# Video Game Reception Prediction

## 1. Abstract

This project aims to forecast the potential reception of video games during the early stages of development by analyzing past releases and identifying key factors that influence player satisfaction. By examining data such as scores, tags, genres, release timelines, and detailed descriptions, the study identifies trends that can assist development teams in refining their concepts. Through a combination of data cleaning, exploratory data analysis (EDA), statistical techniques, and machine learning methods, the project provides valuable insights into what drives the success of similar games. The final product is an interactive tool that enables users to explore these insights and offers practical recommendations for optimizing game design and enhancing player engagement.

---

## 2. Introduction

### 2.1 Background

Developing video games demands substantial time, effort, and financial resources. In a competitive gaming industry, identifying the factors that drive a game's success is essential for developers. Gaining early insights into how a game might be received can guide teams in refining ideas and minimizing risks. By analyzing historical data from prior releases, developers can adopt data-driven approaches to optimize their processes, reduce costs, and enhance the game's potential to succeed.

### 2.2 Objectives

This project aims to:

1. Examine historical video game data to identify key factors impacting their reception.
2. Analyze the relationships between game features such as scores, genres, and release timelines to uncover patterns.
3. Create an interactive tool that allows users to explore insights and offers recommendations to improve game development strategies.

## 2.3 Dataset

The datasets are retrieved from https://store.steampowered.com/ and https://www.kaggle.com/datasets/nikdavis/steam-store-raw and the required data is converted into the csv files bearing 2000+ records

---

## 3.1 Data Cleaning

**Steps Taken**

The data cleaning process was an essential initial step to prepare the dataset for subsequent analysis and modeling. The following outlines the steps taken in detail:

1. **Removing Duplicate Records:**
    - All duplicate entries were identified and removed to ensure the dataset's uniqueness and accuracy.
2. **Standardizing Column Names:**
    - Column names were standardized for consistency and to improve readability, making them easier to work with during analysis.
3. **Handling Missing Values:**
    - Scores for games were retrieved from the SteamSpy dataset to fill gaps where the Steam App dataset had missing review data.
    - Null values were analyzed across all columns, and rows contributing little to the analysis were removed, reducing the dataset by 149 rows.
    - Ensured the type column had no "none" values remaining.
    - Cleaned the name column by removing rows with missing or irrelevant entries.
    - Columns with excessive null values were dropped to streamline the dataset.
4. **Discarding Unnecessary Columns:**

- Columns such as packages, package_groups, screenshots, movies, achievements, support_info, and background were removed as they were not relevant to the analysis and primarily served purchasing or cosmetic purposes.

5. **Replacing Irrelevant Data:**
    - Replaced missing or invalid values in the price column with 0 to represent free games.
6. **Converting Data Types:**
    - Converted the required_age column from float to integer, treating a value of 0 as no age restriction.
    - Addressed an outlier in required_age where a value of 1818 was likely a duplication of "18" and corrected it accordingly.
    - Removed any additional irrelevant columns during this process.
7. **Extracting Price and Currency Information:**
    - Parsed the price_overview column to extract and separate the price and currency into two distinct columns.
    - Assigned a price of -1 for games with missing price data.
    - Adjusted inconsistencies between the is_free and price columns by setting the price to 0 for games marked as free.
    - After extracting and cleaning the data, dropped the original price_overview and is_free columns.
8. **Cleaning Genres and Categories:**
    - Removed rows with missing values in the genres and categories columns.
    - Extracted and cleaned the content in these columns using appropriate methods to ensure standardized and meaningful data.
9. **Detecting and Correcting Outliers:**
    - Identified outliers in key columns and corrected them to maintain data consistency and reliability.

**Impact of Data Cleaning**

- The cleaning process resulted in a streamlined and accurate dataset, free of duplicates, missing values, and irrelevant data.
- Standardized and cleaned columns like genres and categories provide a solid foundation for analysis.
- The dataset is now ready for exploratory data analysis and modeling, ensuring dependable and insightful outcomes.

---

## 3.2 Exploratory Data Analysis (EDA)

The exploratory data analysis phase focused on uncovering valuable insights from the dataset and preparing it for subsequent modeling and analysis. Each team member brought distinct perspectives and hypotheses to examine various facets of the data. The following provides a detailed summary of the contributions and findings:

**Individual Contributions**

1. **Kisore Senthilkumar:**
    - **Questions**:
        - What genres are the most popular by taking the count of games per genre?
        - What are the targeted different age groups for the games that have been published?
    - **Findings**:
        - The top games are mostly Action games so ideally for the game to have best outcome it should be an action game.
        - Most games are targeted at kids as opposed to older people so the game needs to be potentially for kids to get the best outcome
2. **Harshitha Itta:**

- **Questions**:
  - Scores of games from certain developers are associated with higher average scores?
  - Which publishers produce higher-scoring games on average?
- **Findings**:
  - Particular developers consistently have higher scores, indicating that they excel at producing quality content.
  - The relationship between publishers, developers, and game scores is achieved.

3. **Neeraj Gummadi:**
   - **Questions**:
     - Is there a significant difference in “score” between free games and paid games as indicated by the “price”?
     - Do games that support more languages receive higher review scores?
   - **Findings**:
     - Free games are often aimed at earning high scores.
     - The bar plot shows a positive trend, it suggests that games with more supported languages (25) tend to have higher review scores.
4. **Shashank Govindu:**
   - **Questions**:
     - Does length of the game's detailed description affect the review score?
     - What game category ('Multi-player' vs 'Single-player') significantly affects the average review score?
   - **Findings**:
     - While longer descriptions don't always guarantee higher reviews, there might be a sweet spot for the ideal length. This suggests that concise and clear descriptions can be more effective than overly long or short ones.
     - Multiplayer games tend to have higher average review scores than single-player games. This suggests that multiplayer experiences may be more popular with players.

---

**Visualizations**

In this phase, various visualizations were utilized to enhance the analysis:

- Boxplots were used to detect patterns and outliers.
- Scatter plots helped explore correlations and trends.

- Bar charts enabled comparisons of key metrics across models.

These insights from the EDA served as a basis for feature selection and dataset refinement, setting the stage for Phase 2, where machine learning models were employed for deeper analysis.

---

# 4. Phase 2: Machine Learning and Statistical Analysis

In Phase 2, the emphasis shifted to utilizing machine learning and statistical models to gain deeper insights from the dataset and achieve the project goals. Team members contributed distinct analyses by applying various algorithms and statistical methods, focusing on specific questions regarding model performance and behavior.

---

## Individual Contributions and Key Findings

1. **Kisore Senthilkumar:**
    - **Questions Addressed**:
        - How the genres listed for the game affect its rating. Will specific genres cause them to have a higher probability of a positive rating?
    - **Methods/References**:

      XGBoost documentation at https://xgboost.readthedocs.io/en/stable/

      Random Forest Classifier documentation at https://scikit-learn.org/1.5/modules/generated/sklearn.ensemble.RandomForestClassifier.html
    - **Findings**:
        - The target variable and relationship to the features isn't consistent.

      .
2. **Harshitha Itta:**
    - **Questions Addressed**:
        - Can we classify games into different ownership levels (Low, High) based on their price?
        - Can we predict the log of ownership counts based on a game's positive ratings and average playtime?
    - **Methods/References**:

https://www.kaggle.com/code/prashant111/lightgbm-classifier-in-python ; https://www.analyticsvidhya.com/blog/2021/08/complete-guide-on-how-to-use-lightgbm-in-python/ ; https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMClassifier.html for LightGBM used in hypo 1

https://scikit-learn.org/dev/modules/generated/sklearn.linear_model.ElasticNet.html ; https://medium.com/@abhishekjainindore24/elastic-net-regression-combined-features-of-l1-and-l2-regularization-6181a660c3a5 for elasticnet used in hypo2

https://scikit-learn.org/dev/modules/generated/sklearn.ensemble.StackingRegressor.html https://www.analyticsvidhya.com/blog/2020/12/improve-predictive-model-score-stacking-regressor/ for stacking regressor used in hyo2

- **Findings**:
  - The increase in accuracy from binary to multi-class classification results from the fact that the three-class split (Low, Medium, High) provides a closer alignment with the natural distribution and structure of the data.
  - The plot compares predicted vs. actual values. Points close to the diagonal line show accurate predictions. The model does well for ownership counts (values close to the line).

3. **Neeraj Gummadi:**
   - **Questions Addressed**:
     - Is there a significant difference in “score” between free games and paid games as indicated by the “price”?
     - Do games with more “supported_languages” tend to receive higher “scores”?
   - **Methods/References**:

     Scikit-learn documentation on Gradient Boosting:

     https://scikit-learn.org/stable/modules/ensemble.html#gradient-boosting

     Gradient Boosting tutorial with scikit-learn:

     https://scikit-learn.org/stable/auto_examples/ensemble/plot_gradient_boosting_regression.html

     Logistic Regression:

     Scikit-learn documentation on Logistic Regression:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

Logistic Regression tutorial with scikit-learn:

https://scikit-learn.org/stable/auto_examples/linear_model/plot_logistic.html

- **Findings**:
  - Logistic Regression with expanded features achieved a 65.3% accuracy and a 0.715 ROC-AUC score, indicating reasonable predictive power for classifying game scores.
  - A scatter plot showing the predicted vs. actual log-transformed scores based on the number of supported languages, categories, and release year.

4. **Shashank Govindu:**
   - **Questions Addressed**:
     - What is the relationship between the length of the "detailed_description" and the "score" of the game?
     - How do the "categories" affect the "score" of a game?
   - **Methods/References**:

     Scipy documentation for ANOVA (scipy.stats module):

     https://docs.scipy.org/doc/scipy/reference/stats.html#anova

     ANOVA tutorial with scipy.stats:

     https://www.statology.org/one-way-anova-python/

     Polynomial Ridge Regression:

     Scikit-learn Polynomial Features and Ridge Regression:

     https://scikit-learn.org/stable/auto_examples/linear_model/plot_polynomial_interpolation.html

     Polynomial Regression tutorial (with Ridge as an extension):

     https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.PolynomialFeatures.html

     https://scikit-learn.org/stable/modules/linear_model.html#ridge-regression
   - **Findings**:
     - The plot suggests a positive but limited correlation between description length and score. Games with more detailed descriptions tend to have slightly higher scores, up to a point.
     - The box plot reveals some interesting patterns: Multi-player games tend to have a higher median log score compared to Single-player and Other

games. They also show a wider interquartile range, meaning there's greater variability in scores within the Multi-player category.

---

# 5. Phase 3: Data Product Development

The culmination of this project is the development of an interactive application that integrates a front-end interface, back-end processing, and a database. This app bridges the gap between the analyses conducted in Phases 1 and 2 and delivers actionable insights to end-users. Designed as a comprehensive tool, the app enables users to explore data, visualize key metrics, and perform predictive analyses. With its intuitive interface and robust functionality, the application provides developers and researchers with valuable insights into video game reception and related trends.

---

## 5.1 Overview

The Video Game Reception Prediction application is an intuitive platform that seamlessly integrates various aspects of the project into a cohesive tool. Built using Streamlit, the app is designed to be user-friendly, accessible, and efficient. It empowers users to interact with the video game dataset, explore data insights, and make predictions using advanced machine learning models. The app also provides CRUD (Create, Read, Update, Delete) operations to manage the database, ensuring complete control over the data. With predictive tools and data visualization, users can analyze game-related trends and factors affecting reception without needing technical expertise.

**Key Objectives**

- Simplify interaction with the video game database for both data management and exploration.
- Provide intuitive tools to predict and understand factors like game score, pricing, and positive ratings.
- Enhance decision-making through interactive data visualizations and predictive analytics.

## 5.2 Features

The app was designed with a focus on user interaction and actionable insights. Its core features include:

**1. Advanced CRUD Operations (Database Management)**

- **Add New Games:**
  Users can input game details such as name, release date, platforms, developers, genres, and more via a user-friendly form. This ensures the database remains up-to-date and relevant.
  *Example:* Users can add new Steam games using structured fields like App ID, positive/negative ratings, achievements, and average playtime.

- **Update Existing Entries:**
  Modify details of existing games to ensure accuracy and relevance in real-time. For example, users can edit pricing information or add supported platforms.

- **Delete Irrelevant Data:**
  Easily remove outdated or redundant game entries to maintain a clean and optimized database.

- **View and Search Entries:**
  Users can search games by various attributes like genres, release date, and categories, facilitating quick data exploration.

**2. Predictive Analytics**

The app leverages advanced machine learning models to provide precise predictions.

**Predict Average Playtime**

- **Model Used:** LightGBM (Light Gradient Boosting Machine)
    - A highly efficient gradient boosting framework known for its speed and accuracy, especially on large datasets.

- Features considered: Positive Ratings, Negative Ratings, Price, and Achievements.
- **Purpose:** Helps predict how much time players are likely to spend on a game, offering insights into game engagement.

**Predict Positive Ratings**

- **Model Used:** Random Forest Regressor
  - A robust ensemble learning method that averages predictions from multiple decision trees to improve accuracy and reduce overfitting.
  - Features considered: Negative Ratings, Price, Achievements, and Average Playtime.
  - **Purpose:** Estimates the proportion of positive ratings, assisting developers in gauging audience satisfaction.

**Predict Game Pricing**

- **Model Used:** Decision Tree Regressor
  - A simple yet effective tree-based model that predicts outcomes by segmenting data based on key variables.
  - Features considered: Positive Ratings, Negative Ratings, Achievements, and Average Playtime.
  - **Purpose:** Predicts the optimal pricing for games based on market and game characteristics.

**Predict Game Score**

- **Model Used:** XGBoost (Extreme Gradient Boosting)
  - A state-of-the-art gradient boosting algorithm known for its scalability and performance in predictive tasks.
  - Features considered: Is Action, Is Indie, Is Casual, Is Strategy, Is Adventure, Is Free-to-Play, Is Simulation, and Platform Support (Windows, Mac, Linux).
  - **Purpose:** Forecasts the overall reception score of a game, enabling developers to anticipate its market success.

**3. Intuitive User Interface**

Each predictive feature is presented via an intuitive and clean UI, ensuring ease of use for both technical and non-technical users.

- **Dynamic Input Fields:** Tailored forms for specific predictions ensure that users can input accurate data for effective results.
- **Accessible Design:** A minimalist approach ensures that all features are easy to navigate.

**References to Models Used**

1. **LightGBM:**
    - Reference: Ke, Guolin, et al. "LightGBM: A highly efficient gradient boosting decision tree." *Advances in Neural Information Processing Systems* 30 (2017).
    - Chosen for its superior speed and handling of categorical data in predicting playtime.
2. **Random Forest Regressor:**
    - Reference: Breiman, Leo. "Random forests." *Machine Learning* 45.1 (2001): 5-32.
    - Selected for its robustness and ability to handle noisy datasets while predicting positive ratings.
3. **Decision Tree Regressor:**
    - Reference: Quinlan, J. Ross. "Induction of decision trees." *Machine Learning* 1.1 (1986): 81-106.
    - Used for its interpretability and efficiency in predicting game pricing.
4. **XGBoost:**
    - Reference: Chen, Tianqi, and Carlos Guestrin. "XGBoost: A scalable tree boosting system." *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 2016.
    - Ideal for its scalability and high performance in predicting game scores.

**Example Use Cases**

1. **Developers:**
    - Optimize game pricing using predictive tools.
    - Understand key factors influencing scores to improve future releases.
2. **Gamers:**
    - Identify games with longer playtimes.
    - Anticipate player satisfaction based on ratings predictions.
3. **Market Analysts:**
    - Analyze trends in genre-specific ratings and reception.
    - Forecast pricing trends in the gaming industry

---

## 5.3 Implementation

The app was designed with a modular architecture to ensure flexibility and scalability. Below are the primary components:

**1. Database**

- **Structure:**
    - A SQL database was utilized to store and manage structured data, including output.csv and steam.csv, along with additional derived metrics.
- **Functionality:**
    - The database supports full CRUD (Create, Read, Update, Delete) operations, enabling users to add, view, update, and delete data seamlessly through the application interface.
- **Optimization:**
    - Advanced indexing and query optimization techniques were implemented to ensure efficient retrieval and handling of large datasets.

**2. Front-End**

- **Framework:**
    - The front-end was developed using Streamlit, offering a visually engaging and user-friendly interface for interaction.
- **Interactivity:**
    - Users can interact with widgets such as sliders, dropdowns, and text inputs to filter data, create visualizations, and access predictive features.
- **Customization:**
    - The interface adapts dynamically to user inputs, ensuring an intuitive and smooth user experience.

**3. Back-End**

- **Core Logic:**
    - Python scripts handle all data processing, statistical analysis, and machine learning model predictions. These scripts integrate seamlessly with both the front-end and the database.
- **Integration:**
    - Algorithms and insights from output.csv and steam.csv datasets were incorporated, along with metrics derived in earlier project phases.
- **Deployment:**
    - Hosted on a cloud platform, the back-end ensures robust performance and scalability to accommodate multiple users simultaneously.

---

# 6. Conclusion

This project effectively demonstrated the power of integrating statistical analysis, machine learning methods, and interactive tools to enhance video game reception prediction. By combining comprehensive data cleaning, exploratory analysis, and advanced modeling techniques, the team uncovered valuable insights into game attributes and their influence on predictions. This approach established a solid foundation for improving the efficiency and

accuracy of predictive models, providing actionable insights into factors driving video game performance and reception.

## Future Potential

While this project successfully met its objectives, it also opened new avenues for exploration and improvement:

1. **Expanding Dataset Scope:**
    - Applying the methodologies developed in this project to broader datasets or other gaming platforms could validate the findings and offer a deeper understanding of trends in video game reception.
2. **Enhanced Model Optimization:**
    - Future work could explore advanced optimization techniques, such as Bayesian optimization or reinforcement learning, to improve the performance and accuracy of predictive models.
3. **Automated Insights and Reports:**
    - Integrating features like automated report generation and advanced analytics in future iterations of the app could streamline insights and reduce reliance on manual exploration.

## Final Reflection

This project demonstrates how data-intensive computing can be harnessed to address complex challenges in machine learning. The comprehensive integration of statistical analysis, machine learning techniques, and an interactive application highlights the transformative potential of these tools in understanding and optimizing training pipelines. By focusing on actionable insights and practical applications, the project provides a scalable and impactful solution for improving machine learning workflows.

# 7. Appendices

1. **Dataset Details**:
    - Snapshots of output.csv and steam.csv.
2. **Code References**:
    - Cleaning and EDA scripts: Steam_Data_Phase_1.ipynb
    - Phase 2 analysis: Individual notebooks.