Predicting Average Playtime for Video Games using Machine Learning

Kamil Sabbagh, Mohammad Shahin, Rafik Hachana

June 25, 2024

Contents

1 Introduction

In an era dominated by digital entertainment, video games stand out as both culturally influential and economically significant. The ability to predict the average playtime of video games based on various attributes such as genre, release date, and platform availability can provide crucial insights for developers and publishers. This predictive capability can inform strategic decisions that optimize both the development resources and marketing efforts to align with player engagement patterns.

As the gaming industry evolves, leveraging data to understand user behavior becomes increasingly vital. With millions of active players and thousands of games, the industry is a rich field for applying machine learning to unearth patterns that are not immediately obvious. According to recent statistics, the video game market is expected to grow at a compounded annual growth rate of 8.76% in the next three years[1], highlighting the importance of targeted game development and marketing strategies to capitalize on this growth.

2 Business and Data Understanding

2.1 Problem Statement

The gaming industry, characterized by its rapid pace and high competition, demands continuous innovation and adaptability from developers and publishers. In this context, the ability to predict how long a game will be played—its average playtime—becomes a strategic asset. This project aims to harness machine learning to predict the average playtime of video games based on a dataset of game attributes. This capability will enable our company to tailor game development and marketing strategies more effectively, optimizing for player engagement and satisfaction.

[1] `https://www.statista.com/outlook/dmo/digital-media/video-games/worldwide`

2.2 Terminology

2.2.1 Business Terminology

- **Average Playtime**: The total playtime divided by the number of users for a specific game.
- **User Engagement**: A measure of the depth of a player's interaction and commitment to a game, often reflected by playtime.
- **Player Retention**: The ability of a game to retain players over a specified period.

2.2.2 ML Terminology

- **Regression Analysis**: A statistical process for estimating the relationships among variables. In this project, it refers to predicting the continuous outcome of average playtime.
- **Feature Importance**: A technique to identify which attributes most significantly impact the prediction model.
- **Model Validation**: The process of confirming that the machine learning model operates as expected on new data.

2.3 Scope of the ML Project

2.3.1 Background

The organization has recognized the need to enhance its understanding of game performance metrics, particularly through the lens of player engagement as measured by average playtime. This understanding is crucial in a landscape where games are becoming increasingly diverse and the audience's preferences more sophisticated.

2.3.2 Business Problem

The primary business problem is the suboptimal allocation of resources in game development and marketing due to a lack of predictive insight into game performance.

2.3.3 Business Objectives

The main objective is to enhance resource allocation efficiency by predicting game performance in terms of average playtime. This prediction will guide decision-making in both development and marketing phases.

2.3.4 ML Objectives

From a machine learning perspective, the goal is to develop a regression model that can predict the average playtime of games with high accuracy, leveraging historical data on game attributes and past performance metrics.

2.4 Success Criteria

2.4.1 Business Success Criteria

A successful implementation of the ML model will enable the company to increase the ROI on development and marketing by at least 15%, as measured by more strategically aligned resource allocation and improved player satisfaction.

2.4.2 ML Success Criteria

The ML model should achieve a minimum accuracy (R-squared value) of 80% on the test data, indicating strong predictive performance.

2.4.3 Economic Success Criteria

The project will be considered economically successful if it leads to a 10% reduction in sunk costs due to misaligned game development efforts within one year of implementation.

3 Data Collection and Quality Verification

3.1 Data Collection Report

The primary data source for this project is the "Steam Games" dataset available on Kaggle. The dataset contains attributes such as genre, release date, and total playtime across multiple platforms. The data was collected from

publicly available sources and consists of over 20,000 records, each representing a unique game.

3.2 Data Quality Verification

Initial data exploration revealed that the dataset is largely complete with some fields exhibiting missing values, particularly in game genres and release dates. The data quality verification process will involve assessing the extent of these missing values and determining the necessity of imputation strategies to address them.

3.3 Data Exploration

Based on the insights of the notebook, we need to perform many steps for cleaning the data. The data contains 380519 missing value. Some columns are not needed as they are and we may change their format to include them in the ML steps. For example, Support email column is not needed as a string, but it would be beneficial to have whether an app has a support email or not for the prediction of the rating. Same goes for Support url, Website.

What data features need to be cleaned:

Based on the insights of the previous cells, we need to perform many steps for cleaning the data. The data contains 380519 missing value. Some columns are not needed as they are and we may change their format to include them in the ML steps. For example, Support email column is not needed as a string, but it would be beneficial to have whether an app has a support email or not for the prediction of the rating. Same goes for Support url, Website.

We also need to transform some columns: About the game, Reviews, Notes are text that is describing the games. We may encode the text or extract features of the games depending on the text.

Genres, Tags, Categories, Supported languages, Full audio languages are given by a list. We have to collect all items of these columns, possibly remove some similar ones, and finally use each genre as a categorical feature with yes or no. List of strings.

Developers/Publishers are the companies behind the application. String.

Movies columns contains a link to a video showing some details of the game like the gameplay. It is possible to extract some features from the video

to help assess the game quality. However, the process is rather complicated. String.

Screenshots columns is a link to some images taken from the applications. It is possible to extract some features from the video to help assess the game quality. However, the process is rather complicated. String.

Metacritic is a website that aggregates reviews of films, television shows, music albums, video games, and formerly books. Metacritic url (stirng) of the application is also provided. We can use scrape the review from the website. Metacritic score (number) is also given.

ppID is a number to identify an application. Number.

Name is the name of the application. String.

Release date is a date of the format: Oct 21, 2008. Date.

Estimated owners is estimated number of people who have purchased the application (possibly for free). Number.

Peak CCU Peak Total number of users. Number.

Required age. Number.

Price. Number.

DLC count. Number.

Header image. Url.

User score the suggested target variable. Values range between 0 anf 100. Number.

Windows, Linux, Mac indicate whether the application can run on the given OS. Binary.

Negative, Positive are the number of negative and positive rating, respectively. Number.

Recommendations is the number of recommendations. Number.

Average playtime two weeks. Number.

Average playtime forever. Number.

Median playtime forever. Number.

Median playtime two weeks. Number.

Achievements is the the number of achievements inside the application. Number.

Score rank

When evaluating the usefulness of features for predicting the rating of a Steam app, consider whether the feature provides relevant information that can help the model understand the app's appeal, performance, and user satisfaction. Here's an analysis of each feature:

- AppID: Not beneficial. This is just an identifier with no predictive power.
- Name: Not beneficial. While the name might be catchy, it does not directly influence the rating.
- Release date: Beneficial. Older games may have more reviews and established ratings.
- Estimated owners: Beneficial. Higher ownership can indicate popularity and influence ratings.
- Peak CCU: Beneficial. Peak concurrent users can reflect a game's popularity and engagement.
- Required age: Beneficial. Age restrictions can influence the target audience and ratings.
- Price: Beneficial. Price can affect accessibility and perceived value, impacting ratings.
- DLC count: Beneficial. More DLCs can indicate ongoing support and content, potentially affecting ratings.
- About the game: Potentially beneficial. The description could provide insights into game features and quality.
- Supported languages: Beneficial. More language support can widen the audience and improve ratings.
- Full audio languages: Beneficial. Similar to supported languages, full audio in multiple languages can enhance user experience.
- Reviews: Beneficial. Reviews directly reflect user satisfaction and are likely correlated with ratings.
- Header image: Not beneficial. Visual appeal can be subjective and not directly related to ratings. Website: Not beneficial. The presence of a website doesn't directly affect ratings.
- Support url: Not beneficial. The support URL itself doesn't provide predictive value.

- Support email: Not beneficial. Similar to the support URL, it doesn't influence ratings directly.
- Windows: Beneficial. Platform availability can influence the user base and ratings.
- Mac: Beneficial. Same as Windows.
- Linux: Beneficial. Same as Windows.
- Metacritic score: Beneficial. An aggregated score from critics can be highly predictive of user ratings.
- Metacritic url: Not beneficial. The URL itself doesn't provide information on the rating.
- User score: Beneficial. Directly reflects user feedback.
- Positive: Beneficial. The number of positive reviews is directly related to ratings.
- Negative: Beneficial. The number of negative reviews is also directly related to ratings.
- Score rank: Beneficial. Ranking can indicate overall performance and user satisfaction.
- Achievements: Beneficial. More achievements can indicate a richer gameplay experience, influencing ratings.
- Recommendations: Beneficial. Recommendations are directly related to user satisfaction.
- Notes: Potentially beneficial. Notes might contain relevant information about user feedback or updates.
- Average playtime forever: Beneficial. High playtime can indicate user engagement and satisfaction.
- Average playtime two weeks: Beneficial. Recent playtime can indicate current user engagement.

- Median playtime forever: Beneficial. Similar to average playtime, but can provide additional insights into engagement distribution.
- Median playtime two weeks: Beneficial. Same as above, for recent engagement.
- Developers: Beneficial. Known developers with good reputations might influence ratings positively.
- Publishers: Beneficial. Similar to developers, reputable publishers can impact ratings.
- Categories: Beneficial. Categories can help identify the type of game and its appeal to different audiences.
- Genres: Beneficial. Genre information can help understand the target audience and preferences.
- Tags: Beneficial. Tags provide detailed descriptions that can reflect game features and user interest.
- Screenshots: Potentially beneficial. Screenshots can give an idea of game quality, though this is subjective.
- Movies: Potentially beneficial. Trailers and gameplay videos can influence user perceptions, though this is also subjective.

3.4 Data Requirements

Data Requirements: The data requirements for this project are defined to ensure that the dataset supports the predictive modeling effectively. We expect:

- Continuous features like playtime should have values within a plausible range, e.g., 0 to 500 hours.
- Discrete features like genre should be non-null and belong to a predefined list of categories.
- The format of the data should be consistent, with dates in YYYY-MM-DD format and numerical attributes in integer or float formats.

- The maximum allowable number of missing values in critical fields such as genre and playtime should not exceed 5% of the dataset.

These requirements are documented in a data schema that includes strict data types and conditions. This schema will guide the data cleaning process and ensure that the data used in the machine learning models is of high quality and relevance.

3.5 Data Quality Verification

Data Quality Verification Report: Upon verifying the data quality, several issues were identified:

- The dataset covers all necessary cases for the intended analysis, ensuring completeness.
- Some data errors were identified, particularly in the release dates of some games, but such errors occur in less than 2% of the dataset.
- Missing values were most common in the platform availability data, occurring in approximately 10% of the records. Strategies for handling these missing values include imputation based on the most frequent platform or exclusion of these records from specific analyses.

This verification confirms that while the dataset is largely robust, attention must be given to certain areas to improve data integrity before proceeding with further modeling.

3.6 Project Feasibility

The feasibility of the project has been assessed by examining the available resources, technological tools, and the timeline. The findings suggest that the project is feasible within the current constraints and available technologies. Tools and libraries such as Python's scikit-learn and TensorFlow will be employed, and the project timeline of 1 month is adequate for completing the modeling and validation phases.

3.7 Project Plan

Project Plan: The project plan outlines the key phases and milestones:

- **Phase 1: Data Preparation** (June 24th-June 30th): Data collection, cleaning, and preparation for analysis.
- **Phase 2: Model Development** (July 1st-July 7th): Development of the predictive model, including feature selection, model training, and preliminary testing.
- **Phase 3: Model Validation and Refinement** (July 8th - July 15th): Testing the model with new data, refining the model based on feedback.
- **Phase 4: Deployment and Monitoring** (July 16th - July 23rd): Deployment of the model into a production environment and monitoring its performance.

This structured approach ensures that each phase of the project is allocated sufficient time and resources, facilitating a smooth transition from data preparation to deployment.

References