

C10: Predicting Steam Game Popularity

TEAM:

Karl Hannes Pallon

Lisbeth Lepp

Kusti Sammul

Link to Git: [khpallon/predicting-steam-game-popularity](https://github.com/khpallon/predicting-steam-game-popularity)

Business Understanding

Background

The video game industry is a rapidly growing market in which thousands of new titles are released each year. However, only a small portion of games achieve substantial popularity. For developers, publishers, and investors, predicting a game's success early, ideally before or at launch, is critical for resource allocation, marketing decisions, and financial risk assessment.

The goal of our project is to build a model that can predict a game's popularity based on the information available at launch. By understanding which characteristics correlate with success, stakeholders can make more informed decisions.

Business Objective

The core objective is to create a model that can predict a game's popularity using only launch-day metadata. This includes features such as:

- Publisher and developer
- Price
- Supported platforms
- Genre and categories

In addition, we have a table with monthly player count data over time for each title. This allows us to find patterns in the fluctuating player counts of games, and compare them to each other to define "popularity".

The business goals for this model are:

1. Help developers and publishers estimate potential interest in a game before significant costs.
2. Determine which price ranges tend to perform better.
3. Assist investors in identifying games with higher predicted popularity.
4. Allow studios to assess where their game lies in relation to current market trends.

Definition of "Popularity"

To create our model, we will define popularity based on changes in player counts. Suitable definitions include: average monthly players in the first 3 months after launch, changes in player gain per month.

The exact choice of metric will be based on usefulness and availability.

Success Criteria

The model will be considered successful if it:

- Enables stakeholders to understand **why** certain games are predicted to succeed.
- Enables earlier and more informed decision-making about game planning, pricing, or resource allocation.
- Is able to classify games as “likely to succeed” and “likely to fail” without overfitting to our dataset

In regards to the last point, we do not believe that 90% accuracy is viable for our model, so we will consider it successful if it helps guide informed decisions and filter the set of games with potential to a more manageable level.

Data Constraints

- Only data **available at launch** can be used for prediction, since we don’t have data on publicity stunts, word of mouth advertising, competing concurrent releases, budget, or other such factors.
- Since the data is gathered over many years, we expect to see numbers steadily go up as more gamers join the Steam platform, so we will likely have to deal with relative numbers instead of absolutes.
- Since the data is gathered from the Steam store, we will not have data on how games perform on other marketplaces, especially some free games that might have their own download pages entirely.
- Monthly user count data may include games with missing entries or irregular tracking.

Risks and Challenges

- **Imbalanced Popularity Distribution:** Most games have very low player counts; only a small percentage are highly successful. This can bias models toward predicting “low popularity” unless handled carefully.

- **Publisher/Developer Bias:** Large studios may inherently inflate popularity predictions due to reputation rather than actual game quality.
- **Model Drift:** Gaming trends change quickly; features that predict popularity today may not hold in future years.
- **Overfitting bias:** Predictive models can not take into account emerging new genres and innovations finding great success.

Data Understanding

1. Gathering data

In order to achieve our goals, we need data to work with.

1.1 Outline data requirements

We require data about games that have yet to gain popularity, games that have gained popularity, what their name is and what genre these games are classified as, who developed and published them, how much they're priced as or whether they're free, positive and negative reviews, currently active player count, supported languages and when popular games gained players. The data will be from June 2012 to September 2025. Data is formatted in CSV.

1.2 Verify data availability

For this model we have found a Kaggle dataset which contains data about the monthly average players in the Steam platform. While this data has info about player statistics, it does not provide us details about the games. Therefore we have included an alternative Kaggle data source which includes details about the Steam games. By confirming what data was available and what was missing, we have ensured that our project remained realistic and the model could be built with the resources we have found.

1.3 Define selection criteria

The relevant data is stored in 3 CSV-files. The tables in question are "steamcharts", which describes the game's popularity and player count each month, "steam_app_data" which contains game metadata and "id_name" where we get each game's name.

From the first table "steamcharts", we will use fields that contain information about the month in question, average players during said month, player gain percentage, peak player count during that month and ID of said game.

From table "steam_app_data" we will use fields that contain the ID of said game, age requirement for playing, whether it's free or not, what language options are available, developers, publishers, available platforms to play on, categories, genres and the release date of the game.

From table "id_name" we will use the game's ID and name.

Our case ranges include games that are only on Steam and which reviews are after 2015.

We gathered datasets consisting of games metadata, player count changes and names of all games that we collected data of. We checked the usability by importing the CSV file data into Jupyter Notebook and using Python's Pandas library, where we ran into no errors.

2. Describing data

Starting with our Game Metadata dataset, it contains info about the game's genre, developer, publisher, price and so on. The file is in CSV format making it easy to load into Jupyter Notebook using Python's Pandas library. It contains 6974 games. We will be using in total 10 columns of data. The data is structured and consistent.

Then we have Player Count dataset, which has info on average player gain or decrease each month with dates and game IDs. The format is also CSV. It contains 612265 rows of information. In total we will be using 5 columns. This time-series data is suitable for analyzing trends and finding spikes in popularity.

Lastly there is the Names dataset, that contains all of the videogame names. The format is yet again CSV. It contains 6938 game names. We will use both 2 columns being ID and the name of the game. The data is formatted cleanly and structured well.

The available data is suitable for identifying patterns that cause games to gain popularity. Together, these datasets cover the necessary fields and have a sufficient number of cases to perform meaningful statistical analysis.

3. Exploring data

The purpose of this exploration phase is to gain familiarity with the datasets, inspect the distributions of key variables, and identify potential data quality issues.

Game Metadata - Contains information such as genres, price, publisher and such. We observed that there are some metadata fields that contain null values that will require cleaning and also some duplicate data that might cause merging issues. Overall the metadata is rich and varied enough for modeling.

Player Count - Contains data such as average players, peak players and gain. Some games have missing months which may require filtering. There are also occasional large spikes in gain percentage, which may represent real events such as updates or promotions.

Overall, the datasets appear suitable for the data-mining goals. There is enough variability in price, genres, and player trends to develop meaningful predictive models. Time-series player data provides strong signals for popularity trends.

4. Verifying data quality

After examining the available datasets, we evaluated whether the data is complete, consistent and good enough for our goals. Overall the required fields exist, but with some minor issues such as missing metadata values. Despite this issue, most values are logically consistent and don't contain critical errors like negative player counts or impossible dates. The amount of missing data is manageable and can be addressed through filtering. We believe that the datasets are of adequate quality to support our project and the found issues don't prevent us from moving forward.

Project Plan

1. Gather data

We got our data by combining two Kaggle datasets (see Data Understanding above), and cleaned them.

2. Define “popularity”

As explained in Business Understanding, we need to define what our model will take as a success metric. We will do so by analysing patterns in our data about player counts and comparing specific games to those median values.

3. Find what correlates with popularity

We are interested in seeing what correlates with a successful game, such as how much influence does a publisher have, is there a price range that is most favored among players, etc..

4. Train a model for predicting a new game’s popularity

We will attempt to train a model that can predict which games become successful based on launch-day metadata. Our datasets will be split into test and train sets.

5. Evaluate our model

Since we are predicting a highly variable metric off of rather limited data, our model will likely have many issues that don’t reflect the real world. As the last step of our project, we will find those issues and reflect on ways to remedy them.