Predicting Average Playtime for Video Games using Machine Learning

Kamil Sabbagh, Mohammad Shahin, Rafik Hachana

July 24, 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.

Our team comprises three members, each fulfilling a role necessary for this machine learning task: a machine learning engineer, a data scientist, and a data engineer.

- Data scientist's primary focus: Analyzing data to extract insights, build predictive models, and inform business decisions.
- Machine learning engineer primary focus: Designing, building, and deploying machine learning models and systems in production environments.
- Data engineer primary focus: Building and maintaining the data infrastructure and pipelines necessary for data analysis and machine learning.

Here is the distribution of these roles among our team members:

Name	**Role**
Rafik Hachana	Data Engineer
Mohammad Shahin	Data Scientist
Kamil Sabbagh	Machine Learning Engineer

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

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.

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 RMSE (root mean squared error) of 250 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.

2.5 Data quality verification

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 71,000 records, each representing a unique game.

2.5.1 Data Description

Initial data exploration revealed that the dataset is largely complete with some fields exhibiting missing values, particularly in game genres and languages. The data quality verification process will involve assessing the extent of these missing values and determining the necessity of imputation strategies to address them. Here we will list the fields of the dataset used along with surface-level properties about them.

The data contains 380519 missing values in total. The dataset has missing values in the following attributes: *Name* (1 missing values), *About the game* (2436 missing values), *Reviews* (62549 missing values), *Website* (36643 missing values), *Support url* (35462 missing values), *Support email* (11118 missing values), *Metacritic url* (67938 missing values), *Score rank* (71674 missing values), *Notes* (61274 missing values), *Developers* (2460 missing values), *Publishers* (2694 missing values), *Categories* (3407 missing values), *Genres* (2439 missing values), *Tags* (14014 missing values), *Screenshots* (1329 missing values), *Movies* (5048 missing values),

2.5.2 Data Exploration

Figure 1 depicts the distribution of games across different operating systems. The data aligns with expectations, showing that the majority of games are available on Windows, while a smaller number are available on Mac, and even fewer on Linux.

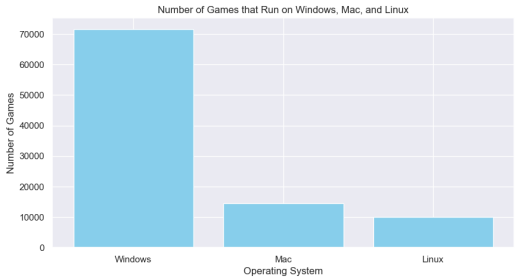

Figure 1: Games per OS

Some attributes in the data have the type of list. There two types of these attributes. A list of an attribute of the first type is a subset of a larger set, for example, *Supported languages* may include be English, French, etc. but the list of possible elements is finite and unique. For the other type, the values are not shared between records, and they can be random values. For example, the column *Movies*, is a list of URLs to trailers of the game separated by a comma. *Categories, Tags, Full audio languages, Supported languages* belong to the first type. To clean these data and transform them into appropriate format, we will apply something similar to one hot encoding. We get all the possible values of these columns and include each one of them as a new constructed binary column (e.g. a column of *English* representing whether English is present in *Supported languages*).

Figure 2 illustrates the correlation between the target variable, which is the average playtime of the game over a two-week period, and all other features. It is evident that the target variable is highly correlated with *Median playtime two weeks*. This feature, along with *Average playtime forever* and *Median playtime forever*, is removed during data transformation to prevent data leakage. Additionally, *Score rank* and *Metacritic score* show the highest correlations with the target, having correlation values of 0.12 and 0.11, respectively. Given the low linear correlation with the target, the task is complex and necessitates the use of various models to accurately predict the target variable.

2.6 Data Requirements

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

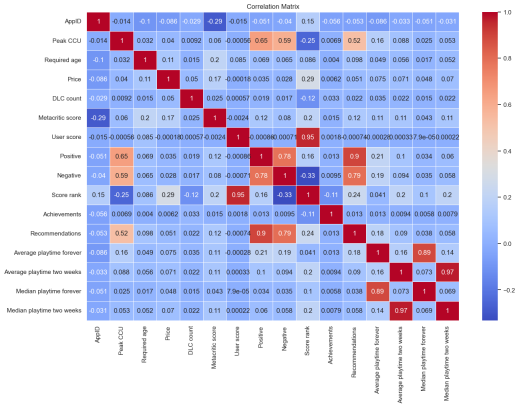

Figure 2: Correlation of the target with other attributes

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 %d, %m or YYYY %m.

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.

2.7 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.
- Values in the url and webiste columns were found to be invalid. Some of them contained some unicode symbols, and others were found to not

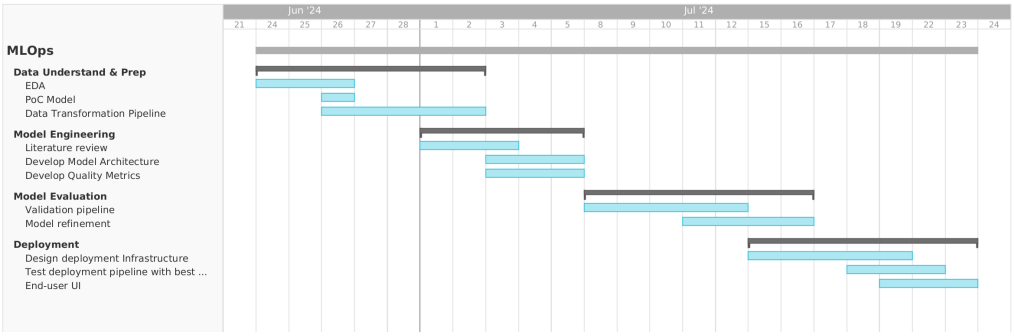

Figure 3: Gantt chart of the project plan

follow regular expressions for URLs. "http://null" is one invalid value found in the data quality verfication process.

- Supported languages also invalid list of languages. A valid value would be the string "['English", 'French']". One of the invalid samples found is "['English", 'French', K'ichie']". Making possible impossible without replacing K'ichie'.

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.

2.8 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.

2.9 Project Plan

The project plan outlines the key phases and milestones. The phase are described both in our Gantt chart (Figure 3) and in the bullet-points below:

- **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.

3 Data Preparation

The data preparation phase covers all activities to construct the final dataset (data that will be fed into the machine learning pipelines) from the initial raw data. Data preparation tasks are likely to be performed multiple times and not in any prescribed order. Tasks include table, record and attribute selection as well as transformation and cleaning of data for the modeling phase. The total number of columns after this phase is 610.

3.1 Select Data

Decide on the data to be used for analysis. Criteria include relevance to the machine learning goals, quality, and technical constraints such as limits on data volume or data types. Note that data selection covers selection of attributes (columns) as well as selection of records (rows) in a table. List the data to be included/excluded and the reasons for these decisions.

3.1.1 Included Data

The following data attributes have been selected for inclusion based on their relevance to the target variable (average playtime over two weeks) and their quality:

- **Price** - Directly affects purchasing decisions and playtime.
- **Required age** - May influence playtime patterns.
- **Release date** - Older games might have different playtime dynamics.
- **Metacritic score** - Indicator of game quality and potential playtime.
- **Achievements** - May correlate with engagement and playtime.
- **Peak CCU (Concurrent Users)** - Reflects game popularity and engagement.
- **User score** - Reflects player satisfaction which could impact playtime.
- **Positive reviews** - Indicates popularity and potential playtime.
- **Negative reviews** - May negatively impact playtime.

- **Recommendations** - Could indicate game's appeal and playtime.
- **Website** - Transformed to has_website.
- **Support URL** - Transformed to has_support_url.
- **Support email** - Transformed to has_support_email.
- **Metacritic URL** - Transformed to has_metacritic_url.
- **Categories** - One hot encoding (40).
- **Genres** - One hot encoding (33 unique values).
- **Tags** - One hot encoding (446 unique values).
- **Movies** - Transformed to num_movies (trailers).
- **Supported languages** - One hot encoding (134 unique values).
- **Full audio languages** - One hot encoding (121 unique values).
- **Estimated owners** - Transformed from format (min - max) to two numbers.

3.1.2 Excluded Data

The following data attributes have been excluded due to various reasons such as a high number of unique values, significant missing data, or irrelevance to the analysis:

- **Header image**
- **Score rank** - Has 71,674 null values out of 71,716.
- **Developers** - Too many unique values (42,615).
- **Publishers** - Too many unique values (36,815).
- **Screenshots**
- **AppID**
- **Name**

3.2 Clean Data

Raise the data quality to the level required by the selected machine learning techniques. This involves selection of clean subsets of the data, the insertion of suitable defaults or more ambitious techniques such as the estimation of missing data by modeling.

As discussed in a 2.6, the languages columns in the dataset had some issue related to having the "K'iche" instead of "Kiche". All string were replaced to ensure the validity of language columns.

Another problem we faced was the invalid URLs in URL and website columns. These URLs were considered non-existent. This was useful because for a column *Support url*, we're replacing it with the feature has_support_email. The same applies to all other URL based features.

3.3 Construct Data

This task includes constructive data preparation operations such as the production of derived attributes, entire new records, or transformed values for existing attributes. Derived attributes are new attributes that are constructed from one or more existing attributes in the same record.

- **Website** - Transformed to has_website.
- **Support URL** - Transformed to has_support_url.
- **Support email** - Transformed to has_support_email.
- **Metacritic URL** - Transformed to has_metacritic_url.
- **Categories** - One hot encoding and missing values filled (3407).
- **Genres** - One hot encoding and missing values filled (2439).
- **Tags** - One hot encoding and missing values filled.
- **Movies** - Transformed to num_movies (trailers).
- **Supported languages** - One hot encoding (134 unique values).
- **Full audio languages** - One hot encoding (121 unique values).
- **Estimated owners** - Transformed from format (min - max) to two numbers.

3.3.1 Derived Attributes

The following transformations have been applied to create new attributes from existing data:

- **Website** - Transformed to has_website.
- **Support URL** - Transformed to has_support_url.
- **Support email** - Transformed to has_support_email.
- **Metacritic URL** - Transformed to has_metacritic_url.
- **Categories** - One hot encoding and missing values filled (3407).
- **Genres** - One hot encoding and missing values filled (2439).
- **Tags** - One hot encoding and missing values filled.
- **Movies** - Transformed to num_movies (trailers).
- **Supported languages** - One hot encoding (134 unique values).
- **Full audio languages** - One hot encoding (121 unique values).
- **Estimated owners** - Transformed from format (min - max) to two numbers.

3.3.2 Target Variable

The target variable for this project is average playtime over two weeks. To avoid data leakage, only one of the following attributes is chosen and the rest are dropped:

- Average playtime forever
- Average playtime two weeks
- Median playtime forever
- Median playtime two weeks

3.4 Standardize data

This section outlines the normalization methods implemented to standardize the dataset. For binary columns, which contain only two values (0 or 1), no normalization is required. For all other columns, we utilize a standard scaler to ensure that each feature is given equal importance by the machine learning models.

4 Model engineering

In this section, we present our plan to use machine learning models to fulfills the goals defined in the first phase of this project. We first start with a quick literature search on similar tasks and problems, then definining some quality measures of the model (since we plan to train multiple models, and only select the best model at the end for deployment), and other details such as the usage of domain knowledge (in our case in the video game industry) and the criteria we have implemented to ensure that our results are reproducible.

4.1 Literature research

We have found a few research papers that tackle tasks similar to the one in our project. However, each one of the studies has a particular variant of the task, and we think that none of them is exactly the same as what we aim to do. So what we are shooting for is in a way novel, which makes the project a bit risky, and also prevents us from directly comparing the metrics of evaluation with the evaluation of other baseline model from the literature.

Among the studies that we have reviewed, the closest we found to ours is [4], with the only difference being that the paper has a classification task instead of the regression task that we have. [2] does something similar, but the task is even further from our because they run the experiment exclusively on mobile games, and also use the user data to predict the game time for each specific user (which is the opposite of what we do, since we start with the game attribute and want to predict the average playtime over all users for a given game). Other studies, like [3] and [1], focus on predicting the conversion rate or the churn rate of a given game, which is a different classification or regression task, but it provides similar insights at the level of the business.

Most studies use simple models like Random Forest classifiers, we aim to use a Deep Neural Network, aiming that its better expressiveness would yield better results than Decision Tree or linear models.

4.2 Model Quality Measures

The modeling strategy aims to balance multiple objectives, including performance, robustness, explainability, scalability, resource demand, and model complexity. The quality measures defined for this project are:

- **Performance Metric:** Mean Squared Error (MSE) is used to evaluate the accuracy of the model's predictions.
- **Robustness:** Assessed by evaluating the model's performance on different subsets of data and checking for variance in the results.
- **Explainability:** The ability of the model to provide understandable results is crucial, especially for stakeholders who need to trust the model's decisions.
- **Scalability:** The model's ability to handle increasing amounts of data and computation efficiently.
- **Resource Demand:** The computational resources required to train and deploy the model, including memory and processing power.
- **Model Complexity:** The number of parameters and the depth of the model, aiming to keep it as simple as possible without sacrificing performance.
- **Fairness:** Ensuring that the model's predictions are unbiased and fair across different demographic groups.

  These measures are weighted according to their importance in the specific application context, and models are ranked either by summing up the weighted quality measures to a scalar or by identifying Pareto optimal models in a multi-objective optimization process.

4.3 Model Selection

The model selection process begins with simpler models to serve as baselines and progressively increases in complexity. The models selected for this project are:

- **Neural Networks:** Specifically, two architectures, Model1 and Model2, have been developed:
  - **Model1:** A neural network with an input layer, two hidden layers (each with 64 neurons), batch normalization, and dropout for regularization.

    * **Model2:** A more complex neural network with three hidden layers (each with 128 neurons), batch normalization, and dropout.

Model Signature:

- **Input Dimensions:** The number of features in the dataset.
- **Output Dimensions:** A single continuous value representing the predicted average playtime.

4.4 Leveraging Domain Knowledge

Incorporating domain knowledge helps tailor the models to the specific problem, enhancing performance. The process includes:

- Ensuring the selected quality metrics are relevant to the business problem.
- Validating the incorporated domain knowledge against a baseline to avoid false assumptions.
- Summarizing the inclusion of domain knowledge to justify its impact on the model's performance.

In this project, domain knowledge about player behavior and game mechanics has been integrated to improve the model's predictions.

4.5 Model Training

The training process includes defining the objective, optimizer, regularization, and cross-validation strategies:

- **Objective:** Minimizing the Mean Squared Error (MSE) to ensure accurate predictions.
- **Optimizer:** Adam optimizer is used for its efficiency and effectiveness in training neural networks.
- **Regularization:** Dropout and batch normalization are employed to reduce overfitting.

- **Cross-Validation:** Stratified K-Fold cross-validation is used to ensure the model generalizes well to unseen data.

The dataset is split into training, validation, and test sets, with experiments documented to compare different runs and configurations.

4.6 Assuring Reproducibility

Reproducibility is ensured at both the method and result levels:

- **Method Reproducibility:** Detailed descriptions of algorithms, datasets, hyper-parameters, and runtime environments are provided. Meta-level explanations, including assumptions, are included. Random seeds are included as well.
- **Result Reproducibility:** Validating mean performance and assessing variance across different random seeds. Documenting experimental modifications and their impacts on model performance.
- **Experimental Documentation:** Maintaining comprehensive records of model changes, performance metrics, and the causes behind these changes. Utilizing tool-based approaches for version control and metadata handling.

5 Evaluation

5.1 Model validation report

We have tested the model on the test dataset and tracked its metrics. The model has a Root Mean Squared Error score of 202. The Giskard evaluation on the test dataset gave an MAE score of 9,92.

We have also used the Giskard tool in order to scan the model for vulnerabilities, such as biases in some cases. The report generated by Giskard mentions that it could not find any vulnerabilities in our model.

5.2 Discussion

The ML modeling phase provided us with multiple metrics, such as RMSE which is equal to 202. In comparison, Giskard found an MAE of 9.92.

In the initial business understanding phase, we have defined our ML success criteria to be a Mean Absolute Error Score of less than 200. The RMSE scores generated by the ML modeling phase, and the MAE by the validation done with Giskard, are both lower than the maximum acceptable threshold. Therefore, we think that the resulting model of our experiment fulfills the ML success criteria define in phase 1.

5.3 Deployment Decision

In light of the discussion above, the champion model of the experiment fulfills our ML success criteria, therefore it is eligible to be deployed. We proceed therefore with the deployment of this model. The details of the deployment are discussed in the following section.

6 Model deployment

We deploy our model on an on-premise local setup, as a preliminary production server for our business. The model would run on a machine with an Nvidia Geforce RTX3060 GPU, and an AMD Ryzen 7. Our empirical observation shows that this hardware is enough to run inference on our model.

6.1 Practical usability of the model

We have tested the model inference in practice, using some sample data given by the interested stakeholders. The users can put their feature values through an easy-to-use interface and get the prediction of the model in a couple of seconds. In practice, the deployed inference tool is useful and can inform its users about the predicted average playtime given the features of a game, which fulfills our initial business success criteria. Our initial goal is to give a tool to decision makers in the company, so that they can try to predict the effect of their decision on video game products, and specifically on the average playtime of the game. We think that this first iteration of the product is already useful in practice.

6.2 Deployment strategy

We deploy our model using a Flask API with two endpoints: `/info` and `/predict`, which provide the model metadata (signature, input schema, ...) and inference results, respectively. The model can also be deployed to a ready-to-ship Flask backend using a Dockerfile that was generated with the help of the MLFlow library. Furthermore, we have used GradIO to make a simple interface of the user, which sends REST API requests to our Flask API.

We are currently able to provide the tool on the public web for free, since we can deploy it publicly for free with GradIO. However, we think we can benefit from a better deployment setup. We plan to build an infrastructure consisting of a Kubernetes cluster, probably starting with a tool like MiniKube to setup the nodes, and then we

would dockerize the Flask API and GradIO frontend more properly and deploy them as pods. This would give us the possibility to have better deployment strategies like Blue-Green deployment or Canary deployment.

References

[1] Paul Bertens, Anna Guitart, and África Periáñez. Games and Big Data: A Scalable Multi-Dimensional Churn Prediction Model. In *2017 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 33–36, August 2017. arXiv:1710.02262 [stat].

[2] Anders Drachen, Eric Lundquist, Yungjen Kung, Pranav Rao, Rafet Sifa, Julian Runge, and Diego Klabjan. Rapid Prediction of Player Retention in Free-to-Play Mobile Games. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 12(1):23–29, June 2021.

[3] Anna Guitart, Shi Hui Tan, Ana Fernández del Río, Pei Pei Chen, and África Periáñez. From Non-Paying to Premium: Predicting User Conversion in Video Games with Ensemble Learning. In *Proceedings of the 14th International Conference on the Foundations of Digital Games*, pages 1–9, August 2019. arXiv:1906.10320 [cs, stat].

[4] Daniel Johnson, John Gardner, and Penelope Sweetser. Motivations for videogame play: Predictors of time spent playing. *Computers in Human Behavior*, 63:805–812, October 2016.