Data Science Group Project

Group 6

Areebah hassan, jae loh, linghui zeng, ruixuan zhang, yihao dang

# Dataset Selection and Justification

## Appropriate Dataset Chosen

For this project, the Steam Games dataset from Kaggle was selected, focusing specifically on paid games. This dataset is comprehensive, containing detailed information on various attributes of paid games available on the Steam platform, including but not limited to game prices, user reviews, recommendations, DLC counts, and required age ratings. The dataset's diversity in features allows for a robust analysis of factors that contribute to game popularity and user engagement on Steam.

Key attributes of the dataset include:

* ***Game Prices****:* Detailed price information, enabling an analysis of how pricing strategies affect game recommendations and sales.
* ***User Reviews and Recommendations****: Provides* insights into user satisfaction and game quality, crucial for understanding the factors driving game popularity.
* ***DLC Counts****:* Information on the number of downloadable content packs available for each game, which can impact user engagement and long-term revenue.
* ***Required Age Ratings***: Age restrictions for games, essential for market segmentation and understanding the target demographics.
* ***Platform Support***: Details on platform compatibility (Windows, Mac, Linux), helping to analyze the influence of platform support on game recommendations.

This dataset was chosen because it encapsulates a wide range of factors that can influence the success of paid games on a major gaming platform, making it ideal for a detailed exploratory data analysis and the development of predictive models.

# Problem Statement and Objectives

## Problem Statement

The primary problem addressed in this project is: ***‘Identifying key factors that contribute to the growth, engagement, and popularity of paid games on the Steam platform’.***

Given the vast number of paid games available, understanding these factors is crucial for developers and publishers to optimize their strategies and maximize their game's visibility and success.

## Objectives

* ***Identify Key Factors Influencing Game Recommendations:*** Analyze the relationship between game prices, user reviews, and recommendations to determine how pricing and user feedback influence game popularity.
* ***Understand the Impact of DLC and Age Ratings:*** Investigate how the number of DLCs and age ratings affect user engagement and recommendations, providing insights into content strategy and target demographics.
* ***Evaluate the Role of Platform Support and Localization:*** Examine the influence of platform support (Windows, Mac, Linux) and language localization on game recommendations, helping developers optimize their game releases for broader audiences.
* ***Develop Predictive Models for Game Recommendations:*** Use the insights gained from the exploratory data analysis to develop predictive models that can forecast game recommendations based on various features, aiding in strategic decision-making for game development and marketing.

By addressing these objectives, the project aims to provide actionable insights for game developers and publishers, helping them to enhance their game development, marketing, and engagement strategies on the Steam platform. The focus on paid games ensures that the analysis is relevant to understanding the dynamics of games that generate direct revenue, offering more targeted and financially impactful insights.

# Hypothesis

## Hypothesis

***‘Games with higher prices, made by experienced developers, and more DLCs, will have significantly higher recommendations on the Steam platform’.***

This hypothesis assumes that higher-priced games are often associated with better quality and more extensive content, which can lead to higher user satisfaction and, consequently, more recommendations. Additionally, games with more downloadable content (DLCs) are likely to engage users for longer periods, enhancing their overall experience and leading to more positive reviews and recommendations. User reviews serve as a direct indicator of game quality and user satisfaction; thus, games with better reviews are expected to be recommended more frequently. By testing this hypothesis, we aim to validate whether these factors are indeed significant predictors of game recommendations, providing valuable insights for developers and publishers to optimize their strategies.

# Data Preparation

In the initial phase of data preparation, we conducted a thorough examination of the dataset to check for missing values and data types to understand the completeness and consistency of the data. This preliminary check helped in identifying areas that required data cleaning and transformation. One of the first transformations applied was converting date columns from object type to datetime format. This conversion was necessary for accurate time-based analysis and ensured consistency in handling date-related data.

We then performed a review of the number of statistics for different items in the price column, which involved counting the occurrences for each price value. This step was critical in understanding the distribution of price data. Following this, we dropped columns with a very high percentage of missing values as they provided limited useful information. The columns removed included score\_rank, reviews, notes, websites, support\_url, support\_email, about\_the\_game, detailed\_description, and short\_description. These columns were deemed not valuable for our analysis due to their high rate of missing data.

Additionally, columns that were irrelevant to our analysis were removed. This included header\_image, screenshots, movies, and tags. The user\_score column was also dropped since most of its values were zero, making it uninformative for our purposes. After dropping these unnecessary columns, we created a new DataFrame to work with a more streamlined and relevant set of data.

Further data cleaning involved handling missing values in specific columns. For instance, rows with three missing values in the name column were dropped. We also addressed duplicate rows by checking for duplicates based on AppID and game names. This process revealed 636 duplicate names, which were then removed, keeping only the first occurrence of each duplicate set. Sorting the data by recommendations in descending order, or by release\_date if recommendations were the same, ensured that the most relevant data was prioritized.

We took care to exclude unofficially released games using regular expressions (regex) to identify patterns included in a special\_game\_token. Rows where the game name included any word from this token were filtered out. This step helped in maintaining the integrity of the dataset by ensuring only officially released games were analyzed. The cleaned DataFrame was then verified to be free of duplicate names and irrelevant columns.

For columns with list inputs such as supported\_languages, full\_audio\_languages, developers, publishers, categories, and genres, we converted these from string format to actual lists. This transformation was essential for accurate analysis of these categorical variables. We also created counts for supported languages and full audio languages, providing a quantitative measure of these features.

To handle columns with empty lists, we replaced these with the value Unknown, ensuring no data was left ambiguous. A new target column was created to relate to recommendations: games with zero recommendations were marked as 0, while those with recommendations greater than zero were marked as 1. This categorization allowed us to distinguish between played and unplayed games.

Finally, to focus on the core business problem, we filtered the dataset to analyze only paid games by removing rows where the price was zero. This filtering ensured that our analysis was relevant to games that were monetarily significant. The final DataFrame was thus prepared, ready for further analysis and model building.

# Exploratory Data Analysis

## Initial Descriptive Analysis

The initial examination of the recommendation data reveals substantial insights into the variability and distribution of game recommendations. The average number of recommendations per game is approximately 4,634, indicating a generally favorable reception across the platform. However, this average is accompanied by a high standard deviation of approximately 29,757.73, signifying a considerable range in the number of recommendations games receive. This variability suggests that while some games garner few recommendations, a select few achieve significantly higher numbers.

A closer look at the data highlights a right-skewed distribution, as evidenced by the large disparity between the median (411) and the mean (4,634). This skewness indicates that a small number of games are exceptionally popular, driving the mean upwards. The presence of outliers is further underscored by the maximum recommendation count of 1,088,708, demonstrating that certain games are overwhelmingly favored compared to the rest. The histogram of the recommendations supports this, showing a pronounced right skew, with most games receiving relatively few recommendations and a minority receiving an extremely high count. The standard deviation's magnitude underscores the substantial variability, and the range between the maximum and minimum values confirms the wide distribution spread.

## Log Transformation

To mitigate the skewness and high variability in the data, a log transformation was applied to the recommendation counts. This transformation significantly improved the data's distribution, making it more suitable for further analysis and modeling. The resulting box plot of the log-transformed recommendations exhibits a more symmetric distribution, indicating a more balanced spread of values.

The log transformation also reduced the impact of outliers. Although outliers are still present, their influence is less pronounced compared to the raw data, which helps in minimizing their skewing effect on the overall analysis. The histogram of the log-transformed data reveals a more normalized distribution, with a less pronounced peak and a more gradual tail-off. This normalization brings the mean and median closer together, indicating a reduction in skewness. Additionally, the standard deviation is lowered, reflecting decreased variability. The narrower range of minimum and maximum values further enhances the data's interpretability and ease of analysis.

## Pairplot Findings and Multicollinearity Analysis

The pairplot analysis provides a comprehensive visual representation of the relationships between multiple pairs of features within the dataset. One notable observation from the pairplot is the presence of skewness in certain features, particularly in the distribution of recommendations and playtime metrics. This skewness is characterized by long tails, indicating that a few games have exceptionally high values, which could significantly impact the overall analysis.

### Key Relationships

* Playtime Metrics: The pairplot reveals a strong linear relationship between average playtime forever and median playtime forever. This indicates that games with higher median playtime also tend to have higher average playtime, suggesting consistent player engagement for these games. This consistency highlights their sustained popularity and user interest.
* Recommendations and Positive Ratings: Another important finding is the positive linear relationship between log recommendations and log-positive ratings. Games that receive more positive ratings are likely to receive more recommendations, reinforcing the importance of user satisfaction in driving word-of-mouth promotion.
* Price and Engagement: Upon examination of the scatterplot for price and average playtime, it suggests that moderately priced games tend to have higher engagement compared to very cheap or very expensive games, providing valuable insights for optimizing pricing strategies to enhance player engagement and satisfaction.

### Clusters and Outliers

* Game Segments: The analysis of DLC count versus playtime metrics reveals distinct clusters. Games with more DLCs tend to have higher average playtime, indicating that additional content keeps players engaged for longer periods. This clustering can be useful for segmentation analysis, allowing for more targeted marketing and development efforts.
* Extreme Values: Outliers are identified in the scatter plot between peak concurrent users and other metrics. These outliers, representing blockbuster titles with significant popularity, need to be handled appropriately to ensure robust and reliable analysis outcomes.
* Uncorrelated Features: Some features, such as the relationship between price and recommendations, show no strong correlation. This indicates that these features may not be directly related and might require further transformation or could be deprioritized in model building.

### Multicollinearity

#### Average Playtime vs. Median Playtime (for both forever and 2 weeks): A tight linear pattern between these two features suggests a high degree of multicollinearity. Both features measure playtime, albeit in different ways, and likely move together.

#### Positive Ratings vs. Recommendations: A strong positive linear relationship indicates that games receiving more positive ratings are also likely to be more recommended, leading to potential redundancy if both features are used in a model.

#### Price vs. Playtime Metrics: Any clear linear relationship between price and playtime metrics (e.g., average\_playtime\_forever or median\_playtime\_forever) can indicate multicollinearity. Although price and playtime might not be directly collinear, pricing strategies can influence engagement, leading to potential collinearity with certain transformations.

#### Peak Concurrent Users vs. Engagement Metrics: A strong linear relationship between peak concurrent users and other engagement metrics like average\_playtime\_forever indicates multicollinearity. High peak concurrent users often correlate with higher average playtime, as both are measures of game popularity and engagement.

## Correlation Matrix Findings

The correlation matrix reveals significant positive correlations between several key features. The number of positive reviews is almost perfectly correlated with the number of recommendations, suggesting that games with more positive reviews are highly likely to be recommended more often. Including both in a model could introduce multicollinearity. Similarly, the total number of reviews is also highly correlated with the number of recommendations, reinforcing the importance of user feedback.

Moderate positive correlations are observed between log recommendations and positive ratings, peak concurrent users and positive ratings, and price and log recommendations. These correlations indicate that as the number of positive reviews increases, the log-transformed number of recommendations also increases, though not as strongly as the raw counts. Games with higher peak concurrent users tend to have more positive reviews, which could be due to the popularity and engagement of the game. Higher-priced games tend to get more recommendations, but the relationship is not very strong.

In conclusion, the correlation matrix findings and multicollinearity analysis provide valuable insights into the relationships and interactions between features within the dataset. These insights can inform strategic decisions for feature engineering, model improvement, and targeted marketing efforts, ultimately contributing to a more robust and insightful analysis.

## Analysis Between Recommendations and Various Other Features

### Estimated Owners and Recommendations

The analysis of recommendations versus estimated owners reveals significant insights into game popularity and user engagement on the Steam platform. There is a clear positive correlation between the number of estimated owners and the number of recommendations a game receives. This relationship is evident in both the raw and log-transformed data, indicating that as the number of owners increases, so does the number of recommendations.

#### Observations from Raw Recommendations Data

Games with higher estimated owners tend to have more recommendations. This trend is consistent across all owner ranges, with the median number of recommendations increasing proportionally to the owner range. However, there is considerable variability in recommendations for games with a large number of owners. Some games with many owners receive an exceptionally high number of recommendations, while others do not perform as well. Additionally, extreme outliers exist, especially in higher owner ranges, where a few games receive disproportionately high recommendations compared to others in the same category.

#### Insights from Log-Transformed Recommendations

Applying log transformation to the recommendation data normalizes it, reducing skewness and compressing the range of recommendations. The log-transformed data reveals a more distinct and linear increase in recommendations with more owners, making it clearer that each increase in the owner range generally corresponds to an increase in the median log recommendations. Furthermore, the log transformation reduces the impact of outliers, making the core distribution within each owner range more apparent. The consistency within each owner range improves, facilitating easier comparison across different ranges and better understanding of how owner numbers impact recommendations.

#### Key Findings and Strategic Insights

The analysis highlights a strong, positive relationship between the number of estimated owners and the number of recommendations a game receives. Games with a large number of owners exhibit high variability in recommendations, suggesting that while owner count is a significant factor, it alone does not guarantee high recommendations. The log transformation benefits the analysis by providing a clearer view of the data, reducing skewness, and making the data more suitable for regression modeling and other analyses that assume normally distributed data.

Outliers, particularly in higher owner ranges, indicate exceptional cases of game popularity. These outliers offer valuable insights and should be investigated further to understand what drives their high recommendation counts.

For game developers, these findings suggest that increasing the owner base can significantly impact the number of recommendations. Targeted marketing and engagement strategies to boost owner numbers can lead to higher recommendations and greater visibility and popularity.

### Platform Support and Recommendations

The analysis of platform support versus recommendations reveals that the presence of Mac and Linux support does not significantly impact the number of recommendations a game receives. Both supported and unsupported categories show similar distributions in log recommendations. However, games supported on these platforms tend to achieve recommendations consistently, albeit there are fewer games. In contrast, all games with recommendations support Windows, highlighting Windows as a crucial platform for achieving recommendations on Steam. This suggests that developers aiming to maximize recommendations should ensure Windows compatibility. Supporting Mac and Linux platforms, while not drastically changing recommendation dynamics, can still benefit niche markets and broader accessibility.

#### Marketing and Development Focus

Developers should prioritize optimizing their games for Windows to maximize reach and recommendations. Additional support for Mac and Linux can be considered for incremental gains and to cater to specific user segments. Enhancing game features and engagement strategies regardless of platform support is crucial, as platform support alone does not significantly impact recommendations. Insights from high-recommendation games across platforms can help identify common features or strategies that could be applied universally.

### Language Support and Recommendations

Adding support for multiple languages, especially niche ones, can boost user engagement and recommendations. Developers should consider localization as a strategy to reach wider audiences. Supporting multiple languages, such as Danish and Czech, can lead to higher recommendations. Adding language support can target specific markets and enhance engagement. Games published by well-known or reputable publishers tend to receive higher recommendations due to better marketing, higher quality, or established fan bases. Collaboration with well-known publishers or building a strong brand reputation can positively impact game success.

### Genre Preferences and Game Design

Developing games in highly engaging genres like Massively Multiplayer and RPG can lead to higher recommendations. Understanding user preferences and genre trends can help in game development and marketing strategies. Certain genres like Massively Multiplayer, RPG, and Action are more likely to receive higher recommendations, aligning with user preferences and enhancing game popularity. There is no direct overlap between the top 10 games with the most achievements and the top 10 games with the most recommendations. While achievements can enhance engagement, they do not necessarily correlate with the highest player recommendations. Games with high recommendations are generally more popular and well-regarded, but they may not rely heavily on the number of achievements to maintain player interest.

#### Player Engagement Strategies

Games like LOGistICAL focus on providing many in-game tasks, appealing to completionists and players who enjoy collecting achievements. On the other hand, games like Tom Clancy's Rainbow Six® Siege focus on delivering high-quality core gameplay and maintaining active player communities, which drives recommendations. Developers can consider a balance between offering a rich achievement system and ensuring high-quality gameplay to maximize both player engagement and recommendations. Understanding what drives player satisfaction and community support is crucial for designing games that are both engaging and highly recommended.

The analysis of games with the most achievements and recommendations highlights different strategies for engaging players. Achievements can enhance player retention and engagement, while recommendations reflect overall player satisfaction and popularity. By balancing these elements, developers can create games that are both engaging and widely recommended, leading to greater success on platforms like Steam.

### Number of Supported Languages

English is the most frequently supported language by a significant margin, indicating its importance in the gaming industry for reaching a broad audience. Other popularly supported languages include German, French, Simplified Chinese, and Spanish (Spain), suggesting that games are localized to cater to major markets. The support for multiple languages indicates an effort by developers to reach diverse markets and improve accessibility for non-English speaking players.

#### Audio Languages

English is likely the most supported audio language, followed by other commonly spoken languages. However, the current graph's issues suggest incorrect parsing or splitting, showing individual letters instead of whole languages. The inclusion of audio languages further emphasizes the importance of localization in enhancing the gaming experience and accessibility.

### Comparative Insights

Most games support English text and audio, reflecting its global reach. Popular languages for text are also reflected in audio support, though not all text-supported languages may have corresponding audio support due to production costs and resource constraints. A wide range of languages supported in both text and audio suggests developers aim to maximize their game's reach and inclusivity, enhancing user experience across different regions.

Localization is a critical strategy in the gaming industry. By supporting multiple languages in both text and audio, developers can attract a diverse player base and improve user engagement. English remains dominant, but other languages also play a crucial role in reaching non-English speaking markets, ensuring that games are accessible and enjoyable for a global audience.

# Prepare Data for Building Model

## Data Extraction and Transformation

To prepare the data for model building, we initiated the process by creating a new DataFrame named steam\_model, using AppID as the unique identifier. This ensures that each game entry is uniquely identifiable and facilitates merging additional features in subsequent steps.

## Feature Engineering

# mention target variable and binary nature and how we made it

### Price

We first visualized the price distribution to understand its characteristics. The distribution revealed that the majority of games are priced below $50, with a few outliers priced significantly higher. Based on this distribution, we categorized the games into four price categories: cheap\_less\_than\_5 (games priced at or below $4.99), cheap\_4.99\_9.99 (games priced between $4.99 and $9.99), high\_9.99\_19.99 (games priced between $9.99 and $19.99), and expensive\_19.99\_above (games priced above $19.99). These categories were then encoded into the DataFrame using one-hot encoding.

### Required Age

We analyzed the required\_age distribution and observed that most games had no age restriction. We then scaled the required\_age variable using MinMaxScaler to standardize its range. This scaling helps in bringing all features to a comparable range, which is beneficial for many machine learning algorithms.

### DLC Count

The dlc\_count distribution was heavily skewed, indicating that most games had few or no downloadable content (DLC) items. To address this skewness, we applied a log transformation. This transformation reduces the impact of outliers and makes the data more normally distributed, which is advantageous for model performance.

### Platform Availability

We included platform availability indicators (Windows, Mac, Linux) in the model to account for the operating systems supported by each game. These binary indicators provide information about the game's accessibility across different platforms, which can influence user engagement and satisfaction.

### Achievements and Recommendations

Similar to DLC count, the achievements and recommendations distributions were skewed. We applied log transformations to these variables as well. Transforming these features helps in managing their skewness and makes them more suitable for modeling. Additionally, we created a binary classification for recommendations by setting a threshold at the 50th percentile, categorizing games into those with high and low recommendations.

### Game Categories

To capture the diversity of game types, we identified the top 20 game categories and used one-hot encoding to include these categories as features. This step allows us to represent the categorical information of game genres in a numerical format that can be easily used by machine learning algorithms.

### Experienced Developers

We identified experienced developers from a predefined list and added a corresponding feature to the model. This feature indicates whether a game was developed by a recognized, experienced developer. Including this feature helps in understanding the impact of developer experience on game success and user engagement.

### Supported Languages

Finally, we categorized the supported languages into four categories based on the number of languages supported by each game. We then applied one-hot encoding to include these categories as features. This categorization helps in understanding the accessibility and potential market reach of the games, which can be a significant factor in their success.

The final DataFrame, meticulously constructed through a series of extraction, transformation, and feature engineering steps, is now well-equipped for the subsequent modeling phase. This DataFrame integrates a diverse array of features that capture various aspects of the games, such as pricing strategies, platform availability, age requirements, downloadable content, achievements, recommendations, game categories, developer experience, and supported languages.

# Building Model

After preparing the dataset, the next crucial step involved building machine learning models to predict our target variable. Given the binary nature of our problem, we considered using classification algorithms such as Logistic Regression, K-Nearest Neighbors (KNN), and Naive Bayes for making predictions. These models are well-suited for binary classification problems and offer a balance between simplicity and effectiveness.

We began by importing the necessary libraries for these models, including LogisticRegression, KNeighborsClassifier, and GaussianNB from sklearn. Additionally, we imported various performance metrics like accuracy, precision, recall, and F1 score to evaluate our models comprehensively.

Before fitting the models, we performed a critical step of removing any duplicate columns from our final dataframe. This ensured that the models would not be misled by redundant data, which could potentially skew the results and reduce the accuracy of our predictions. Furthermore, we replaced spaces in column names with underscores to prevent any potential issues during model training, as some libraries may not handle spaces well in column names.

We then split the dataset into training and testing sets using a 70-30 split ratio. This division allowed us to train the models on a substantial portion of the data while reserving a separate set for evaluating the model's performance. The train\_test\_split function from sklearn was utilized for this purpose.

To prepare the data for modeling, we separated the features and the target variable. The features selected were price, log\_dlc\_count, log\_achievements, Single-player, \_Multi-player, \_Online\_PvP, \_Online\_Co-op, Experienced\_developers, supported\_languages\_low, and supported\_languages\_very\_high. These features were chosen based on their potential impact on the target variable and their relevance to the problem at hand. The target variable was binary\_output, which indicates the binary classification we aimed to predict.

# Model Evaluation

To evaluate the performance of each model, we used cross-validation with five folds. Cross-validation is a robust technique that helps ensure that the results are not dependent on a specific train-test split. By dividing the data into multiple folds and training the model on each fold, we can get a more accurate estimate of the model's performance. This approach helps in identifying overfitting and ensures that the model generalizes well to unseen data.

We calculated the cross-validation scores for each model and printed their mean values to determine the best-performing model. The initial results provided a comparative view of how each model performed across different folds, giving us insights into their stability and reliability.

The K-Nearest Neighbors (KNN) classifier was then further evaluated by testing different values of k to find the optimal number of neighbors. The choice of k in KNN significantly affects the model's performance, as it determines the number of nearest neighbors considered when making predictions. We plotted the accuracy scores against the k values to visualize the impact of different k values on the model's performance. This visualization helped us identify the best k value that provided the highest accuracy. After optimizing the k value, we observed an improvement in the model accuracy from 0.607 to 0.647.

Additionally, we examined the distribution of the required\_age\_scaled feature using histograms to understand its impact on the model. Visualizing the data distribution helps in identifying any potential outliers or skewness in the data, which could affect the model's predictions. By understanding the data distribution, we can make informed decisions about data preprocessing and feature engineering.

In summary, we built and evaluated three different classification models: Logistic Regression, K-Nearest Neighbors, and Naive Bayes. We used cross-validation to determine the best model and fine-tuned the KNN model to achieve optimal performance. The results highlighted the importance of careful feature selection, hyperparameter tuning, and data preprocessing in building effective machine learning models. The next steps involve further improving the model and addressing any remaining issues to enhance its predictive capabilities.

Linear regression didn’t work well

# Business Impact

The primary business problem addressed by our model is identifying key factors contributing to the growth, engagement, and popularity of games on the Steam platform. By understanding these factors, we can provide actionable insights to game developers, marketers, and platform administrators to enhance their strategic decisions, optimize their resources, and ultimately increase their revenue and market share.

## Key Insights and Their Implications

* ***Developer Experience:*** Our model incorporates an 'Experienced Developers' feature, which has shown a significant correlation with game success. This insight suggests that games developed by well-known and experienced developers tend to perform better. Game developers and publishers can leverage this information to form strategic partnerships and collaborations with experienced developers to enhance their game's credibility and potential success.
* ***Supported Languages:*** The analysis revealed that the number of supported languages is a crucial factor for a game's popularity. Games with higher language support cater to a broader audience, thus increasing their potential user base. Developers should consider investing in localization and language support to reach a more diverse and global audience.
* ***Game Features and Pricing:*** Features such as achievements, single-player mode, and price categories were also found to be influential. Understanding the optimal pricing strategy and feature set that resonates with the target audience can help developers and marketers position their games more effectively in the market. For instance, games with more achievements and a lower price point may attract more budget-conscious gamers looking for a fulfilling experience.

## Limitations

## ***Genre Specific Insights:*** The model also highlighted specific genres that have a higher impact on game recommendations and revenue. By focusing on genres with a higher growth potential, developers can tailor their game development to meet market demand more precisely.

We didny look at genres holistically, looked at first genre so it limited a holistic view of the data, but we did look at first genre

Hard to extract all genres from the list provided, it was a data type that wasn’t extractable , due to liitations of coding ability

## Business Case for Model Implementation

## Implementing this model provides several tangible benefits for game developers, publishers, and the Steam platform:

* ***Enhanced Decision-Making:*** The model offers data-driven insights that can significantly enhance decision-making processes. Developers and marketers can make informed choices about game features, pricing, and target audiences, leading to better product-market fit and higher chances of success.
* ***Revenue Growth:*** With better-targeted games and optimized features, developers can attract more users and increase engagement, leading to higher sales and in-game purchases. This directly contributes to revenue growth and profitability for both developers and the Steam platform.
* ***Competitive Advantage***: Leveraging advanced predictive models provides a competitive edge in the highly saturated gaming market. Companies that adopt these insights can stay ahead of the curve, adapting quickly to market trends and consumer preferences.

Search marketing and get rid of eveth

## Recommendations

## Based on the insights derived from the model, we propose the following recommendations to enhance business outcomes for game developers, publishers, and the Steam platform:

* ***Form Strategic Partnerships with Experienced Developers:*** Collaborate with well-known and experienced developers to boost game credibility and success potential. This strategy leverages the established reputation and expertise of seasoned developers, which our model has shown to significantly correlate with game popularity.
* ***Invest in Localization and Language Support:*** Expand language support to cater to a more diverse and global audience.
* ***Optimize Pricing Strategies:*** Use data-driven insights to develop pricing strategies that attract budget-conscious gamers while ensuring value for premium segments. Consider tiered pricing models that offer basic and premium versions of the game to cater to different market segments.
* ***Enhance Game Features Based on User Preferences:*** Focus on incorporating features that are highly valued by users, such as achievements and robust single-player modes. These features have shown a positive impact on game popularity and can significantly enhance user satisfaction and retention.
* ***Target High-Growth Genres:*** Prioritize game development efforts in genres that have demonstrated higher growth potential. This targeted approach ensures that development resources are focused on areas with the highest return on investment.
* ***Continuous Monitoring and Model Refinement***: Implement a process for continuous monitoring of game performance and user feedback. Regularly update the predictive model with new data to ensure it remains accurate and relevant, adapting to changing market trends and user preferences.
* ***Add something for genres target high growth genres, we weren’t able to give full picture but believe in that***