of an invaluable academic experience, and as such I would like to thank all of those who helped me along the way, without whom this would not have been possible.

First, I am forever grateful to my parents, who supported me unconditionally and consistently motivated me to see this journey through. Your belief in me is my greatest strength.

I am also thankful to my brothers who offered plenty of advice and support and are a source of inspiration to me.

To my friends, who motivated and helped me unwind when I needed it most, I could not have done it without you.

And I would also like to express my immense gratitude for Professor Doutor João Afonso Bastos, who guided me and offered precious advice throughout the whole process.

# 1. INTRODUCTION

## *1.1 Research question and Relevance of the Study*

The video game industry is a competitive one. Video games have been compared to the making of Hollywood movies, with extensive budgets and massive teams (Pashkov, 2021), leading to the creation of jobs and, occasionally, masterpieces. In this industry, the independent game developers, or "indie" game developers may be considered the "underdogs". These are video game developers, usually individuals or small teams, who do not receive large amounts of financial aid or do not have a contract with a video game publisher. Some of these developers have seen their games become some of the most famous on the online game distributor Steam! In fact, on Steam, almost all games are indie games: in 2024, indie games released on Steam accounted for 98.9% of the market share. Despite this, they only made up half of video game units sold and generated around a third of Steam's total game revenue (Stoll, 2025a). AAA (or triple-A) gaming studios are mid-sized or major publishers, which usually have more resources for the production, distribution and marketing of video games. These studios tend to make more profit due to their larger budgets, better understanding of the market they are in, and higher prices, while also facing higher risks if a project is badly received by the audience or falls through at launch (Semenchenko et al., 2025). It is through this lens that the research question "What are the main characteristics of a successful indie game?" will be helpful for future indie game developers to understand their audience and adjust costs and expectations when working on such a big project as a video game.

## *1.2 Objectives of the Study*

Using a dataset scraped from Steam in 2024, it is possible to apply Machine Learning techniques and obtain relevant conclusions about this sector of the industry. With these conclusions, indie game developers will be able to make informed decisions about the genre, style and type of game they are producing, improving their audience targeting and possibly increasing their sales.

## *1.3 Study Structure*

This study is divided into five chapters.

The first chapter, the Introduction, builds a context for the study and establishes its goals.

The second chapter, the Literature Review, examines and compiles previous academic works pertaining to the video game industry and indie game development and publishing, as well as the models and evaluation criteria most commonly used.

The third chapter is where the data cleaning and preparation steps will be described, as well as the description of how the models will be applied.

In the fourth chapter, the Results will be shown, the models chosen and conclusions will be taken from the SHAP tool. The Discussion will compare the findings from the Results with those from the Literature Review, and addresses its limitations, as well as the chance for further research to be done.

Lastly, the Conclusion presents this study's findings, limitations and further research in a more summarized manner.

# 2. LITERATURE REVIEW

In this segment of the work, research is conducted to comprehend the context and position of indie gaming in the industry of video games, check which characteristics are thought to be the most prominent in successful video games, in order to compare them with our findings later on, and find the best models from which our conclusions will be drawn and measures to evaluate these models.

## *2.1 Video Game Industry*

With the evolution of technology, video games no longer needed to be sold in person, and that opened doors for all kinds of developers, with large and small budgets (Josef et al., 2022), as there was no need for physical manufacturing, distribution and reliance on retail as stated by Klézl & Kelly (2022). Now, game developers only need to focus on building the game.

As mentioned in Josef et al. (2022), at a high level, there are three main market players: developers, the teams or individuals who actually make the game; investors, who fund the game monetarily; and publishers, who offer a helping hand when bringing the game into the market, being a combination of investors, marketers and support developers. It should also be noted that developers can be part of the publishing company, in which case they are called first-party developers; developers may also be commissioned by the publisher to create a game that will fit their vision, these are second-party developers; and third-party developers are those who have their own game and try to sell it to a publishing company (Klézl & Kelly, 2022). The third-party developers are where indie developers, the focus of this work, are sometimes situated, if they decide not to self-publish.

While some authors argue that the definition of indie game developers is debatable (Josef et al., 2022), a consensus seems to be that these are small teams or individuals, who are not owned by a publisher, but may resort to one at a later stage in game development (Klézl & Kelly, 2022). These teams also typically work with smaller budgets, and may resort to crowdfunding if they have a large enough following on social media.

The whole process of developing and publishing a game will not be the focus of this work, but for industry context purposes here are some tasks that go into it: the design, art, programming and music are all core development tasks. The following tasks are the most

common ones that indie game developers tend to seek help from publishers: quality assurance, porting to additional game systems (that is, making the game available on multiple platforms, such as PC, consoles and mobile devices), translations, distribution and marketing.

Indie teams have given rise to mainstream games such as Minecraft, Terraria, Rocket League and Among Us, among others. We can also see that, as they rise in popularity, they may attract larger corporate entities, as Mojang, the firm behind Minecraft, did when it was purchased by Microsoft in 2014, for 2.5 billion dollars (Miller, 2014). As Keogh (2019) states, "Individual content creators are increasingly afforded by and rely on formal corporate platforms to distribute their creative activity to dispersed niche audiences while, at the same time, these formal platforms are commercially dependent on the content produced by informal creators".

Another definition that is relevant for the context is the concept of AAA (triple-A) studios. These are well-established developer studios, with large production budgets, big teams, partners and which typically do their own publishing, so they are first-party developers. The games they make have been compared to Hollywood blockbusters (Josef et al., 2022).

As was mentioned previously, self-publishing has become much easier in the present, and it is noticeable. There has been a significant increase in self-publishing through platforms like Steam, and they are becoming a "central part of the industry and culture of digital games" (Klézl & Kelly, 2022), as they are estimated to account for 98.9% of all games on Steam as of 2024, and produced almost half of Steam's total game revenue, according to VGI (2024).

The video game industry had experienced growth since the pandemic, but this surge seems to have slowed down. Nonetheless, the global games market was estimated to have generated $188.8 billion in 2025, which corresponds to a 3.4% increase from 2024 (Buijsman, 2025). However, due to technological advancements and customer expectations, game development costs have been increasing, which has generated some difficulty for mid-tier actors. This has created a trend where major actors in the industry are acquiring these mid-tier actors (Klézl & Kelly, 2022).

Although not included in this project, it is impossible to talk about video games without mentioning the biggest gaming platform by global revenue: the mobile phone. In 2025, a study performed showed that this platform amassed about 55% of the global revenue, beating PC (21%) and console (24%) by a landslide (Buijsman et al., 2025). This sector of the video game industry, while "younger" than the other two, accounted for 83% of all players worldwide in 2025. Due to its lack of need for a fixed location (whether in homes or arcades), casual nature and sheer offer volume (Stoll, 2025b), video games on the mobile phone have seeped into the daily lives of many people and have become mainstream (Mäyrä & Alha, 2020). Back when smartphones were starting to appear, video game apps followed a premium business model, where there was a one-time purchase fee, but due to increasing competition, prices were lowered and eventually the market shifted towards a free-to-play revenue model, by showing in-game advertisements and in-app purchases, which is what Mäyrä & Alha (2020) call the Freemium model. This model draws in high profits from a small percentage of players. This has been a source of criticism, as companies profit from social and psychological techniques to create these high spenders.

### *2.2 What makes a video game good*

In Ullmann et al. (2022), the authors analysed a data set of 200 video games regarding several factors: from team sizes to game genres and game modes. Note that this data set is not limited to indie games. Their most relevant findings to this study revealed that shooter games, RPG (Role-playing games) or simulation games were more prone to success, whereas action, adventure, strategy and platform games tend to underperform. They also found that multiplayer games tend to appear more among high-rated games than low-rated ones. The same was found for 3D games.

Other authors, such as Bond & Beale (2009), whose studies are based on the analysis of video game reviews, also mention that some of the more important elements of a good game come from its game design, which is the process of creating and shaping the mechanics, systems, rules, and gameplay of a game. The authors point out that a cohesive, varied game with good user interaction and meaningful social interaction is what users believe to make a good game. This view is supported by Keijser (2012), where the authors explain that while developing and producing the game, one should always think about

how the players will play and interact with the game and with each other. Ullmann et al. (2022) sustain that game design is what many game developers think should be one of the main focuses of game developing. In many of their interviews with high-rated game developers, having a "vision" or "blueprint", and having it maintained by designers and managers throughout the process is said to be key to the success of a game.

Another point of view brought up by Keijser (2012), is the artistic view. While it is true that having a target audience and catering to it is important, various developers state that designing a game that you can be proud of and enjoy making might be one of the most important factors.

This may be especially true for indie games, where the team might be smaller and closer to its audience via social media, which makes interactions between players and developers seem even more important. These interactions are where the players get to see all the work and love that goes into building a game and may create some emotional connections between players, the game and the developer.

### *2.3 Models used for prediction*

As previously stated, the goal of the work is to identify the most impactful characteristics of a successful indie game, so here we will discuss which models are more adequate for this task.

Defining a successful game may be viewed as classification problem or as a regression problem. The first one would have us define two categories: successful and not successful (Cox, 1958), with successful encoded as 1 and not successful as 0. Here, success consists of a game rating above 5, on a 1 to 10 scale. While it is not an especially strict threshold, given that a rating of 6, for example, may not be considered a "success" by most companies' standards, it provides an intuitive benchmark, as the midpoint of the scale. The regression problem would have us trying to estimate the rating itself, from 0 to 10, where 10 is the highest possible rating. For this work, the approach chosen was classification.

A good baseline for comparing models in a classification setting is the Logistic Regression Model. It is a statistical classification model used to estimate the probability of an outcome. Given its S-shaped curve, it makes sensible predictions bounded to the interval (0,1). The Logistic Regression Model is based on the logistic function, which

transforms a linear combination of predictors into a probability between 0 and 1, and for a single independent variable can be written down as

$$p(Y = 1 \mid X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}, \quad (1)$$

where $p(Y = 1 \mid X)$ represents the probability that the dependent variable belongs to the class encoded as 1. By rearranging this expression, the final equation is reached:

$$\log(\frac{p(Y = 1 \mid X)}{1 - p(Y = 1 \mid X)}) = \beta_0 + \beta_1 X, \quad (2)$$

where the left-hand side is called the log odds or logit. The odds, $\frac{\mathrm{p(X)}}{[1-\mathrm{p(X)}]}$, take values between 0 and ∞, meaning very low and very high probabilities of something happening, respectively. The parameters in this model are estimated using the Maximum Likelihood Estimation, which seeks the set of coefficients that maximise the likelihood of observing the given data. The Logistic Regression Model outputs probabilities that an observation belongs to a given class, which in the case of a binary classification problem such as this one, can then be converted into class predictions using a decision threshold, commonly 0.5.

Decision trees are another popular way of solving classification and regression problems, due to their simplicity. As McCarthy et al. (2019) explain, "decision trees use algorithms to determine splits within variables that create branches. This forms a tree like structure. The algorithms create a series of IF-THEN-ELSE rules that split the data into successively smaller segments. Each rule evaluates the value of a single variable and based upon its value splits it into one of two or more segments.".

XGBoost, LightGBM and CatBoost are gradient boosting frameworks built on decision trees. Gradient Boosting (hereafter referred to as GB) methods are those which learn from their mistakes in past trees, by focusing on the residuals and building on top of each new model, and repeating the process until the limit established or until the fit of the model no longer improves.

XGBoost, short for eXtreme Gradient Boosting and one of the most well-known GB methods, makes solving real world scale problems possible while minimising resource usage (Chen & Guestrin, 2016). It was, in 2014, a significant advancement which cut

down training duration time by significant amounts and allowed data scientists to process larger quantities of data, making it a scalable option (Verma, 2024). This scalability is partly achieved through parallelized computation and support for both row and column subsampling, which speeds up the splitting process and helps prevent overfitting. XGBoost works on a depth-wise tree, meaning it grows one full level at a time, producing a balanced tree where splits (decisions) vary across nodes.

LightGBM is another well-known GB decision tree algorithm developed by Microsoft in 2016. Using a histogram-based algorithm speeds up the training process and reduces memory usage, as it buckets continuous feature values into discrete bins rather than evaluating every possible split point. Both Ke et al. (2017) and Machado et al. (2019) describe the model as a faster alternative to XGBoost, and Machado et al. (2019) report better results with the LightGBM model. LightGBM is a leaf-wise tree, where at each step, the algorithm picks the single leaf with the largest loss reduction and grows the tree where loss improvement is highest. This contrasts with XGBoost's depth-wise approach and can yield better accuracy for the same number of iterations, though it carries a higher risk of overfitting on smaller datasets. This process repeats until the maximum number of leaves defined in the parameters is reached.

CatBoost is an algorithm specifically designed for categorical problems, developed in 2017, and which claims to outperform other gradient boosting decision trees, like XGBoost and LightGBM (Prokhorenkova et al., 2018). This model is based on a symmetric tree, which uses the same split, that is the same feature in the same threshold, at every node in a given level, and chooses the split with the largest loss reduction and applies it to all leaves. Being a model dedicated to categorical problems, it includes an embedded one-hot encoding feature, handling them internally through a method based on target statistics.

### *2.4 Evaluation Measures*

The following evaluation measures will be used to compare our models.

a. Accuracy

$$Accuracy = \frac{TN + TP}{TN + FN + TP + FP} = \frac{Correct\ Classifications}{Total\ Classifications}. \quad (3)$$

Accuracy measures overall prediction correctness, and divides correctly predicted events by total events.

b. Precision

$$Precision = \frac{TP}{TP + FP}. \quad (4)$$

Precision measures the quality of positive predictions, as it divides correctly predicted positive events by all positive predictions. A higher precision score means that the algorithm is providing more "relevant" results than "irrelevant" ones.

c. Recall (Sensitivity)

$$Recall = \frac{TP}{TP + FN}. \quad (5)$$

Recall evaluates the model's ability to detect positive events correctly, that is the percentage of accurately predicted positive events out of all actual positive events. A higher recall means the algorithm predicts the majority of the relevant results correctly.

d. F1-Score

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall}. \quad (6)$$

The F1-Score is a harmonic mean of the precision and recall of a model. The F1-score of a perfect model is 1, in which the contribution of precision and recall are equal.

e. ROC Curve & Area Under Curve

The Receiver Operating Characteristic (ROC) Curve measures the discriminatory power of a binary classifier, that is, how well the model can distinguish between the two classes. The ROC curve plots the Recall or True Positive Rate (TPR) on the y-axis against the False Positive Rate (FPR)

$$FPR = \frac{FP}{FP + TN}, \quad (7)$$

on the x-axis.

The larger the value, the better discriminatory power the model has. The area under the ROC is the Area Under Curve (AUC), and it ranges from 0 to 1, with values closest to 1 indicating a better discriminative power and fewer classification errors.

### *2.5 TreeSHAP*

While feature importance is a useful tool, there is a way to tell how each feature impacts the dependent variable Y.

SHAP (SHapley Additive exPlanations), and TreeSHAP by design, depend on Shapley Values, introduced in 1953 by Shapley (1953). These values are based on game theory and consist of assuming a prediction is a "payout", and each feature value is a "player". These Shapley Values then tell us how to fairly distribute the "payout" among the players, and thus, who has contributed more in the eyes of the model (Molnar, 2025a). The Shapley Value for each feature represents the average marginal contribution of a feature value across all subsets of features.

SHAP as a machine learning interpretability method was proposed in 2017 by Lundberg & Lee (2017), and it is essentially a new way of estimating the Shapley Values. It proposes a unified framework for interpreting predictions, as well as a way of seeing why each prediction was classified the way it was. The "Additive" in its name stands for its additive feature attribution, meaning the sum of SHAP values in a prediction equals the model's prediction minus the average prediction.

TreeSHAP is a variant of SHAP for tree-based machine learning models, just like the ones that will be used in this project. Proposed by Lundberg et al. (2019), it was introduced as a fast alternative to KernelSHAP, another SHAP variant.

According to Molnar (2025b), SHAP has clear advantages: a solid theoretical foundation in game theory, fair distributions of the prediction among feature values and explanations that allow the user to compare between the prediction and the average prediction.

# 3. DATA PREPARATION AND MODELLING

## *3.1 Data Understanding*

The data are readily available, as they had been previously scraped and uploaded on GitHub (NewbieIndieGameDev, 2024) in October of 2024. They consist of several .csv files imported from Steam and SteamSpy. It is important to acknowledge that we are working under the assumption that these data have not been altered since scraping and are aware that there is a risk of the data being incorrect.

The initial data, described below in Table I, include several interesting features besides the ones that will be used, such as descriptions of the game and data from SteamSpy, a website which gathers data and statistics from Steam.

TABLE I – DATASET CONTENTS

| File Name | Contents |
|---|---|
| categories.csv | app_id and categories |
| descriptions.csv | app_id, summary, extensive, and about |
| games.csv | app_id, name, release_date, is_free, price_overview, languages and type |
| genres.csv | app_id and genre |
| promotional.csv | app_id, header_image, background_image, screenshots, and movies |
| reviews.csv | review_score, review_score_description, positive, negative, total, metacritic_score, reviews, recommendations, steamspy_user_score, steamspy_score_rank, steamspy_positive, and steamspy_negative |
| steamspy_insights.csv | app_id, developer, publisher, owner_range, concurrent_users_yesterday, playtime_average_forever, playtime_average_2weeks, playtime_median_forever, playtime_median_2weeks, price, initial_price, discount, languages, genres |
| tags.csv | app_id and tag |

After examining the contents of each file, and taking this study's goal and scope into account, the files selected for further analysis were games.csv, categories.csv, genres.csv and reviews.csv.

The games.csv file contains necessary information for identifying the game being analysed, while the reviews.csv contains what will be the deciding factor of success: the number on a scale of 1 to 10, with 10 being the highest score a game can have. The categories.csv and genres.csv are the closest to objective game characteristics there is. While genre.csv includes the genre of the game (Action, Indie, Strategy, Sports, etc.), categories.csv covers how each game is played or functionalities available (Single-player, Multi-player, Virtual Reality Support, Split Screen, etc.). The tags.csv was also considered for this project, as it contained more keywords for each game, but due to computational and time restrictions, it was not included.

Following this decision, identifying any data quality issues is crucial. In the games.csv file, unformatted text can be found, along with part of the content being misaligned with the headers. In the genre and category files, different languages were used to describe the same word. In the reviews file, some columns display no data.

Almost all the data that will be used for the models are categorical, meaning that they will be one-hot encoded when the final database is being prepared.

### *3.2 Data Preparation*

The initial data were in separate .csv files and had to be joined into one.

In the first .csv file, Genres, we filtered out all games that were not classified as “Indie”. All genres were then translated into English to avoid duplicated genres in different languages. This table was then one-hot encoded.

The Categories dataset was treated similarly: non-indie video games were filtered out, all categories were translated to English and one-hot encoded into columns. Lastly, MMO (Massive Multiplayer Online) was included in both Genre and Category. The decision was made to eliminate it from Category, to avoid confusion and mistakes in the next stages of the modelling process.

In the Games dataset, some additional preprocessing steps were required, due to missing game release dates and misaligned header and content in the original table. The table was then filtered for indie games only.

The Reviews dataset had some data type issues, which were corrected, and was then filtered for only indie games. Columns that had substantial amounts of missing data or

text content and that would not be used for this work were dropped. The Reviews file was not one-hot encoded, since the data are quantitative and the target variable originates from this file. Lastly, these four datasets were all merged into one dataframe.

After analysing the full dataframe, a few columns were dropped, because they were not going to be used in this study. The "demo" type games were excluded from this table, since they are only free demonstrations that are released early in the development stages, that work as a showcase for how the real game will be and are a way to get feedback, and thus, do not represent finished games.

The target variable was then created: from the feature "review_score", which has values ranging from 1 to 10, the feature "Success" is defined. It is a Boolean feature which takes a value of 1 when the "review_score" is above 5.

The dataframe now has 28 237 rows and 79 columns, where each row corresponds to one game. From this dataframe, we can analyse the class ratio: around 20.3% of this dataset is classified as successes.

In preparation for the application of models, the independent variables were defined, the dependent variable was identified, and the constant columns were removed. The data was then divided into two sets: a training and validation set (*X_trainval*), and a test set (*X_test*), where the test set consists of 20% of all the observations. Due to the class ratio, a stratified k-fold has also been implemented, with k=5, and "shuffle" set to True. The process of splitting the training and validation set into two subsets is done later on, by the models themselves.

### *3.3 Modelling*

After creating the dataframes, we can now train the models.

Firstly, the Logistic Regression is trained as it will be used as a benchmark for the remaining models. The only parameter that was set was the *max_iter*, set to 1000, which controls the maximum number of iterations the solver is allowed to run. Then, having separated the data into 5 folds, the repetitive process of selecting a fold as a validation set and using the remaining folds as the training set begins, and happens 5 times, as there are 5 folds and each fold will be a validation set once. From this process, we can gather the average of six metrics across the 5 steps: Accuracy, F1-Score, ROC, Precision and Recall.

These measures are used for checking consistency within steps through their average value and standard deviation. After cross-validation, the model is retrained on the entire training and validation set (*X_trainval*), and evaluated once on the test set (*X_test*), which the model has never seen before. The same metrics are then measured, and these are the ones that will be used to compare the model's performance against the others.

XGBoost is the next model. To set a limit on the algorithm, *n_estimators* was set to 500, meaning that there will be a maximum of 500 trees built. To determine the best hyperparameters and instead of exhaustively trying all combinations from a selection of grid values, randomized search with cross-validation was used. It samples *n_iter* (50 in all these cases) random combinations from a provided hyperparameter grid, reported in Table II, with the values tried.

TABLE II - XGBOOST'S HYPERPARAMETERS

| Parameter | Role | Values in grid | Value selected |
|---|---|---|---|
| max_depth | Maximum depth of each tree | 3, 5, 7, 9, 11 | 5 |
| min_child_weight | Minimum sum of instance weights in a child node | 1, 3, 5, 7 | 1 |
| subsample | Fraction of training rows sampled per tree | 0.6, 0.7, 0.8, 0.9, 1.0 | 0.6 |
| colsample_bytree | Fraction of features sampled per tree | 0.6, 0.7, 0.8, 0.9, 1.0 | 1.0 |
| learning_rate | Shrinkage applied to each tree's contribution | 0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.29 | 0.01 |

The search is evaluated with the same stratified folds and optimised for ROC, so that the best hyperparameter set will maximise that measure.

Ideally, early stopping would be applied in this step, but due to *sklearn API* limitations, early stopping during cross-validation was only applicable to CatBoost. For XGBoost and LightGBM, early stopping was applied during final model retraining, following the approach recommended in their respective official documentation.

Once again, the results are gathered per fold and an average with standard deviation is calculated to understand the stability of the model. After gathering the best hyperparameters, the model is retrained on 90% of the *X_trainval* set, with the remaining 10% used as an early stopping set (*eval_set*). Early stopping was set to 15, meaning that if the *logloss* on the *eval_set* set does not improve for 15 consecutive rounds training will stop. This helps prevent overfitting the model to the training data. The final evaluation of the model via those 6 metrics is, again, tested on the *X_test* set, which the model has not seen before.

LightGBM follows the same method as XGBoost, with the main difference being the parameters that are tuned. Table III shows the hyperparameters and respective values tuned.

TABLE III - LIGHTGBM'S HYPERPARAMETERS

| **Parameter** | **Role** | **Values in grid** | **Value selected** |
|---|---|---|---|
| max_depth | Maximum depth of each tree | 3, 5, 7, 9, 11 | 9 |
| subsample | Fraction of training rows sampled per tree | 0.6, 0.7, 0.8, 0.9, 1.0 | 1.0 |
| learning_rate | Shrinkage applied to each tree's contribution | 0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.29 | 0.01 |
| num_leaves | Controls the maximum number of leaves a single tree can have | 20, 40, 60, 80, 120, 150 | 20 |
| min_data_in_leaf | Sets the minimum number of samples required in a leaf node before a split is accepted | 1, 3, 5, 7, 10 | 7 |

The remaining process follows XGBoost's steps.

CatBoost is the final model that was tested in this work. The process itself is the same, with the exception that early stopping was a possibility in the training phase, accelerating the process.

TABLE IV - CATBOOST'S HYPERPARAMETERS

| Parameter | Role | Values in grid | Value selected |
|---|---|---|---|
| iterations | CatBoost's equivalent of n_estimators | 50, 100, 200, 500 | 100 |
| depth | Maximum depth of each tree | 3, 5, 7, 9, 11 | 5 |
| learning_rate | Shrinkage applied to each tree's contribution | 0.01, 0.03, 0.05, 0.1, 0.2, 0.3 | 0.1 |
| l2_leaf_reg | L2 regularisation penalty applied to the leaf values | 1, 3, 5, 7, 10 | 3 |
| bagging_temperature | Controls the randomness of the Bayesian bootstrap used for row sampling | 0.0, 0.25, 0.5, 0.75, 1.0 | 0.25 |
| border_count | Controls how many split points are considered for numerical features | 32, 64, 128, 200, 255 | 32 |
| random_strength | Controls how much randomness is injected into the score of each candidate split during tree building. A mechanism for preventing overfitting during the early stages of training. | 1, 3, 5, 7, 10 | 1 |

The search strategy is identical to the previous two models: 50 random combinations, stratified K-fold cross-validation, with k=5, ROC as the selection criterion, and results across folds reported as mean and standard deviation. Early stopping is again set to 15 rounds, with 10% of *X_trainval* used as the early stopping validation set. The final metrics come from the *X_test* once again.

# 4. RESULTS

## *4.1. Model Evaluation*

Table V shows the performance metrics discussed in the literature, applied to all the models described above.

TABLE V - MODELS' RESULTS

| Model / Metric | Logistic Regression | XGBoost | LightGBM | CatBoost |
|---|---|---|---|---|
| **Accuracy** | 0.8548 | 0.8550 | 0.8546 | 0.8532 |
| **F1-Score** | 0.6943 | 0.6934 | 0.6919 | 0.6973 |
| **Precision** | 0.6069 | 0.6080 | 0.6078 | 0.6003 |
| **Recall** | 0.8110 | 0.8066 | 0.8031 | 0.8319 |
| **ROC (AUC)** | 0.9249 | 0.9254 | 0.9243 | 0.9264 |

Overall, the four models perform similarly.

Accuracy values exhibit little variation, remaining around 85.5%, with CatBoost ranking below the other models at 85.32%. XGBoost shows the greatest Accuracy value at 85.5%.

The F1-Score values are all approximately 69%, with CatBoost achieving the highest score at 69.73%. These are, in general, lower values than Accuracy, but that may be explained by the class imbalance in the dataset: only 20.3% of the dataset consists of positive cases. Because Accuracy incorporates true negatives into its calculation, a model can achieve relatively high Accuracy by correctly classifying the majority negative class. However, Precision and Recall only consider how well the model handles the positive class, which in this dataset is the minority class. As a result, these metrics may provide a more informative assessment of the model's ability to identify successful games.

Regarding Precision, all models perform similarly, with values clustered around 60%, ranging from 60.03% (CatBoost) to 60.80% (XGBoost). This suggests all models produce a fair number of false positives. Recall is higher than Precision across all models, with

values ranging from 80.31% to 83.19%. This pattern reflects a trade-off between Precision and Recall: the models identify a high proportion of actual positives (high recall) but at the cost of an increased number of false positive predictions (lower precision).

In ROC-AUC, which measures a model's capacity to discriminate between classes, the models' scores range from 92.43% to 92.64%. If we optimize for ROC-AUC, CatBoost edges out the competition, but the margin is narrow, only becoming apparent at the first decimal place when expressed as a percentage.

The chosen model is CatBoost with the original dataframe, due to the marginally better performance metric in F1-Score and ROC, although the other models prove to be close competitors.

### *4.2 TreeSHAP Application and Results*

The TreeSHAP summary plot, shown in Figure 1, combines feature importance with feature effects. Each point represents a single observation, with its horizontal position representing the SHAP values, indicating the direction and magnitude of that feature's contribution to the model's predicted outcome (successful or unsuccessful) for each game. The feature value can only be 1 or 0, thus explaining the exclusively red or blue colouring, respectively. The features are ranked from most to least impactful along the vertical axis.

The most influential feature, by a considerable margin, is *is_free*, a binary indicator of whether a game is available for free or not. Figure 1 suggests a strong positive relationship between this feature and the likelihood of success of a game. The magnitude of this feature's effect exceeds all others in the feature set, as can also be observed in Figure 4 in the Appendices.

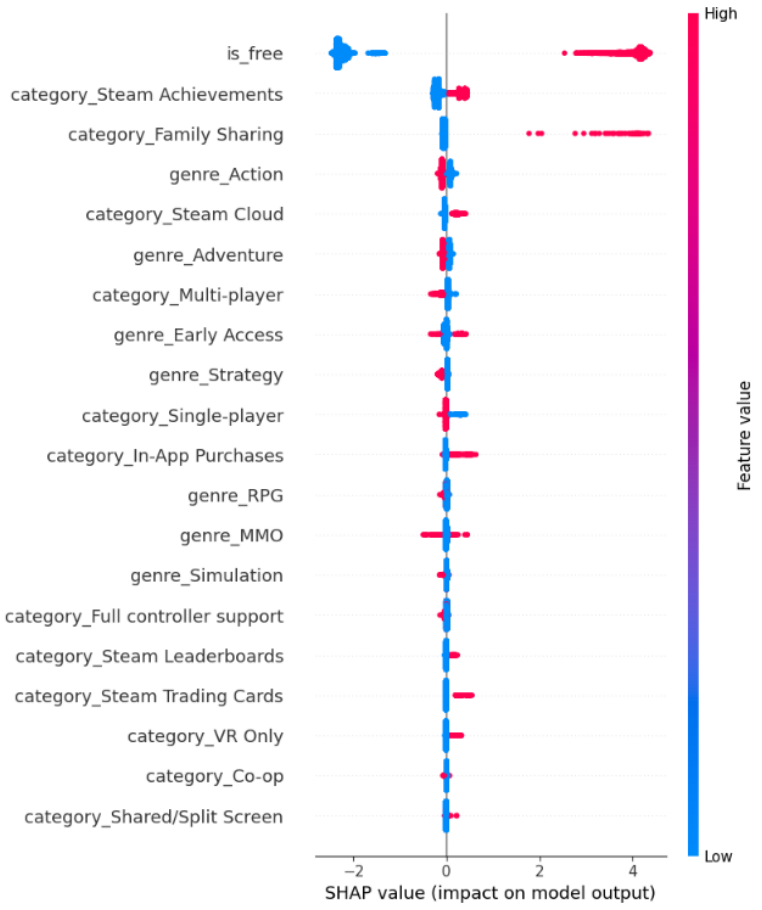


Figure 1 – Summary plot of CatBoost

The second most impactful feature for the model is Steam Achievements, a widely adopted mechanism where players are awarded a badge-like recognition when achieving a notable in-game milestone. This feature appears to have a positive relationship with the target variable "Success", although less impactful than *is_free*.

Family Sharing is the third most impactful feature and belongs to the Category space, similar to Steam Achievements. This is a Steam feature which allows members of a shared family group access to one another's game libraries, thus reducing the need for duplicate purchases (Steam, 2024). Figure 1 indicates a positive relationship between this feature and the likelihood of success, though it displays high variability, meaning its effect on the prediction changes significantly across different observations.

The first genre-level feature to appear in the ranking is Action. Action is the most impactful feature with a negative relationship with the target variable.

The following feature is part of the Category space, and it is Steam Cloud. Steam Cloud enables automatic synchronisation of game save data to Valve's (Steam's developer) cloud infrastructure, and it is subject to per-game storage limits (Steam, 2021). The summary plot reveals a positive relationship for this feature.

Adventure is the second genre to appear in the top features. Similar to Action, the Adventure genre appears to have a negative relationship with the likelihood of success.

Multi-player and single-player present a seemingly counterintuitive pair of results. Multi-player means multiple people can play and interact with each other in-game, while single-player implies an unaccompanied experience. According to the summary plot, both multi-player and single-player features have negative relationships with the likelihood of success of a game. This suggests that the likelihood of success of a game would increase, if it was neither multi-player nor single-player, which may seem paradoxical. However, on Steam, these two tags are not mutually exclusive, since a game may be tagged as both, neither, or as only one of them; as such, a game that was not tagged as multi-player will not necessarily be tagged as single-player.

For a more detailed analysis, this summary plot was split into the two main groups: Genre and Category, respectively shown in Figures 2 and 3.

The genre-level plot, Figure 2, reveals the patterns in greater detail. Besides the already mentioned Action and Adventure genres, below are a few more genres that had relevant impact, as can also be verified in Figure 5, available in the Appendices.

Early Access (an early version of the full game with more content than a demonstration ("demo") version of a game, and that is paid for) displays no clear relationship, since positive values aren't associated with a particular direction, and displays a wide SHAP value spread, which means this feature has high variability in its impact on model output.

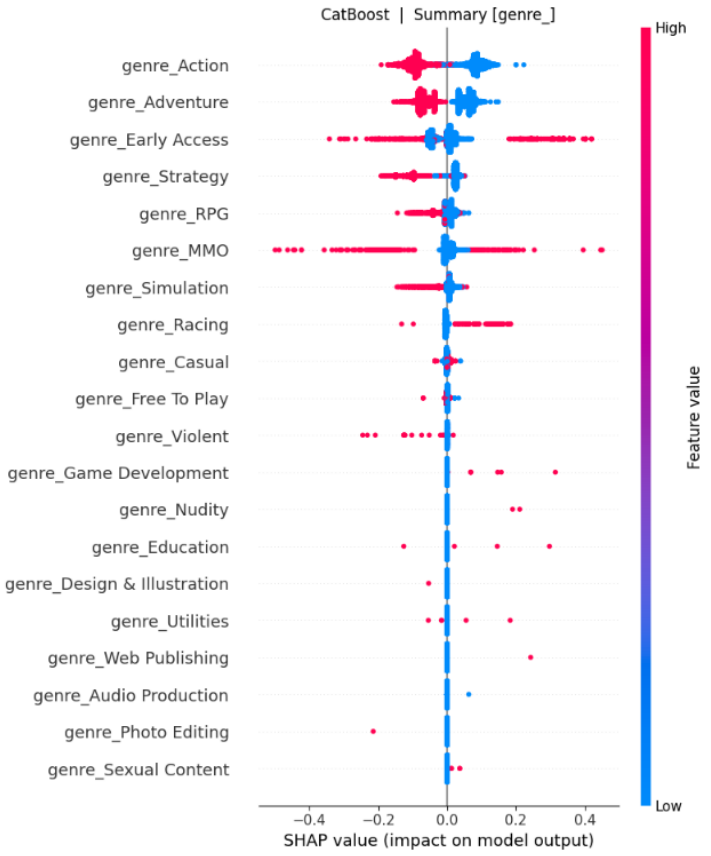


Figure 2 - Summary plot (Genre) of CatBoost

Strategy and RPG (Role-playing games) follow similar patterns to each other, with both features revealing a mainly negative relationship with the likelihood of success.

MMO experiences a similar distribution of outcome impact to Early Access, as there is not a clear direction for the relationship between this feature and the likelihood of success of a game.

Simulation, when present, also displays a negative relationship, similar to that of Strategy and RPG.

Racing stands out as a genre that has a positive relationship with the likelihood of success.

The category sub-plot, shown in Figure 3, allows us to see more features. In-App Purchases, surprisingly, is another category which displays a positive relationship with the target variable, although it has a moderate SHAP value spread.

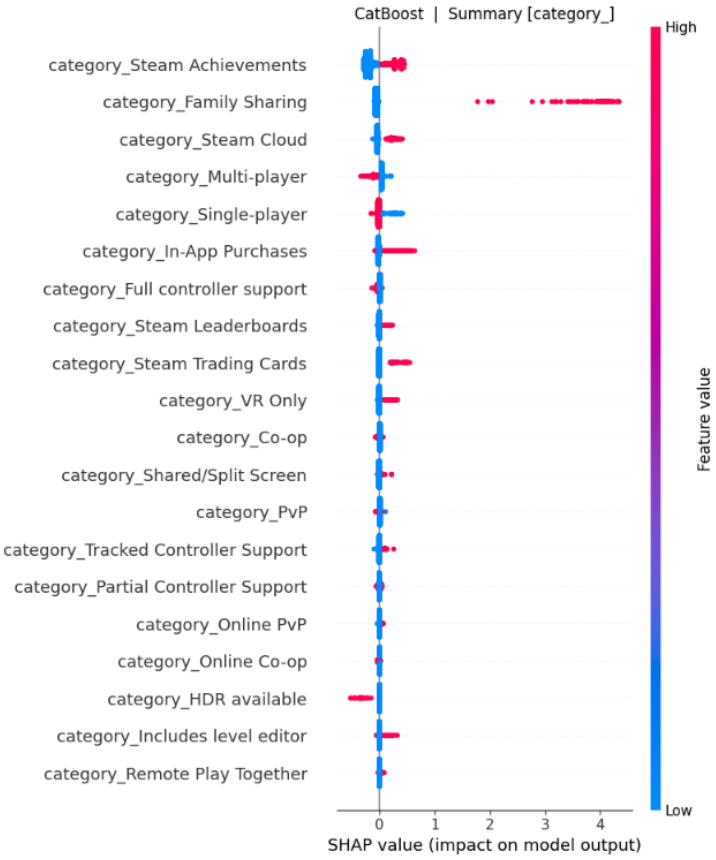


Figure 3 - Summary plot (Category) of CatBoost

In summary, the feature *is_free* is shown to be the greatest in feature importance. Among the five most relevant features, Steam-related features from the category space are prominently represented, with genre-related features becoming more prominent below that threshold. Across genres, a pattern emerges, where most of them appear to have a negative relationship with the likelihood of success of a game.

### *4.3 Discussion*

The previous subchapter was able to answer our question: "What are the main characteristics of a successful indie game?". If an indie game is free, and if it is being released on Steam, having Steam-related features available such as Cloud, Achievements and Family Sharing, seem to contribute positively to the likelihood of success of a game.

The genres of a game are more prominent further down in importance ranking, from the perspective of the model. Nevertheless, some of Ullmann et al.'s (2022) research, mentioned in the literature review and referring to all games, was applicable to indie games. These findings support their claims that Action, Adventure and Strategy may negatively affect the rating of games in the indie games' market segment. However, this study's findings disagree that RPG and Simulation games are more prone to success as the literature states, as there seems to be a negative relationship between these features and the likelihood of success. As for the Platform and Shooter games mentioned by Ullmann et al. (2022), these genres were not present in the final dataset, and so, their claims regarding them could not be verified. These authors also mentioned that multi-player features were more common in high-rated games than low-rated ones, which is not the case here, as was noted earlier, that the multi-player feature had a negative relationship with the likelihood of success. As for their assertion regarding 3D games, with this study, it was not possible to confirm it, as there is no data regarding 3D games in the genre or category datasets that were used in this study.

The literature review also mentioned game design, and good user and social interactions as important factors in game development. The data used has no means to directly evaluate this, but a proxy for social interactions in multi-player games could be the PvP (Player versus Player) and Co-op (Cooperative) tags, both online and not. However, these tags were very low on the feature importance axis, as can be seen in Figures 3 and 6 (available in the Appendices), and yield unclear information regarding their relationship with the target variable (Figure 3).

The fact that the *is_free* feature has a positive relationship with the target variable "success" is understandable, perhaps due to the lowered expectations of game quality when it comes to indie games. However, In-App Purchases having a positive relationship with the likelihood of success had not been anticipated. A plausible explanation could be

that when a game is free, it is common for there to be paid transactions after downloading the game. Since being free is such a benefit for a game, perhaps the positive signal for In-App Purchases is a result from this relationship between these features.

This study has possible limitations. As was stated before, this work was done under the assumption that the data, previously scraped, have not been altered and are not incorrect. If these assumptions prove to be incorrect, it would, evidently, affect the results of the research. Secondly, the threshold decided for a "successful" game was 5, as it is the midway point in the classification system. This is not a strict limit, and includes results closer to 5 as well, which may not be a desirable result. However, the trade-off to increasing this threshold is the reduction of the "Success" class, which currently corresponds to 20% of the dataset. Finally, given that the dataset is from 2024, there may have been changes in game scores and in which features are considered more attractive in an indie game since then.

In potential follow-up studies, it could be interesting to use the remainder of the dataset: the Tags file contains more tags to be analysed and using the summaries of games to identify any keywords that might make the games pop out more could prove to be useful. Methodologically speaking, using other statistical models may give different insights, and treating the problem as a regression rather than classification could be interesting.

# 5. CONCLUSIONS

In this study, the aim was to answer the question of "What are the main characteristics of a successful indie game?", in the hope of revealing market insights that future indie game developers can use, to guide their expectations and understanding of the industry.

As a base for the research, a dataset extracted from Steam in 2024 was analysed. A classification problem was created, where an indie game was successful when its given review score was above 5, on a scale from 1 to 10.

The models used were the Logistic Regression as a benchmark, XGBoost, LightGBM and CatBoost. The best-performing model was CatBoost, when applied to a one-hot encoded dataframe, although the occurring margins comparing to the other models were quite small. From this model, TreeSHAP was applied, and it is from this tool that we have derived our conclusions.

The feature representing whether a game is free or not displayed a positive relationship with the target variable "Success". If a developer is planning to release their game on Steam, categories mentioning Steam-related features were highly ranked in the model's feature importance. While the genre itself seemed to have a slight lower impact on the model's classifications compared to a game's category, there was a noticeable pattern of negative relationships between genres and the likelihood of success of a game, as demonstrated by the features Action, Adventure, and Strategy. These findings were supported by the literature that was not exclusive to indie games, although some others were not, as was the case of RPG and Simulation games, which seem to follow the same pattern described above. Some of the literature was unable to be verified due to lack of data in the selected datasets.

Further research could include a more in-depth analysis of the complete dataset, possibly branching out to keyword analysis of the summaries of games, and testing other statistical models.

## REFERENCES

Bond, M., & Beale, R. (2009). What makes a good game? Using reviews to inform design. *Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology*, 418–422. https://dl.acm.org/doi/10.5555/1671011.1671065

Buijsman, M. (2025, September 9). *Global games market to hit $189 billion in 2025 as growth shifts to console*. Retrieved November 14 2025. https://newzoo.com/resources/blog/global-games-market-to-hit-189-billion-in-2025

Buijsman, M., Rosier, E., Gu, T., Şahin, M., Kuzuhara, T., Kulijewicz, M., Wagner, M., Mitu, H., Hunt, B., Zhou, D., Reis, T., & Anema, I. (2025). *Global Games Market Report*.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *13-17-August-2016*, 785–794. https://doi.org/10.1145/2939672.2939785

Cox, D. R. (1958). The Regression Analysis of Binary Sequences. In *Source: Journal of the Royal Statistical Society. Series B (Methodological)* (Vol. 20, Number 2). https://doi.org/10.1111/j.2517-6161.1958.tb00292.x

Josef, A., Van Lepp, A., & Carper, M. (2022). *The Business of Indie Games; Everything You Need to Know to Conquer the Indie Games Industry*. CRC Press. https://doi.org/10.1201/9781003215264

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems 30* (pp. 3146–3154).

Keijser, R. (2012). *Popularity of Indie Games*.

Keogh, B. (2019). From aggressively formalised to intensely in/formalised: accounting for a wider range of videogame development practices. *Creative Industries Journal*, *12*(1), 14–33. https://doi.org/10.1080/17510694.2018.1532760

Klézl, V., & Kelly, S. (2022, September). Digital transformation in the video game industry: Exploring indie developers' perspective. *38th Annual IMP Conference*.

Lundberg, S. M., Erion, G. G., & Lee, S.-I. (2019). *Consistent Individualized Feature Attribution for Tree Ensembles*. https://doi.org/0.48550/arXiv.1802.03888

Lundberg, S. M., & Lee, S.-I. (2017). *A Unified Approach to Interpreting Model Predictions*. https://doi.org/10.48550/arXiv.1705.07874

Machado, M. R., Karray, S., & de Sousa, I. T. (2019). LightGBM: an Effective Decision Tree Gradient Boosting Method to Predict Customer Loyalty in the Finance Industry. *14th International Conference on Computer Science & Education (ICCSE)*, 1111–1116. https://doi.org/10.1109/ICCSE.2019.8845529

Mäyrä, F., & Alha, K. (2020). "Mobile Gaming." In *The Video Game Debate 2: Revisiting the Physical, Social, and Psychological Effects of Video Games* (pp. 107–121). Routledge. https://doi.org/10.4324/9780429351815

McCarthy, R. V., McCarthy, M. M., Ceccucci, W., & Halawi, L. (2019). Predictive Models Using Decision Trees. In *Applying Predictive Analytics* (pp. 123–144). Springer International Publishing. https://doi.org/10.1007/978-3-030-14038-0_5

Miller, J. (2014, September 15). *Microsoft pays $2.5bn for Minecraft maker Mojang - BBC News*. Retrieved November 14 2025. https://www.bbc.com/news/technology-29204518

Molnar, C. (2025a). *17 Shapley Values – Interpretable Machine Learning*. Retrieved April 10 2026. https://christophm.github.io/interpretable-ml-book/shapley.html

Molnar, C. (2025b). *18 SHAP – Interpretable Machine Learning*. Retrieved April 10 2026. https://christophm.github.io/interpretable-ml-book/shap.html

NewbieIndieGameDev. (2024, October). *NewbieIndieGameDev/steam-insights: Database of Steam video game data from October 2024, including game details, genres, reviews, tags, and SteamSpy insights*. Retrieved November 22 2025. https://github.com/NewbieIndieGameDev/steam-insights

Pashkov, S. (2021). *Video Game Industry Market Analysis: Approaches that resulted in industry success and high demand*.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. *Neural Information Processing Systems*. https://doi.org/10.48550/arXiv.1706.09516

Semenchenko, N., Melnychuk, V., Zaporozhchenko, A., & Molchanov, M. (2025). VIDEO GAME INDUSTRY AS A PART OF CREATIVE AND DIGITAL ECONOMICS. *Norwegian Journal of Development of the International Science*, (155), 51–54. https://doi.org/10.5281/zenodo.15298383

Shapley, L. S. (1953). A Value for n-Person Games. In *Theory of Games: II* (17). https://doi.org/10.1515/9781400881970-018

Steam. (2021). *Steam Support :: Steam Cloud*. Retrieved May 13 2026. https://help.steampowered.com/en/faqs/view/68D2-35AB-09A9-7678

Steam. (2024, September 11). *Steam News - Steam Families is here - Steam News*. Retrieved May 13 2026. https://store.steampowered.com/news/app/593110/view/4605582245626919823

Stoll, J. (2025a, February 5). *Indie gaming - statistics & facts | Statista*. Retrieved November 13 2025. https://www.statista.com/topics/13173/indie-gaming/#topicOverview

Stoll, J. (2025b, December). *Mobile gaming market worldwide - Statistics & Facts | Statista*. Retrieved November 13 2025. https://www.statista.com/topics/7950/mobile-gaming-market-worldwide/

Ullmann, G. C., Politowski, C., Gueheneuc, Y. G., & Petrillo, F. (2022). What Makes a Game High-rated? Towards Factors of Video Game Success. *Proceedings - 6th International ICSE Workshop on Games and Software Engineering: Engineering Fun, Inspiration, and Motivation, GAS 2022*, 16–23. https://doi.org/10.1145/3524494.3527628

Verma, V. (2024). Exploring Key XGBoost Hyperparameters: A Study on Optimal Search Spaces and Practical Recommendations for Regression and Classification. *International Journal of All Research Education and Scientific Methods (IJARESM)*, *12*(10), 3259–3266.

VGI. (2024). *Steam Indie Games Market Today Indie Market Segments Indie Market Maturity Overview of the VGI Global Indie Market Report 2024 2*.

APPENDICES

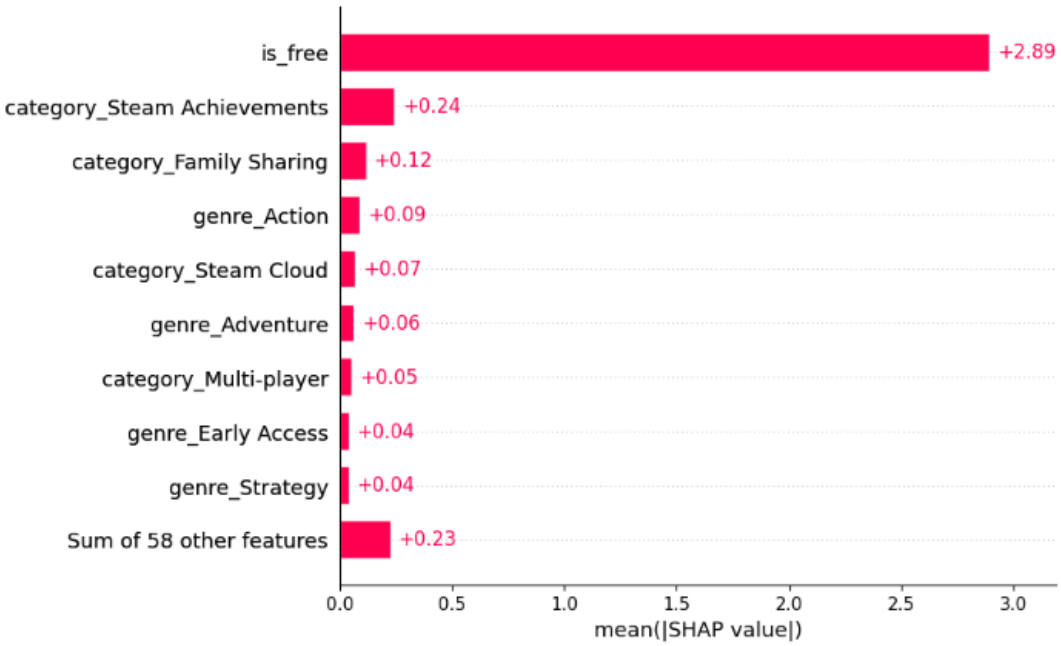


Figure 4 - Bar Plot of Feature Importance of CatBoost

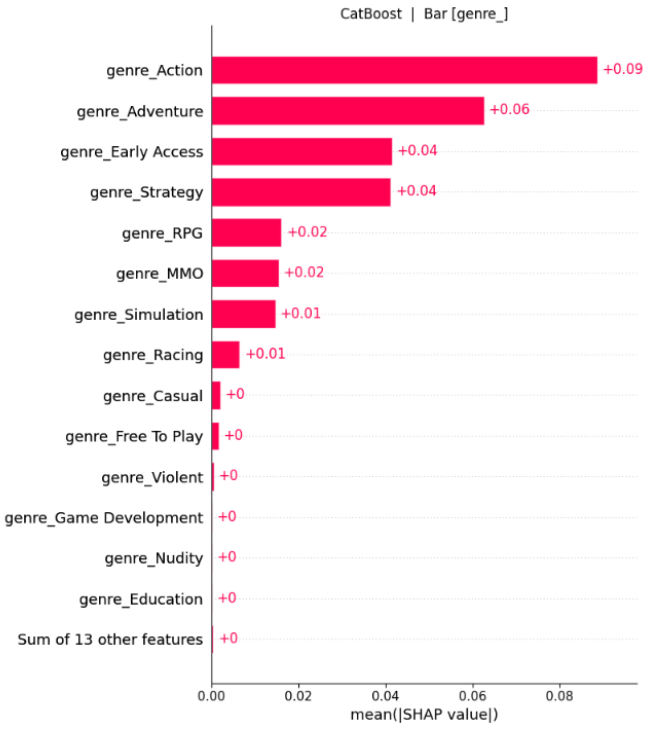


Figure 5 - Bar Plot of Feature Importance (Genre) of CatBoost

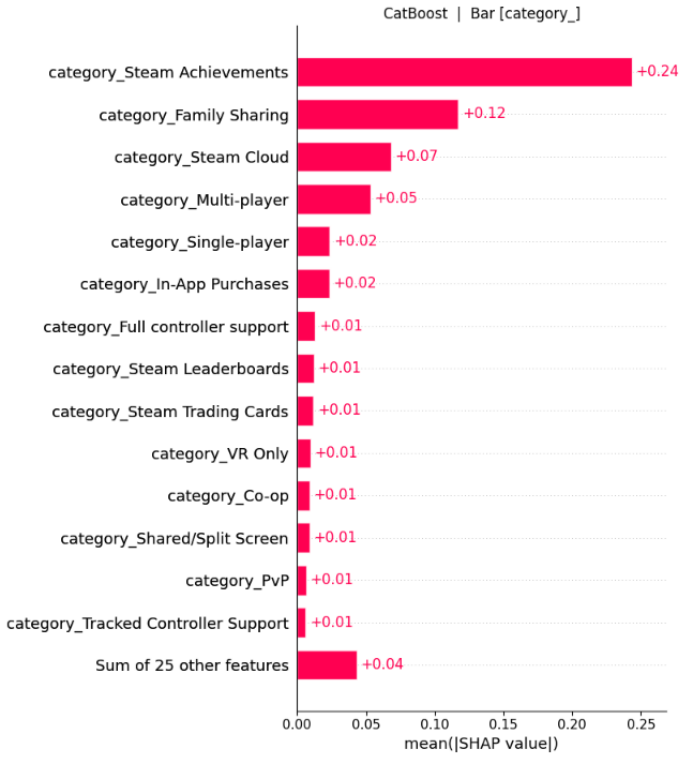


Figure 6 - Bar Plot of Feature Importance (Category) of CatBoost