

**COVENTRY UNIVERSITY**

**College of Engineering, Environment and Science**

**School of Science**

***7150CEM - Data Science Project***

**Predicting Sales Performance of Steam Games Using Machine Learning: Analyzing Developer, Categories and Publisher Impact**

**Author: Vinay Kumar Chinthamalla**

**SID: 14734425**

**Supervisor: Mark Johnston**

**Submitted in partial fulfilment of the requirements for the Degree of Master of Science in Data Science**

**Academic Year: 2024/25**

**Declaration of Originality**

I declare that this project is all my own work and has not been copied in part or in whole from any other source except where duly acknowledged. As such, all use of previously published work (from books, journals, magazines, internet etc.) has been acknowledged by citation within the main report to an item in the References or Bibliography lists. I also agree that an electronic copy of this project may be stored and used for the purposes of plagiarism prevention and detection.

**Statement of copyright**

I acknowledge that the copyright of this project report, and any product developed as part of the project, belong to Coventry University. Support, including funding, is available to commercialise products and services developed by staff and students. Any revenue that is generated is split with the inventor/s of the product or service. For further information please see [www.coventry.ac.uk/ipr](http://www.coventry.ac.uk/ipr) or contact [ipr@coventry.ac.uk](mailto:ipr@coventry.ac.uk).

**Statement of ethical engagement**

I declare that a proposal for this project has been submitted to the Coventry University ethics monitoring website (https://ethics.coventry.ac.uk/) and that the application number is listed below (Note: Projects without an ethical application number will be rejected for marking)

Signed: Vinay Kumar Chinthamalla Date: 13-12-2024

Please complete all fields.

|  |  |
| --- | --- |
| First Name: | Vinay Kumar |
| Last Name: | Chinthamalla |
| Student ID number | 14734425 |
| Ethics Application Number | P181020 |
| 1st Supervisor Name | Mark Johnston |
| 2nd Supervisor Name | Anup Pandey |

**This form must be completed, scanned and included with your project submission to Turnitin. Failure to append these declarations may result in your project being rejected for marking.**

# Abstract

This project develops a machine learning based predictive model to make the prediction of Steam sales performance of games based on information about publishers, platforms, categories and developers. The project aimed to identify factors which impact game sales based on a dataset upwards of 80,000 entries and 39 columns, to supply actionable insights useful for developers and publishers to optimize their strategies with. Effective data pre-processing, feature selection, data model development and rigorous evaluations were the primary objectives of machine learning techniques.

Data first went through data cleaning and preparation by filling missing values, encoding categorical variables and balancing the data trying to improve the model performance. With EDA, we discovered key observations and relationships between features and the dependent variable, sales performance. User reviews were sentiment analysed to further deepen the feature set with qualitative information. Random forest, gradient boosting, and decision trees were implemented and compared, with hyper-parameter tuning to improve the performance of all algorithms. To evaluate, metrics like MSE, R² score etc. and confusion matrixes were used to have a thorough understanding of the model’s effectiveness.

In results, it was found that ensemble models, particularly, Gradient Boosting, were the best models with respect to predictive accuracy. Using feature importance analysis and a set of critical factors like developer reputation, pricing, genre, and review sentiments, all had impacted the sales of a game. These findings point to which factors are key to success in the Steam gaming ecosystem. However, the project faced challenges such as imbalanced data distribution, and missing data, which were solved through imputation and resampling techniques respectively. Future work could include bringing in additional sources of data, including marketing trends and demographic data, or using different deep learning models, such as CNNs or LSTMs, to improve the performance even more. The project has proven to be a success, successfully achieving its overall objective in tandem with its goals of understanding and predicting game sales performance using machine learning.

# Table of Contents

[Abstract 2](#_Toc184994253)

[Table of Contents 3](#_Toc184994254)

[Acknowledgements 6](#_Toc184994255)

[1 Introduction 7](#_Toc184994256)

[1.1 Background to the Project 7](#_Toc184994257)

[1.2 Research Aim 8](#_Toc184994258)

[1.3 Project Objectives 9](#_Toc184994259)

[1.4 Research Questions 9](#_Toc184994260)

[1.5 Overview of This Report 10](#_Toc184994261)

[2 Literature Review 11](#_Toc184994262)

[2.1 Introduction 11](#_Toc184994263)

[2.1.1 The role of developer and publisher in game success 11](#_Toc184994264)

[2.1.2 Broader Perspectives on Developer and Publisher Influence 11](#_Toc184994265)

[2.1.3 Influence on Developer Reputation 11](#_Toc184994266)

[2.1.4 Case studies: Publisher Backing and the Market Influence 12](#_Toc184994267)

[2.2 Machine learning predicting game sales performance 12](#_Toc184994268)

[2.2.1 Use of Machine Learning in Sales prediction 12](#_Toc184994269)

[2.2.2 Challenges in Sales Prediction by Machine Learning 13](#_Toc184994270)

[2.2.3 Building Sentiment Analysis and User Reviews 13](#_Toc184994271)

[2.2.4 Existing Studies and their work 13](#_Toc184994272)

[2.3 Impact of early access and release strategies 14](#_Toc184994273)

[2.3.1 Early Access: Opportunities and Risks 14](#_Toc184994274)

[2.3.2 Full Release: Traditional versus Staggered Launches 15](#_Toc184994275)

[2.3.3 Timed Exclusivity & Subscription Editions 15](#_Toc184994276)

[2.3.4 Pricing strategies and their Impact on sales 15](#_Toc184994277)

[2.3.5 First Pricing and Consumer Perception 15](#_Toc184994278)

[2.3.6 Short Term Effects of Discounts 16](#_Toc184994279)

[2.3.7 Bundles and Tiered Pricing 16](#_Toc184994280)

[2.3.8 Free to Play and Microtransactions 16](#_Toc184994281)

[2.4 Sentiment Analysis and Text Mining for Sales Prediction 17](#_Toc184994282)

[2.4.1 Sales Prediction with Text Mining Techniques 17](#_Toc184994283)

[2.4.2 Existing Research Gaps and Proposed Contribution 18](#_Toc184994284)

[2.5 Critical analysis 18](#_Toc184994285)

[2.6 Conceptual framework 20](#_Toc184994286)

[2.7 Summary 21](#_Toc184994287)

[3 Methodology 22](#_Toc184994288)

[3.1 Research approach 22](#_Toc184994289)

[3.2 Data Collection and Data Description 22](#_Toc184994290)

[3.3 Data pre-processing 24](#_Toc184994291)

[3.4 Exploratory Data Analysis (EDA) 24](#_Toc184994292)

[3.5 Model Development 25](#_Toc184994293)

[3.6 Model Evaluation 26](#_Toc184994294)

[4 Requirements 27](#_Toc184994295)

[5 Implementation 29](#_Toc184994296)

[5.1 Data collection and loading 29](#_Toc184994297)

[5.2 Data Pre-processing 30](#_Toc184994298)

[5.3 Model Implementation 32](#_Toc184994299)

[5.3.1 Machine Learning Models 32](#_Toc184994300)

[5.3.2 Deep Learning Models 34](#_Toc184994301)

[5.3.3 Justification for model selection 36](#_Toc184994302)

[5.4 Model Evaluation 36](#_Toc184994303)

[5.5 Data Balancing 37](#_Toc184994304)

[6 Analysis 38](#_Toc184994305)

[6.1 Reason For Using Positive/Negative Sentiment Instead Of Raw Polarity Scores 48](#_Toc184994306)

[7 Project Management 50](#_Toc184994307)

[7.1 Project Schedule 50](#_Toc184994308)

[7.1.1 Project Management and Adaptability 51](#_Toc184994309)

[7.1.2 Project Supervision and Feedback on Drafts 51](#_Toc184994310)

[7.2 Risk Management 51](#_Toc184994311)

[7.2.1 Risk Materialization and Response 52](#_Toc184994312)

[7.3 Quality Management 52](#_Toc184994313)

[7.3.1 Standards Adopted 52](#_Toc184994314)

[7.3.2 Progress Review Techniques 52](#_Toc184994315)

[7.3.3 Methods for Outcomes Evaluations 52](#_Toc184994316)

[7.4 Social, Legal, Ethical and Professional Considerations 53](#_Toc184994317)

[7.4.1 Professional Considerations 53](#_Toc184994318)

[7.4.2 Social Considerations 53](#_Toc184994319)

[7.4.3 Legal Considerations 53](#_Toc184994320)

[7.4.4 Ethical Considerations 53](#_Toc184994321)

[7.4.5 Privacy Considerations 53](#_Toc184994322)

[8 Critical Appraisal 54](#_Toc184994323)

[8.1 Positive Aspects 54](#_Toc184994324)

[8.2 Negative Aspects and Limitation 54](#_Toc184994325)

[8.3 Knowledge Gained 55](#_Toc184994326)

[9 Conclusions 56](#_Toc184994327)

[9.1 Achievements 56](#_Toc184994328)

[9.2 Future Work 56](#_Toc184994329)

[10 Student Reflections 57](#_Toc184994330)

[10.1 Achievements and Strengths 57](#_Toc184994331)

[10.2 Challenges Encountered 57](#_Toc184994332)

[10.3 Lessons Learned and Areas for Improvement 57](#_Toc184994333)

[Bibliography and References 59](#_Toc184994334)

[Appendix A – Project Source Code 1](#_Toc184994335)

[Appendix B – Certificate of Ethics Approval 10](#_Toc184994336)

# Acknowledgements

I would like to express my sincere gratitude to my supervisor, **Mark Johnston**, for their invaluable guidance, encouragement, and support throughout the research and development of this dissertation. Their insights and expertise have been instrumental in shaping this work.

I am also deeply grateful to Coventry University for providing the resources and learning environment necessary to complete this project. Special thanks go to the College of Engineering, Environment, and Science for their assistance and resources that enabled the successful execution of this research.

I extend my appreciation to my family and friends for their unwavering encouragement and understanding during the challenges faced throughout this academic journey. Their support has been a source of great strength and motivation.

Finally, I would like to thank the developers and contributors of the Steam dataset for making the data available, which served as the foundation for this research. Without this dataset, the insights and findings presented here would not have been possible.

Thank you all for your invaluable contributions and support.

# Introduction

Today, game distribution has been revolutionised with platforms like “Steam” website, and the gaming industry is one of the fastest growing sectors of the global economy (Wang et al., 2020). Steam is a platform where indie (small independent group of game developers) and major game developers can bring their products into public and reach millions of Steam players all over the world. Developers, publishers and stakeholders need to understand why games work on such platforms, as well as which factors contribute to that success. Factors with an influence on sales performance include developer reputation, publisher backing, user engagement, pricing strategies and release approaches. This dissertation utilizes machine learning techniques to forecast the sales performance of a game to discover what factors drive game success on “Steam” (Syam & Sharma, 2018). Through such predictive analysis, the study aims to provide actionable recommendations for better gaming development and marketing strategy and to aid stakeholders in making data-based decisions.

## Background to the Project

Over the last 10 years, the video game industry has developed very fast, due to advancement of technology, increase in the number of people using the internet and its change to digital distribution (Lin et al., 2018). The other space that has emerged as one of key importance is the Steam, created by Valve Corporation, which is a marketplace website where the games from both of the big developers and small number of indie developers are available to millions of gamers around the world. With over 50,000 titles in its library today, “Steam” is a highly competitive market and most games don't stand a chance of commercial success (Wang et al., 2020). Many developers and publishers have found that with the platform's diverse and extensive catalog, and a large base of digital games, its sales performance is driven by factors that are difficult to understand (Lin et al., 2018).

The landscape of game distribution has dramatically changed for independent developers. The advancement of game development tools has lowered barrier to entry and enabled smaller teams to create games of higher quality than ever before without huge upfront investment. However, with an oversaturated market, it’s become harder to stand out and bring in major sales. Especially for indie developers who don’t have as much money as larger publishers to invest into marketing, it’s important for people to know some things that influence marketability and how successful their game could be. This can help developers customize their marketing strategies as well as determine their realistic sales target and better release time plan (Wang et al., 2020).

However, publishers are involved in publishing many game titles at once and require data driven insights for better resource allocation. So, they must decide which games to heavily market, which to embed in sales bundles, and when to launch in order to avoid competing with other major launches. A poorly timed or under promoted release can lose money, and because the stakes are high, it is a risk that publisher cannot afford (Tobon and Abril, 2024). Thus, in that context predictive models become very helpful as they can forecast that sales performance using historical data and market trend and publishers can use that to make important decisions and maximise their return on investment.

Although there is a wealth of data available on game performance, very little of this is used to understand how different variables interact and influence sales performance, especially for indie developers and publishers who manage multiple titles. Tools and techniques have empowered smaller developers to make great games, but without insight into market trends and user behaviour, there are fewer great games than there are developers (Rahman et al., 2024).

Individual factors like user reviews, engagement metrics, and market timing are considered in the existing literature with fragmentary perspectives (Rahman et al., 2024), but integrated models across the variables to predict the behaviour do not exist, nor provide actionable insights. The existence of this gap shows that indie developers and publishers need robust, data driven frameworks to aid them allocate resource, devise marketing strategies and plan release dates to better flourish in this competitive market (Rahman et al., 2024).

In addition to user generated content, reviews, and ratings on “Steam”, sales performance has become even more complex to predict. Reviews and ratings are a critical component to a game’s reputation as well as to the potential selling of a game. It means a good review will show the game to more people on the platform while a bad review can absolutely impact sales (Lin et al., 2018). Other metrics of engagement like how many people play the game and how long they spend playing also tell you that the game is popular and signals potential long-term sales. For developers and publishers trying to boost user satisfaction and game trade, understanding how these metrics affect them is essential (Lin et al., 2018).

Additionally, developers and publishers can employ pricing strategies that have very large impacts on sales outcomes. Games are susceptible to price fluctuations, and since Steam’s monthly sales and discount events are frequent, buyer behaviour can change because of the prices (Josef et al., 2022). An appropriate strategy has to be planned regarding the decisions on initial pricing, discounts and bundle to optimize revenue without diminishing the perceived value. One of the benefits of an effective pricing strategy is impulse buying during sales events or creating long-term value perception for some games especially those from small developers (Syam & Sharma, 2018).

With such complexities, it is necessary to have a complete solution that allows the game sales to be analysed in terms of numerous factors acting on it and provide actionable insights on them. This project seeks to clarify these issues by creating multiple machine learning based predictive model that studies developer reputation, publisher influence, user engagement metrics, and price strategy attributes. By using machine learning techniques, the project attempts to find key drivers of sales performance and contribute as a strong tool for developers and publishers to make better decisions.

Potential users of this solution include indie developers that want to optimize their launch strategies, large publishers who want to allocate marketing resources efficiently, and industry analysts who seek to predict market trends. This project can offer valuable data driven insights to the gaming industry by giving access to an approach to understand what factors tend to game success and predicting the release game on Steam, and help the stakeholders to make the informed choices that could lead to better financial outcomes and successful game releases on Steam.

## Research Aim

The aim of this study is to build and compare several machine learning-based systems that predicts the sales performance of Steam games by analysing developer, platform, and publisher data, helping stakeholders make informed decisions regarding game releases and marketing strategies.

## Project Objectives

* To pre-process and analyze the Steam games dataset, focusing on developer, platform, and publisher attributes.
* To develop machine learning models capable of predicting game sales performance based on these attributes.
* To evaluate the models using performance regression metrics such as MSE, R² score
* To identify the most influential factors impacting sales performance and provide insights into the gaming industry’s market dynamics.

## Research Questions

1. **What is the predictive accuracy of various machine learning models in forecasting Steam game sales performance based on developer and publisher attributes?**
   * This question now focuses on the predictive power of different machine learning models while examining the developer and publisher impact on game sales performance.
2. **Which specific developer and publisher attributes are the most influential in determining a game's sales performance on Steam?**
   * By investigating specific attributes such as developer reputation and publisher backing, this question delves into the primary factors driving game success without considering platform data.
3. **What challenges arise in predicting game sales performance on Steam, particularly in terms of data limitations, model interpretability, and feature interactions, and how can these challenges be mitigated?**
   * This explores broader challenges in predictive modelling, focusing on how the absence of platform attributes affects the model’s performance and how other factors interact.
4. **How do early access games compare to traditionally released games in terms of sales performance, and what role do machine learning models play in identifying this trend?**
   * Without considering platform data, this question still examines the impact of release strategies (early access vs. traditional) on sales performance using machine learning.
5. **What role do user engagement metrics play in predicting long-term sales performance, and how can machine learning models quantify their influence?**
   * This focuses on the influence of engagement metrics, such as user reviews and ratings, without considering platform-related dynamics.
6. **Can sentiment analysis on user reviews be effectively integrated with traditional sales prediction models to improve accuracy, and how does it impact the interpretability of the results?**
   * The question focuses on integrating text-based data, like sentiment analysis, with numerical data related to developer and publisher, ignoring platform effects.
7. **How do different pricing strategies influence sales, and can machine learning models accurately predict the optimal pricing model for different game types?**
   * This explores the economic factors driving sales, such as pricing strategies, while excluding platform-based analysis.

## Overview of This Report

An Introduction chapter gives background and the research goal and goals in this structured research report. Next, the Literature Review critically evaluates several research articles and related investigations. After that, the Methodology chapter describes the research techniques. After that, the Requirements and Analysis chapters provide the study's needs and analysis. The study finishes with Project Management, Critical Appraisal, and a Conclusion to summarise the research and its consequences.

# Literature Review

## Introduction

The platform Steam is set up by a gaming industry with a large game library, both created by the big companies and some smaller independent companies. By challenges of developers, publishers, and other stakeholders attempting to strike a balance between the quantity and quality of games and apps that can be developed and released, predicting sales performance in such a dynamic marketplace is an important task (Wang et al., 2020). Identifying these patterns and forecasting sales based on developer attributes, publisher backing and user engagement metrics, has brought machine learning to the forefront as a powerful tool. The purpose of this literature review is to review the current research in predicting sales performance in the gaming industry by understanding how machine learning models have been used, how different approaches are effective and what are the important variables which determine success of a game. This review critically analyses various studies, gaps in the literature, and indicates a way of further research.

### The role of developer and publisher in game success

There is a massive amount of research out there done in the gaming industry dealing with factors that might affect the success of video games: such as gameplay quality, genre, innovation, and others. Within this broad scope however, the role of developers as well as publishers have both grown to become significant determinants of a game’s commercial performance. Developer and publisher’s reputation and influence will determine a game’s sales success by setting market perception and expectations (Wang et al. 2020). This section evaluates the effects of these roles through increasingly tight focus, from trends in general industry, to individual case studies that reveal the roles' importance in practice.

### Broader Perspectives on Developer and Publisher Influence

In early studies examining how the gaming market works, society focused on the collective effect of market forces as a whole, including preferences for certain genres of games or towards players involving themselves with the gamification process. According to the research conducted by Syam and Sharma (2018), games backed by the publisher had a better visibility and sales due to publisher's access to big marketing campaigns, distribution channels as well as promotional opportunities. McKaughan, et al (2017) also found that the publishers’ financial resources, and pre-release marketing strategies played a perhaps equal role, in whether a game made a commercial success. These studies illustrate the great influence publishers have in the marketplace. For example, by running global campaigns for games like Assassin’s Creed (Video Game) with the help of games like Assassin's Creed, Ubisoft has gained a cult following of rabid fans who are more invested in the brand than in its critical reception, and thus help to create big hype, and post-release sales (Santos, 2022).

### Influence on Developer Reputation

A number of studies have proven that the reputation of a developer can substantially affects the game’s sales. Tong (2021) show that games from known developers with a track record of successful releases generate more attention and yield superior sales outcomes relative to the same genres of games. On the one hand, the brand loyalty with well-established developers, it benefits because their previous successful games are creating expectations of quality and innovation. If a developer has a history of releasing successful products, there’s a greater likelihood that players who are aware of a developer’s past project will buy its new release. For instance, companies such as Valve, and CD Projekt Red have big followings because of their historical success in making certain they will sell more over time (Muhammed et al., 2023).

But the reputation of well-known developers can work for both ways. Muhammed et al. (2023) maintain that high reputability developers can have backlash even if they don’t meet the high expectations established by their own high successes. As an example, CD Projekt Red’s Cyberpunk 2077 garnered overwhelming demand when it was released, much of which is due to the developer’s good name. Nevertheless, despite serious praising, the game’s release was marred by technical issues and negative reviews along with huge refunds, which reveal dangers of being based on reputation alone (Shih & Wang, 2024).

On the other hand, while indie developers don’t have the established track record of bigger studio, it is harder to reach initial visibility in a saturated space. But many indie developers made it big by introducing new ways of playing, telling a story in a new manner, or running a viral marketing campaign. Reardon and Wright (2021) studies illustrate that indie developers can do well in the marketplace with their small budget and lack of fame based on social media buzz.

### Case studies: Publisher Backing and the Market Influence

Cases of large publishers such as Electronic Arts (EA) and Activision Blizzard are examined to show that their extensive marketing campaigns and cross promotional strategies help to drive game success. By way of such publishers, games tend to gain the run of the field both in terms of sales and marketing, which McKaughan et al. (2017) reflected on in their research. But such market dominance always comes at the expense of creativity. According to Finch and Buchmesse (2019), large publishers such as EA focus too much on making safe investments at the cost of innovation as it relies on annualized franchises such as FIFA and Madden NFL. Although these games are consistently profitable, they are regularly condemned for failing to innovate, and being fixated less on improving gameplay and instead on pushing microtransactions (Williams, 2021).

On the other hand, smaller, independent publishers may foster creativity, but tend to lack the resources to market and distribute games at a global level effectively. This is a common tension in the gaming industry in that developers must negotiate between their creative freedom and the desire to create viable products.

This review shows how developers and publishers impact a game’s success. Through a broader trend analysis, focusing on as narrow, and as granular, as possible, it becomes clear that while games backed by publishers tend to have higher sales figures and a proven trajectory, the industry is changed by independent developers and smaller publishers. This duality is the basis of the current research's focus to understand the relationship between the developer and the publisher's characteristics and their sales performance for the games featured on Steam.

## Machine learning predicting game sales performance

The use of machine learning in the gaming industry has risen sharply, with more accurate predictions, as well as data driven decision making.

### Use of Machine Learning in Sales prediction

Most commonly, these sales predictions are performed with supervised learning models like Random Forest, Gradient Boosting and Decision Trees (Rahman et al., 2024). The usage of these models enables the identification of relationship of game related features and future sales outcomes. Machine learning has been shown in Teja et al., (2023) to be useful in this domain. They used Random Forest and Decision Trees to predict sales on game platforms like Steam, noting that developer reputation, publisher backing, and user engagement metrics do play a large factor in the sales performance. Random Forest, especially, did well in spite of the fact that it could successfully handle large amounts of data and fit well to non-linear relations between the features.

Lee et al. (2021) is another study that used Gradient Boosting Machines (GBM) to predict games sales, including Pay to Play, incentives, discounts, pre-release hype, and marketing efforts. The research showed that pre-release buzz on social media and early user reviews were strong indicators of first sales. By exploiting temporal data and tracking the evolution of marketing campaigns, are able to model the effect of early momentum on game sales. Results showed that early marketing efforts and the use of machine learning in anticipating their impact were important.

### Challenges in Sales Prediction by Machine Learning

However, like other challenges, predicting game sales with machine learning models have many advantages as well. Key questions are on data quality and availability. PraveenaSri and Prasuna (2023) point out that sales data is usually incomplete, inconsistent or unavailable, causing model training to be off and prediction to become less reliable. In addition, many of the factors that influence sales of a game, viral or word-of-mouth-based trends for example, are generally hard to quantify and may not be fully captured in even the data. It’s hard to create comprehensive models that take into account everything, so this is challenging (Rahman et al., 2024).

One of the biggest challenges is that machine learning models are not interpretable. As Random Forest and Gradient Boosting are very complex algorithms, they work as "black boxes" and frequently unable to tell which features have more impact on sales predictions. To help solve this, some researchers, such as Florez and Zuluaga (2023), have used feature importance methods of SHAP (SHapley Additive exPlanations) to learn more about the contribution of different attributes to game sales. It shows how to use this technique to uncover the most important feature, such as user reviews or developer reputation, but leaves the challenge of putting these insights into real strategies for developers and publishers (Florez & Zuluaga, 2023).

### Building Sentiment Analysis and User Reviews

Another rich data source to predict sales performance is user reviews and feedback. Sales prediction models have been enhanced accuracy with the addition of Natural language processing (NLP) techniques, such as sentiment analysis. Machine learning models from Panwar and Bhatnagar (2020) were extended to include sentiment analysis, and they found that games with high sales and positive reviews tended to maintain high sales at all time periods. Machine learning models can take aspects of players’ sentiment present in the user review and give deeper insights on how the feedback affected the players’ sales performance in the long run. Reviews are dynamic for a game as they become updated or expanded over time, making it tricky to truly capture player sentiment well. From sentiment analysis, sophisticated NLP methods are needed, and even most seeking nuanced user feedback may find it difficult for models to interpret consistently (Hamarashid et al., 2022).

### Existing Studies and their work

Rahman et al. (2024), Teja et al. (2023), and Lee et al. (2021) provide great studies of machine learning on the sales prediction in gaming, but we need to evaluate their tested methods, research questions and potential limitations closer. The supervised learning models, namely Random Forest and Decision Trees were considered by Rahman et al (2024) to predict the relationship between game related features and sales outcomes. According to their study, they found these models were adequate for large datasets, and the nonlinear relationship, indicating that they might be a suitable method for dealing with complex and dynamic gaming datasets. One critique of the study would be including more temporal aspects of sales data, which would have likely added depth to long term sales trends.

Building upon this, Teja et al. (2023) specifically tested Random Forest and Decision Tree on gaming platform Steam’s data using features such as developer reputation, presence of publisher, and user’s engagement. Yet they failed to establish generalizability of their findings, which showed that these factors are indeed important. For instance, let’s say that their dataset comes from Steam, and Steam may not be representative of sales on other platforms, like PlayStation or Xbox. Furthermore, I critique that their adherence to user engagement metrics could be criticized for not accounting for external factors such as seasonal demand or macroeconomic conditions.

To do this, Lee et al. (2021) came up with a new approach where they included temporal data into their Gradient Boosting Machines (GBM) model to evaluate how early marketing campaigns affect game sales. The results of their research proved that pre-release buzz and early user reviews had a very strong predictive power. The study however fails to take into account how effective marketing strategies can be across different genres of games or with different target audiences. Additionally, temporal data provides additional depth to their analysis, but freshness of data becomes a challenge in a rapidly changing world of video game industry trends.

These studies collectively point to the promise of machine learning for sales prediction, while identifying several opportunities for future work, including bringing cross platform data together, consideration of external factors, and refining temporal models to capture patterns that evolve over time. On the basis of this critique, further investigation in the current research is set forth by providing a nuanced understanding of the methodologies used and two implications they elicit.

## Impact of early access and release strategies

From two years of gaming industry has increased from 27.4 million to 33 million peak concurrent Steam users worldwide (Clement, 2024). Game sales performance at launch can be greatly impacted by these strategies and how developers and publishers’ approach initial game release. There are other ways to success; early access, staggered releases and timed exclusives are all different paths and each has its own pros and cons. In order to help develop some of the variables we can then understand how these strategies effect game sales and then be able to form some release strategy to get the most out of them in terms of their maximum profitability and longevity.

### Early Access: Opportunities and Risks

Early access is a practice, when the developer puts the unfinished version of the game in the hand of the public in exchange for money and feedback (Varghese et al., 2022). This is a truly suited strategy for indie developers, as now they could continue working on the game without releasing it and at the same time make revenue from the game. Early access studies, like the one by Lin et al. (2018), suggested that offering early access can lead to massive early sales that help developers promote their game before the full release date. This helps early adopters provide feedback that the developers can address and improve gameplay to help raise the likelihood of success when the game goes live.

Early access also has encountered with some risks. The key challenge is keeping player expectations upfront and releasing an unfinished game can send you into negative reviews and first impressions. According to Ozalp (2024), improving a game with a low early access rating can lead to recovery, but not always. The bugs and incomplete features can harm early access as resulted in disappointed user experiences and it can hinder the game for long term success. Early access can be a financially and developmentally positive endeavour, but first developers must first consider the state of their game.

### Full Release: Traditional versus Staggered Launches

The full release strategy includes the release of a fully polished version of a game to the public. It generally goes hand in hand with heavily marketed initial sales which creates lots of anticipation. But, large publishers like this model as it gets them to limit their resources to only one, coordinated release date. For example, Call of Duty: For Modern Warfare II, the game went for full release, which involved a multimillion-dollar marketing campaign of global advertising and even famous people who were all part of the campaign. As a result, this approach produced considerable first week sales proving creating an early impression, and garnering positive impressions and reviews for sustaining long term sales momentum (Hukal et al., 2022). Brunt et al. (2019) noted that traditional full releases generate an immediate spike in sales and early impressions and reviews matter more for long-term sales performance.

More measured approach comes through staggered releases, where a game is released slowly. This lets developers test the game in smaller markets in order to develop the game once before a full global launch. For wide release, Politowski et al. (2021) found that staggered releases will help with fine tune gameplay and to solve issues, maybe to have more successful wide release. Though, staggered releases can also dampen excitement and increase the risk of piracy because some gamers may decide to play the game illegally in regions before it is released.

### Timed Exclusivity & Subscription Editions

Timed exclusives deals are another common game release strategy that are emerging, similar to the release that is available only on a specific platform for a short duration. They can guarantee developers some type of financial security upfront payments by platform holders like Epic Games or PlayStation. But as Brunt et al. (2019) point out, conditioned exclusivity can alienate parts of the player base, particularly those who don’t have the game at the outset. Lower sales are the result of a widely sold and reviewed game.

It’s also opened up a new way for developers to release games through subscription models like Xbox Game Pass. Games spicing up subscription services will allow developers to get their work out there and not just rely on individual sales. These often raise less upfront revenue from game sales but because they have licensing fees and better visibility, especially for smaller titles, they run relatively stable (Visconti, 2021).

### Pricing strategies and their Impact on sales

Pricing strategies play a key role in commercial success of a game, especially in a crowded digital marketplace where the availability of queuing pricing flexibility, promotional tactics and discounting behaviours determine the game’s lifecycle and profitability. Even for platforms like Steam, there is huge opportunity in having an effective pricing strategy for a game affect its perceived value and ultimately sales performance.

### First Pricing and Consumer Perception

The decision of the initial price of a game affects the consumer’s expectations and the value the game is thought to have. According to Josef et al., (2022) a game’s price should be consistent with its genre, quality and market positioning. When AAA publisher launch at premium prices, that’s primarily due to the amount of time spent on development and having excellent content (Parlan, 2017). But indie games are too often set lower prices to widen the market and ease of access. High priced games with no justification may be underperformance due to consumers turning to competitors. On the other hand, under-pricing may signal poor quality, after all, which will discourage buyers even if the game represents a good deal.

Pricing also influences the game’s market perception in addition to consumer’s expectation. The risk with a mispriced game is to lose key early sales and momentum, essential to long term success. Games with prices perceived as fair are studied to achieve more positive reviews (Josef et al., 2022) which in turns boosts sales. This means that developers need to think about not only the pricing dynamics in their genre, but also in the market, as they want to avert damaging consumer trust and losing revenues.

### Short Term Effects of Discounts

Since it’s a widely used strategy in order to bring in more sales, particularly during promotional events like the Steam Sales. Andrews et al. (2014) find that while games often experience sales increases around these events, games that offer a significant discount say greater than 50% tend to be those with the biggest spikes in sales. They also promote discounts not only attracting the price-sensitive buyers, but they can also improve visibility of the game on a platform, which can lead to organic traffic and extra purchases.

However, frequent discounting could come with long term results. Tobon and Abril (2024) warn that a reliance on discounts is a double-edged sword, conditioning consumers to the expectation of additional price reductions, reducing the number of full price sales and reducing overall profitability level. Indie developers in particular catch the brunt of this phenomenon, called price erosion, because they tend to rely less on discounts than their big brother counterparts. Discounts are a great way to get quick revenue, but they must be paired with the fact that we need the game to have long term perceived value.

### Bundles and Tiered Pricing

The other important pricing strategy that you bundle games together to actually improve sales performance. They (bundles) let developers introduce lesser-known titles to a wider audience and push total sales volumes in the right direction. According to Tobon and Abril (2024), development with a large catalog makes bundling particularly effective, enabling developers to expose the consumers to many games, and potentially boosting the visibility of their less popular game titles. Developers often view bundles as being extremely valuable, and can also clear inventory while earnings money.

Another strategy to generate revenue is tiered pricing, wherein games are sold at different price points (editions standard, deluxe, ultimate) depending on their level of frills. Studies demonstrate how this approach is successful as a way to provide developers opportunities to discuss with both budget conscious buyers and buyers who are willing to pay more to add extra content or exclusive features. This strategy keeps the game a premium option, while allowing for different segmentations (Tobon and Abril, 2024).

### Free to Play and Microtransactions

The use of free-to-play (or F2P) model has not only been changing the pricing strategies in the industry but mostly for multiplayer and mobile games. Revenue for F2P games comes from in game purchases (micro transactions) where players can play the base game for free, and purchase the additional content (Flunger et al., 2021). While according to Rizani et al., (2023), Fortnite charge for the core game, games make substantial revenue from their F2P revenue streams to a large player base. But, while very profitable, developers must strike a balance between which in game purchases can be made and which players can survive.

## Sentiment Analysis and Text Mining for Sales Prediction

Text mining and sentiment analysis have played a vital role in the understanding of opinion and prediction of market trends for all gaming industries. Researchers can extract important information about consumer preferences and behaviour through analysis of user generated content, such as reviews, comments, social media posts. These techniques have been used extensively to predict sales in sales prediction models that help decode the fine details of customer feedback and sentiment, both of which are game changers in product success. Despite the large amount of work done on sentiment analysis and mining text for many products, applying these same techniques to prediction of Steam game sales is still very understudied (Yuan et al., 2017).

There are various studies that proven how the sentiment analysis can affect sales in different industries. Fan et al. (2017) used product reviews posted on Amazon to predict a product’s future sales performance. Sentiment analysis techniques were used to determine whether the polarity reviews had a positive or negative sentiment, and it turned out that the more positive the sentiment of the reviews, the more likely were sales to increase. Sharma et al. (2019) examined the relationship between online book reviews and the sales of books on Amazon and found that the volume as well as the valence of reviews has as a significant impact on sales figures. The research they did highlighted the need for customer feedback to predict to sales and how positive reviews leads to more visibility and trust from consumers. This work opens the door for sentiment analysis to become a component of predictive models.

For the gaming industry, user reviews really have a special influence in building a game’s reputation and attracting possible buyers. The quality of games on platforms such as Steam can be judged by user feedback, ratings and reviews, which can instead be considered proxies for product quality. Previous work by Jang et al. (2019) showed that user sentiment can predict the download volume and revenue generated from mobile games. The games in this study were for mobile and did not address PC games on platforms such as Steam where the game ecosystem and user demographics are much different. Thus, although sentiment analysis has had some success in other domains, this space for the use of sentiment analysis to predict sales for Steam games has not been fully explored.

### Sales Prediction with Text Mining Techniques

For deriving insights from large volumes of unstructured text data text mining techniques such as natural language processing (NLP), modelling, and feature extraction have been used. Shudapreyaa et al. (2024) carry out a comprehensive study of text classification, covering many of the methods used for analysing how sentiment is expressed in text data. This work showed the effectiveness of feature extraction techniques, including term frequency-inverse document frequency (TF-IDF) and n-grams, for extracting the underlying sentiment and core themes from user reviews. Similarly, Zhang et al. (2019) employed text mining on social media data to formulaic demand for items in the retail area utilizing NLP strategies for categorizing customer viewpoints and exhibiting market patterns.

Limited text mining in the context of video games has been used to predict sales performance, prompting a gap in the literature. Shudapreyaa et al. (2024) likewise examined how to use topic modelling to analyse “Steam” user reviews and discover the essential themes that can interfere with satisfaction. However, their work did not connect the themes in their study to sales performance, suggesting avenues for additional research. This gap is filled through advanced text mining to extract meaningful features from Steam reviews and integration with a sales prediction model.

### Existing Research Gaps and Proposed Contribution

Sentiment analysis and text mining are successfully applied to a variety of fields to predict sales performance but limited to the context of Steam games. The PC gaming market in general, namely as it is served by Steam, remains surprisingly unexplored compared to mobile games and other retail products. In addition, most prior work has only focused on whether a review is positive or negative, but has not examined finer sentiment nuances like the intensity of sentiment, different feedback theme, etc. along with a range of other review characteristics (review length, helpfulness rating).

Furthermore, in many cases previous research has neglected the combination of sentiment analysis with predictive features such as the pricing strategy, developer reputation, and user engagement metrics. However, as other market factors are omitted by existing sales forecasting models because of this lack of comprehensive modelling, their predictive power has been limited. For instance, positive sentiment is indicative of potential sales growth, however it needs to be combined with pricing strategies and promotion to give a whole picture of how sales are doing (Shudapreyaa et al., 2024).

In light of these gaps, this work attempts to fill the gap by developing a more integrated approach to predicting Steam sales performance using sentiment analysis and text mining. The proposed model attempts to provide deeper and more accurate prediction of game sales by blending textual features obtained from user reviews alongside numerical attributes, including developer reputation, pricing, and user engagement metrics. The textual data will be advanced NLP processed and captured with sentiment intensity analysis and topic modelling, to go beyond simple polarity assessments. As result, it will offer a better concept of how the user sentiment affects the performance in sales and it will result in more valuable information for the industry and bigger accuracy of sales forecasting models for Steam games.

## Critical analysis

|  |  |  |  |
| --- | --- | --- | --- |
| **Article** | **Work Done and Findings** | **Limitations and Research Gaps** | **Contributions** |
| Enhancing churn forecasting with sentiment analysis of Steam reviews | Sentiment analysis of Steam user reviews is integrated into churn forecasting models in this study. The dataset of 12,000 user reviews across four game types is used. They analyse how sentiment polarity can be extracted to create a score tied to time series data. Vector Autoregression and SVM is applied to solve the CF model, the resulting accuracy is promising (89%) (Rahman et al., 2024). | Sentiment analysis and time series data are the focus of the study, with other possible predictive features such as the in-game behaviour or pricing strategies not taken into account. None of these glances at the effect of external variables like promotions, updates, or user demographics on churn rates (Rahman et al., 2024). | Demonstrates that adding sentiment analysis to existing churn forecasting models is effective. It highlights the use of user reviews as a predictor of churn, which can be very useful for Business in deriving insights from customer dissatisfaction and subsequent improvement of retention strategies. |
| A Novel Approach to Predict Success of Online Games Using Random Forest Regressor for Time Series Data | The article presents a novel system a Random Forest Regressor to predict the success of online games given initial data from the game’s release. They evaluate different time series and machine learning models through metrics such as Twitch viewers and viewer ratio then. The goal of this is to shed some light on game performance for the developers (Varghese et al., 2022). | The study is limited to online games, it does not apply to other genres, like single player or mobile games. It also heavily relies on post release data to be applicable to pre-release prediction. In addition, it doesn’t include user sentiment in reviews or feedback (Varghese et al., 2022). | Using time series data and Random Forest Regressor to provide a predictive model for game success and as a way for developers to determine the success of their games and reiterate their marketing strategies. It highlights the importance of player engagement metrics in the early stage, when predicting game success. |
| Using deep learning and steam user data for better video game recommendations | This paper presents STEAMer, a recommendation system based on deep autoencoder model instructed by Steam user data to facilitate better game recommendations. By integrating additional user data with an existing deep neural network-based system, it demonstrates improvements in MAP@10 and NDCG@10 scores as well as diversity of recommendations (Wang et al., 2020). | This study is only about making recommendation accuracy as high as possible without paying much attention on the direct effect of the recommendations to sales or user retention. And it doesn’t take external factors that affect game recommendations into account, including seasonal promotions or developer activities. | It also presents a new deep learning-based approach to video game recommendations in Steam which significantly outperforms traditional methods. The main point is that recommendation systems can benefit from user data to increase recommendation accuracy and increase customer satisfaction (Wang et al., 2020). |
| An empirical study of game reviews on the Steam platform | Following this study analyses empirical reviews from 6,224 games on Steam with respect to what game reviews are and how they compare to mobile app reviews. Using quasi experiments that modify game attributes, it investigates the relation between game attributes that cause review fluctuations, as well as user concerns expressed in reviews (Lin et al., 2018). | For this review, the research is focused mostly on the analysis of reviews without sentiment analysis techniques followed by advanced text mining to understand users’ sentiment more deeply. In particular, it lacks a longitudinal analysis to examine the evolution in review characteristics over time. | Highlighting the distinct properties of the game reviews vs. the mobile app reviews and offering information to developers regarding review patterns and user’s feedback. It suggests that prior mobile app review studies may have to be re-examined for particular game contexts (Lin et al., 2018). |
| Game Data Mining Competition on Churn Prediction and Survival Analysis Using Commercial Game Log Data | This paper reports on an international competition on predicting churn with game log data from Blade & Soul. A dataset of 100 GB of logs of 10,000 players was used, where the competition had to predict player churn and survival analysis. Deep learning, tree boosting and linear regression were among top methods used (Lee et al., 2019). | The study is based on data from only a single game; therefore, it restricts its applicability to other instances. The requirement for large-scale datasets is also a plus in focus on commercial game data, which turns off indie developers with limited access to data. | It serves as a platform for applying current top notch data mining techniques on game log data, exemplify the significance of open data availability in enhancing research in churn prediction. In fact, the study highlighting the need for collaboration between industry and researchers is important (Lee et al., 2019). |

## Conceptual framework

Below is the conceptual framework including independent and dependent variables that form hypothesis

A diagram of a game

Description automatically generated

**Figure 1 conceptual framework (Self-Created)**

The conceptual framework visualizes (Figure.1) the relationships among sentiment analysis along with text mining as independent variables and their influences on the sales prediction, game performance, and marketing strategies effectiveness. It showcases the hypotheses:

* H1 - Sales Prediction Accuracy is positively influenced by Sentiment Analysis Metrics.
* H2 - Sales Prediction Accuracy is positively influenced by the Text Mining Features.
* H3 - Game Performance Metrics improve Sales Prediction Accuracy.
* H4 - More effective marketing strategy decisions are made due to Enhanced Game Performance Metrics.

The text data analysis framework outlines how text analysis in gaming leads to the decision making of sales and strategic business decisions.

## Summary

This literature review discusses important gaps in previous research related to sentiment analysis and text mining in the gaming industry for sales prediction. Previous studies have shown success using sentiment analysis for churn forecasting and text mining to understand customer feedback, but they did not combine these methods for direct sales predictions. Additionally, most prior models isolate whether a user is engaged or have dedicated research toward developing their recommendation systems without exploiting the synergy of user sentiment and textual feature for sales predicting. In order to fill this gap, this project attempts to create a complete model in a combination of sentiment analysis and text mining to predict sales of a game, giving a more holistic view of how sales are predicted.

# Methodology

## Research approach

This project research approach adopts a quantitative approach measuring numerical data to identify patterns and relationships in relation to sales performance of Steam games (Sattar et al., 2017). The reason for using such an approach is however that the data work upon is structured and measurable such as developer information, platform availability, available publisher details and sales figures which makes them all suitable for statistical and machine learning analysing. Quantitative methods are used to model objectively these variables that influences game performance on a platform with billions of potential buyers like Steam and ability to predict sales outcomes. For this project, machine learning models, a fundamental part of this project, are primarily based on a quantitative framework to process an enormous quantity of data to draw meaningful conclusions from it (Sattar et al., 2017). This research can quantify how many of the attributes (e.g., developer reputation, platform presence, publisher backing) affects game sales by employing algorithms such as Random Forest, Gradient Boosting, LSTM. In addition, the use of regression metrics such as MSE, R2 Score provides an objective evaluation of model regression, which ensures that the results are based on measurable outcomes. On the dataset, the quantitative approach is not only appropriate, but it is necessary to derive meaningful insights from the dataset to inform actionable decisions about game releases, marketing strategies and platform engagement. By providing stakeholders with reliable predictions for an ever-changing gaming market, this empirical analysis will provide stakeholders with valuable insight upon which to base their decisions based on evidence. In this project we will follow a structured step-by-step way to achieve research goals, the process underscores some key methods shown in Figure 1:

## Data Collection and Data Description

**Dataset Link**: <https://huggingface.co/datasets/FronkonGames/steam-games-dataset>

This dataset contains information about video games released on the Steam platform, capturing a wide range of attributes related to the game's performance, pricing, and user engagement. This project is using the publicly available Steam Games dataset hosted on Hugging Face. The video games released on the Steam platform are collected along with all the game details, game developer, game publisher, game release year, game price, game sales figures, users rating for the game and other relevant attributes in this dataset. The dataset has 80000 rows, 39 columns to provide the best data on each game to capture which factors are responsible for game sales performance.

The primary data used in this study will focus on the following key attributes -

* Developer: The creator of the game.
* Publisher: The entity which should distribute the game.
* Platform: Available platforms (Windows, Mac, Linux) on which the game is delivered.
* Sales performance: The variable we will predict for, the estimated number of owners or sales for each game.
* Categories: Its popularity and sales can be influence by the game genre or category (Action, Adventure, RPG).

The dataset was first processed and columns containing relevant data were extracted, then we did initial cleaning on the missing data or inconsistent data. Next it was pre-processed and transformed, encoding categorical variables and normalizing numerical features in order to develop machine learning model. Since, game sales performance can be predicted using

machine learning techniques this dataset was chosen as its extensive in nature and relevant to problem statement as it encapsulates the factors likely to influence game sales.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| AppID | Name | Release date | Estimated owners | Peak CCU | Required age | Price | DLC count | About the game | Publishers | Categories | Genres | Tags |
| 0 | 20200 | Galactic Bowling | Oct 21, 2008 | 0 - 20000 | 0 | 0 | 19.99 | 0 | Perpetual FX Creative | Perpetual FX Creative | Single-player,Multi-player,Steam Achievements,... | Casual,Indie,Sports |
| 1 | 655370 | Train Bandit | Oct 12, 2017 | 0 - 20000 | 0 | 0 | 0.99 | 0 | Rusty Moyher | Wild Rooster | Single-player,Steam Achievements,Full controll... | Action,Indie |
| 2 | 1732930 | Jolt Project | Nov 17, 2021 | 0 - 20000 | 0 | 0 | 4.99 | 0 | Campião Games | Campião Games | Single-player | Action,Adventure,Indie,Strategy |
| 3 | 1355720 | Henosis™ | Jul 23, 2020 | 0 - 20000 | 0 | 0 | 5.99 | 0 | Odd Critter Games | Odd Critter Games | Single-player,Full controller support | Adventure,Casual,Indie |
| 4 | 1139950 | Two Weeks in Painland | Feb 3, 2020 | 0 - 20000 | 0 | 0 | 0 | 0 | Unusual Games | Unusual Games | Single-player,Steam Achievements | Adventure,Indie |

A diagram of a software development process

Description automatically generated with medium confidence

**Figure 2 Block Diagram (Self-Created)**

## Data pre-processing

Pre-processing plays an important role in the machine learning pipeline, it means that the dataset is clean, structured and is ready to build the model. In this project, the raw Steam Games dataset will be made to pass through a few steps of pre-processing: handling missing values, inconsistencies in the data, and making the data ready for machine learning. The key steps in the data pre-processing process are as follows:

* **Handling Missing Values:** Methods for identifying missing values in the dataset and for handling them will be used accordingly. When dealing with numerical columns missing data is either imputed by using median, mean, or mode, or rows with too many missing data are removed to have a clean dataset (Abonazel, 2020). In a case of categorical columns, the missing categories are filled with the most frequent category (or labelled 'Unknown' if applicable).
* **Removing Duplicate Entries:** If any duplicates exist it will identify and remove duplicate records in a way that will not skew the model’s learning process. This makes sure that each game in the dataset has its own unique and doesn’t have redundant data.
* **Handling Outliers:** Outliers of numerical features (e.g., sales numbers, prices) may spoil the performance of the model. We will use statistical methods such as Z-scores and boxplots to identify extreme values. Specifically, if these outliers have a large impact on the data distribution, they can be transformed, or removed.
* **Encoding Categorical Variables:** The dataset has many categorical (developer, publisher, categories, platforms) attributes. We will code these into numerical values by label encoding so they can now be fed into the machine learning algorithms (Abonazel, 2020).
* **Feature Engineering:** More insightful information can be derived from the existing columns adding additional features. For example, the "review score" column to create a "rating" feature or the "release date" to compute the game’s age since release. However, these new features could also provide more meaningful patterns which will assist the predictive power of the model.
* **Splitting the Dataset:** Pre-processing of the dataset will be done and the dataset will be split into training, validation and test sets to be carried out correctly when testing the model. This will be a typical 80% training and 20% validation. To tune the hyperparameters, validation set are used and to check the final model’s performance we will use the test set.

## Exploratory Data Analysis (EDA)

An exploratory data analysis (EDA) was undertaken to discern patterns, relationships, and insights in the dataset itself, across the game attributes, user generated reviews, and sales performance metric dimensions (Raman et al., 2022). Histogram and kernel densities plots were used to analyse the distribution of key numerical variables such as price, estimated owners, achievements, and average playtime, such that trends and outliers can be identified. The user review sentiment was visualized through bar charts that revealed the percentage of positive, negative and neutral user reviews using VADER sentiment analysis. Genres and categories were split and exploded from the multi-valued column using the analysis of genres and categories, then we counted the frequencies and plotted bar charts to identify the most popular genres and categories. Numerical features were correlated using a heatmap to identify correlations between variables such as price, achievements, and sentiment scores to provide an indication of how these related to sales.

Positive, negative, and neutral reviews were converted to word clouds generating commonly used words, and the frequency analysis of terms in each sentiment category. Furthermore, scatter plot and box plot were used to study the relationship between sales (estimators of owners) and different features such as seller’s price, sentiment scores, publisher and developer details. EDA was used to understand the data well enough to better select features and develop the model, while simultaneously imparting important knowledge around the factors that driven game sales performance (Raman et al., 2022).

## Model Development

Through the model development phase, we attempted to predict the sales performance of Steam games from game attributes and user reviews. We used both traditional machine learning algorithms and deep learning models on this phase to evaluate and pick the most effective way. We started feature engineering by encoding categorical variables developers, publishers, genres and categories as label encoded features to make them numerical for machine learning models. The dataset was split into training and testing sets, with 80-20 ratio and standardised the values using StandardScaler to standardize the data for better model performance.

Following widely accepted machine learning best practices, the 80-20 training set split was used to balance training and testing. Allocating 80% of the data for training ensures that the model has sufficient data to learn patterns and relationships, while 20% is sufficient to evaluate the model's performance on unseen data. This ratio strikes a balance between reducing overfitting with a robust training set and reliable evaluation metrics, making it beneficial for predictive modelling.

Three Machine learning models will be implemented here to predict game sales: Random Forest Regressor, Decision Tree Regressor, and Gradient Boosting Regressor (Joshi et al., 2021). Grid search and cross validation will be used for hyperparameter tuning that will happen on each model to optimise the parameters. Metrics including mean squared error (MSE) and R-squared (R²) will be used to perform performance evaluation so that we can pick the most effective model.

Then, we will work on deep learning level by generating Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) models (Joshi et al., 2021). We will use timestamps in the data, form a temporal dependency layer for LSTM model and structure the layers for processing of sequential patterns. LSTM and dropout layers, along with dense layers to predict. The convolutional layers will be used to recognize tuff patterns in the input data followed by dropout layers for regularization and fully connected layers for running final output prediction. When compiled, both models will take the Adam optimizer, MSE loss and MAE performance metric (Gangwar et al., 2023).

Reshaping of input data for LSTM and CNN models till they satisfy their input requirements; this will make part of the training process. Above 50 epochs, with batch size of 32, the models will be trained and validated over 20% of the training data. Then their performance will be evaluated on test dataset and check their model using metrics like loss and MAE in order to analyse the predictive accuracy.

Then comparing the results of the machine learning and deep learning models predicting Steam game sales performance in order to identify the most effective method. With this methodology, we ensure a complete analysis as the merit of various modelling techniques is combined to produce accurate and reliable predictions.

## Model Evaluation

This will use an integrated evaluation framework to assess the performance of the developed models. For all of the implemented machine learning models including Random Forest Regressor, Decision Tree Regressor, Gradient Boosting Regressor, we will look at the key metrics like Mean Squared Error (MSE) and R-squared (R²) (Gangwar et al., 2023). To measure the models’ capacity to minimize prediction errors and comment on the variance across the target variable, these will constitute the metrics. We will also run the cross validation to validate whether the results presented generalize to different data such splits.

Then evaluating performance on deep learning models such as LSTM and CNN with Loss (MSE) and Mean Absolute Error (MAE) will perform with training and test datasets. Then the process will look at the validation loss and MAE during training to look out for overfitting, so these models should also generalize well on unseen data. We then visualize the performance of these models’ using loss and accuracy plots, showing how learning curves and convergence change over epochs (Gangwar et al., 2023).

Using the results of all models, we will then be able to see what the best prediction model for Steam game sales. The process of the evaluation will mature to make sure that we select a model that has high predictive accuracy, generalizability and robustness for examination.

# Requirements

To ensure the successful execution of this project, the following requirements are categorized into datasets, tools and technologies, computational resources, and expertise:

1. **Dataset Requirements**

* **Game Data:** The identification of a comprehensive dataset of Steam games with over 80,000 entries and over 39 columns can be used for this. Among the game details are the price, genre, developer, publisher, the average reviews, estimated number of owners, and categories.
* **Data Quality:** The dataset has to be clean for missing and inconsistent values. Columns with large amounts of missing data will either be excluded or imputed if needed.
* **Sentiment Analysis Input:** Extracting sentiment scores and sentiment categories positive, negative or neutral requires that reviews or textual data from the users have to be available.

1. **Tools and Technologies- Python Libraries**

* **Data Preprocessing:** Pandas, NumPy
* **Data Visualization:** Any library like matplotlib, seaborn, wordcloud
* **Natural Language Processing:** An DAPI of (tokenization, lemmatization, removal of stop words, sentiment analysis using VADER) using NLTK
* **Machine Learning Models:** random forest, decision tree, gradient boosting on scikit learn.
* **Deep Learning Models:** (LSTM, CNN for prediction tasks with Tensornfow/Keras)
* **Development Environment:** For ease of implementation and visualization, Jupyter Notebook or Google Colab.
* **Model Evaluation:** mean squared error (MSE), R Squared (R2), mean absolute error (MAE), cross\_validation (Gangwar et al., 2023).

1. **Computational Resources - Hardware**

* In order to train machine learning and deep learning models, we need an 8GB or more RAM and multi core processors computer.
* It is recommended to train computationally intensive models such as LSTM and CNN efficiently through access to GPU resources.

1. **Software**

* Implementation requires Python 3.9 or higher, with all required libraries pre-installed.
* Integrated development environment (IDE) like PyCharm, VS Code or Jupyter Notebook.
* Storage: The dataset and intermediate outputs from pre-processing and model training needs at least 50GB of free storage.

1. **Expertise**

* **Domain Knowledge:** Ability to interpret the results well dependant on familiarity with game development and user preferences.
* **Programming Skills:** Experience in python especially working with libraries for data manipulation, visualization and machine learning.

1. **Machine Learning and Deep Learning**

* Ability to find feature selection, feature scaling, as well as hyper parameter tuning.
* Training, evaluating and interpreting machine learning and deep learning models experience.
* Natural Language Processing (NLP): Familiarity with the text preprocessing techniques such as tokenization, lemmatization and sentiment analysis.

1. **Ethical Considerations**

* **Data Privacy:** It should also include sensitive nor personally identifiable information in the dataset.
* **Transparency:** All code and data transformation must also be documented so they can be verified and results must be reproducible.

# Implementation

This project implementation phase consists of step-by-step sequence of the tasks as designed in the methodology. It helps you quickly transform data pre-processing, exploratory analysis, and machine learning as well as deep learning model development scenarios into a process and architecture.

## Data collection and loading

Data collection and loading is the first part of the project and an essential phase which establishes the base for the following analysis. There are more than 80,000 records with 39 features and the dataset consists of various aspects of Steam games like Featured, Game Name, Developer, Publisher, Price, Genres, Categories, Estimated Owners, Release Date, Reviews etc. This dataset covers all game sales performance factors comprehensively. The selected dataset is publicly available Steam Games dataset hosted on Hugging Face; this data is stored in a CSV format which as such allows to seamlessly integrate with python-based data analysis tools.

The first step is to import all necessary libraries and then to simply loading the dataset into the environment using the Pandas, a powerful Python library for data manipulation (Figure.3). The first process of loading the dataset is to ensure the structure of the dataset: it includes column names, data types and row counts. This inspection tries to identify inconsistencies such as missing values, duplicate entries or wrong data formats. For summary statistics we generate down an overview of numerical and categorical features. We also begin to prepare the Reviews column for further text pre-processing by examining an initial exploration of the review’s column. It is, as the integrity and quality of the loaded data is directly related to the accuracy and reliability of the models produced later in the project.

A screenshot of a computer program

Description automatically generated

A close-up of a text

Description automatically generated

**Figure 3 Loading library packages and reading dataset (Self-Created)**

## Data Pre-processing

**Text Pre-processing:** Text preprocessing steps are applied to reviews in the dataset to make the reviews consistent, and then they feed to the sentiment analysis model (Figure.4). The first thing that we do is we get the reviews and we tokenize it, so it is made up of individual words or tokens. Then these tokens are filtered further by removing stop words (common words like "and," "the,” "is”) and punctuation, which don't convey an important meaning for the analysis carried here. Then a WordNetLemmatizer is applied to morphologically reduce the words to their base (or root) forms. For instance: 'running' is 'run', 'better' is 'good.' That means that the same words that mean the same thing are considered to be the same in this step, thus, it helps the model better understand the text and reduce redundancy in the dataset.

A screenshot of a computer program

Description automatically generated

**Figure 4 Cleaning reviews, Pre-processing (Self-Created)**

**Sentiment Analysis:** To do Sentiment analysis, we use VADER (Valence Aware Dictionary and sentiment Reasoner) sentiment analysis tool that give sentiment scores for each review (Figure.5). This second standard provides the VADER as a compound score, being a value of -1 (most negative) to 1 (most positive), which represents the overall sentiment of the text. Based on the sentiment score, each review is classified into one of three categories: It can be Positive, Negative or Neutral. Reviews that receive scores > 0.05 are considered Positive, scores < -0.05 are Negative, and anything in between is Neutral. That categorization enables a structured analysis of how different kinds of sentiment relate to game features such as price, developers or genre (Figure.6).

A screenshot of a computer program

Description automatically generated

A computer screen shot of a computer code

Description automatically generated

**Figure 5 Sentiment Analysis (Self-Created)**

A screenshot of a computer code

Description automatically generated

**Figure 6 Count of sentiments (Self-Created)**

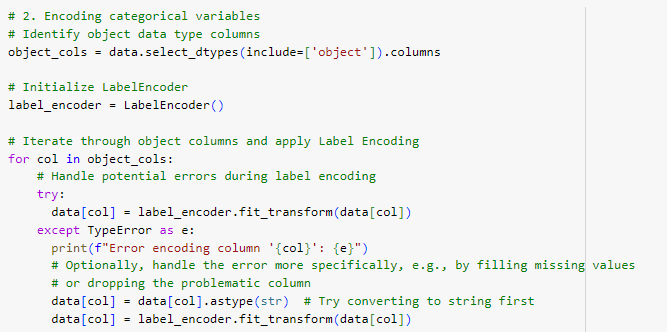
**Data Cleaning:** In the case of the data cleaning process, missing values are dealt with carefully. We fill in missing categorical columns like 'Name', 'About the game', 'Developers', 'Publishers' etc. with a default placeholder (which can be anything but consistent) like "Unknown" to keep the dataset homogenous (Figure.7). Missing values in numerical columns, i.e., 'Estimated owners', are filled up with values computed in a way to avoid biases in the data (such as a mean or median). Moreover, the columns of the dataset containing the large percentage of missing values or making little input to the analysis such as 'Reviews', 'Website', 'Support URL' and 'Metacritic URL' are truly dropped in the dataset for use of only useful and trustful data in the further analysis.

A close-up of a computer code

Description automatically generated

**Figure 7 Data Pre-processing (Self-Created)**

**Feature Encoding:** Encoding techniques are applied to make categorical features machine learning models friendly (Figure.8). Label Encoding is used to transform Categorical columns like 'Developer' or 'Publisher' etc. into numerical values. This method does one important thing, converting the text-based data into a format that can be consumed by Machine Learning Algorithms and assigns each category a unique integer. To illustrate, the model assigns each developer their own unique number, so it can learn what relationships between different developers, publishers, genres, and categories without the model knowing the original textual representation.



**Figure 8 Label Encoding (Self-Created)**

**Feature Scaling:** Feature scaling is to normalize the numerical feature and to avoid one feature’s scale should overshadow the model’s performance (Figure.9). We standardize the numerical features: StandardScaler did the trick with 'Price', 'DLC count', 'Positive' sentiment score and 'Negative' sentiment score. All the features are scaled so that each feature has an average of 0 and a standard deviation of 1 using this technique. It is particularly important if learning models are based on distance calculation, or gradient based optimization and our features scale is not standardised.

A screen shot of a computer code

Description automatically generated

**Figure 9 Train and test split and Normalization (Self-Created)**

## Model Implementation

This project develops several of the machine learning models and deep learning architectures to predict the sales performance of Steam games given many such features including developer, publisher, genres, price, and user sentiment.

### Machine Learning Models

**Random Forest Regressor:** Random Forest resists overfitting and handles large, high-dimensional datasets, the Random Forest Regressor was chosen. The model averages predictions from many decision trees trained on random subsets of data to prevent overfitting and increase prediction stability (Ao et al., 2018). Since developer, publisher, and genre relationships are nonlinear, it is great for sales prediction. Random Forest predicts sales. It captured complex interactions between location, size, and amenities when Ao et al. (2018) estimated property prices. Random Forest can estimate e-commerce sales based on user ratings, pricing, and promotions, showing its versatility in feature-rich settings. The paradigm is computationally intensive, especially with multiple trees, and less interpretable than Decision Trees.

**Decision Tree Regressor:** For simplicity and interpretability, Decision Tree Regressor was used. Hierarchically partitioning data by feature values makes this model ideal for evaluating how user sentiment and price effect sales. While overfitting training data, Decision Trees efficiently capture nonlinear patterns in smaller samples. Predictive analytics employs decision trees extensively. Choice routes reflect individual aspects in retail sales prediction, according to Suthaharan (2015). In another study, Decision Trees predicted video game sales by assessing marketing and pricing methods in feature-rich datasets with nonlinear patterns. Decision Trees are useful, but they can overfit, especially with inadequate data, and be sensitive to dataset changes, producing instability.

**Gradient Boosting Regressor:** Iteratively correcting tree faults improves accuracy, hence the Gradient Boosting Regressor was chosen. This strategy thrives in datasets with complex patterns and feature interactions, making it suitable for predicting Steam game sales affected by user sentiment and developer reputation (Zhang & Haghani, 2015). Sales forecast research has shown Gradient Boosting works. Zhang and Haghani (2015) predicted retail sales using historical data, outperforming simpler models. This model predicts new game sales using marketing budget, game ratings, and competition performance, showing its adaptability to different data sources and interactions. Gradient Boosting is computationally demanding and requires hyperparameter adjustment yet predicts well. Despite its slower training process than simpler models, it can capture complex feature interactions, making it suitable for this study.

Together, these models maximise their strengths. Decision Trees provide interpretability and feature importance, whereas Random Forest and Gradient Boosting handle feature-rich datasets robustly and reliably. Studies have confirmed these models, making them perfect for predicting Steam game sales using multidimensional factors.

A screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated

**Figure 10 Machine Learning model implementation & residuals vs predicted (Self-Created)**

Performance metrics such as Mean Squared Error (MSE) and R-squared (R²) are used to evaluate how good these machine learning models are at predicting the sales performance. Further analysis will be performed on the best performing model (Figure.10).

### Deep Learning Models

Since it can capture temporal dependencies and sequential patterns in data, the LSTM Model was used to predict game sales, which often exhibit time-series behaviour (Guitart et al., 2018). LSTM cells solve the vanishing gradient problem in typical recurrent neural networks by preserving long-term dependencies. This project's LSTM model contains sequential input data processing, dropout layers to reduce overfitting, and dense prediction layers (Figure 11). The model is trained on scaled features using MSE loss and MAE assessment. Time-series data analysis is known to sales forecasters. Learning temporal patterns in prior data, LSTMs have predicted retail sales and finance stock values daily. The computationally intensive LSTM model requires hyperparameter adjustment, but its precision in capturing small temporal correlations makes it effective.

A screenshot of a computer program

Description automatically generated

A computer code with text

Description automatically generated with medium confidence

**Figure 11 Deep Learning Models LSTM (Self-Created)**

Local relationships and complex feature interactions in structured data were revealed by the CNN Model. Conv1D layers enable CNNs analyse structured data over picture and geography (Zhou et al., 2020). The CNN design employs convolutional layers for feature extraction and flatten and dense layers for prediction. This structure allows the model capture complicated feature correlations that simpler models overlook. CNNs excel in finding patterns and interactions in high-dimensional game sales, developer, publisher, and user sentiment datasets (Figure 12). Their independent hierarchical feature representation learning increases prediction performance. Hyperparameter-sensitive CNNs demand plenty of processing resources to train. Their success in retail demand forecasting and recommendation systems suggests that they can capture complex feature connections that machine learning models overlook, making them suited for this study.

A screen shot of a computer code

Description automatically generated

**Figure 12 Deep Learning Models CNN (Self-Created)**

Both LSTM and CNN models are compiled with Adam optimizer and trained for fixed number of epochs. Loss and Mean Absolute Error (MAE) is used for performance evaluation.

### Justification for model selection

The models selected for these tasks are on the basis of their interpretable nature, performance and suitability for the characteristics of the dataset. For simplicity and interpretability of how different features affect sales predictions, Random Forest and Decision Tree Regressor were selected. For this dataset these models as well as fitted with numerical and categorical data so they are appropriate for this dataset. Gradient Boosting Regressor was chosen because it learns iteratively, and that’s valuable in a dataset which have subtle feature interactions that have an impact on predictions.

LSTM is used for the deep learning models because it can capture long term dependencies and temporal relationships that may be a factor in the data as with price changes or game updates. More experimentally is included a model that includes CNN, which can describe spatial and hierarchical patterns in features, and can provide a new way to extract features from categorical variables. Combining these models gives a sound basis for predicting game sales, with traditional and modern machine learning techniques to discover signals from the data.

## Model Evaluation

To assess the performance of the models as well as determine what approach best predicts the sales performance of Steam games, the models need to be evaluated. The models used here for Machine Learning like Random Forest Regressor, Decision Tree Regressor, and Gradient Boosting Regressor are evaluated using Mean Squared Error (MSE), and R square (R²) only. Lower value of MSE indicates better predictive accuracy and measures average squared difference between actual and predicted values. But R^2 tells how much the variance in the target variable has been explained by the model, the greater the value signifies better fit. These are used to evaluate how generalised each model is on the test data and how good each model is at capturing the underlying relationships in the dataset itself.

Loss function (Mean Squared Error) and Mean Absolute Error (MAE) are used for performance evaluation of LSTM and CNN models for data with high dimension and volatility. Loss function gives an overall measure on how far the predicted values are from the actual values and MAE is a simplistic measure of prediction error, it is giving a measure of the average magnitude of error in the predictions. Furthermore, the robustness of the models is checked by cross validating them to prevent overfitting and get more accurate generalization estimate of the model. Evaluation metrics are compared across various models to pick the best model, for which there is clear suitability and ensuring effectiveness to the real-world applications.

## Data Balancing

A screenshot of a computer code

Description automatically generated

**Figure 13 Data Balancing (Self-Created)**

This above code(Figure.13) balances an imbalanced dataset using SMOTE. First, separate input characteristics (X) and target variable (y) from the dataset. To balance a 42 random state, SMOTE generates minority class synthetic samples. The fit\_resample method provides a balanced dataset with equal distribution of classes. Reconstructing balanced data into a DataFrame maintains feature columns and adds the updated target variable. Printing class distribution shows the dataset is balanced.

# Analysis

A screenshot of a computer

Description automatically generated

**Figure 14 Sample Dataframe (Self-Created)**

The above provided image represents a sample dataset containing details about Steam games, including variables such as AppID, name, release date, estimated owners, price, and gameplay metrics (Figure.13). This dataset is a foundational component of your study on predicting game sales performance using machine learning techniques. It offers diverse insights into game attributes like developer/publisher names, supported languages, and user engagement metrics such as average playtime and median playtime. The inclusion of estimated owners and pricing data provides key predictive indicators for analyzing sales performance. The data structure appears tabular, suitable for preprocessing, exploratory data analysis, and predictive modeling to uncover relationships between game features and their market success.

A screenshot of a computer

Description automatically generated

**Figure 15 Processed dataframe (Self-Created)**

The dataset includes two new columns: Cleaned\_Reviews and Sentiment\_Score. The Cleaned\_Reviews column represents processed text data, where reviews are tokenized, stop words removed, and lemmatized for consistency and better analysis (Figure.15). The Sentiment\_Score column quantifies the sentiment of each review, ranging from -1 (negative) to 1 (positive), using tools like VADER for sentiment classification. These additions enable deeper insights into user feedback and its influence on sales performance.

The Sentiment column categorizes reviews based on the Sentiment\_Score, assigning labels such as "Positive," "Neutral," or "Negative." For instance, reviews with a score of 0.0 are labeled as "Neutral." This column provides a quick, interpretable sentiment classification, simplifying analysis of user feedback trends and their potential impact on game performance.

A close-up of numbers

Description automatically generated

A screen shot of a computer

Description automatically generated

**Figure 16 Sentiment distribution before and after balancing Data (Self-Created)**

Before balancing, the dataset's sentiment distribution displays a large class imbalance. Neutral sentiment leads with 75,914, positive sentiment 8,300, and negative sentiment 889 (Figure 15). It appears the sample is substantially skewed towards Neutral. Training machine learning models favours the dominant class, biassing predictions. The model may overpredict Neutral and underpredict Positive and Negative.

Balanced emotion distributions show equal class samples. The Neutral, Positive, and Negative emotions have 8,300 occurrences (Figure 16). This balance ensures the model receives equal examples from all classes during training, promoting fairness and accuracy.

Due to its impact on machine learning model performance, dataset balancing is crucial to this research. The model may overfit Neutral and miss Positive and Negative sensations without balance. Addressing this gap improves model resilience and class forecasting, enhancing metrics.

A close up of words

Description automatically generated

A word cloud with text

Description automatically generated

A close up of words

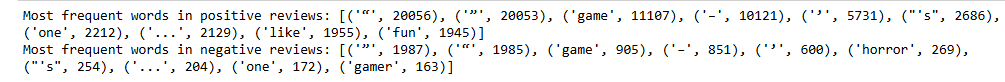
Description automatically generated

**Figure 17 Worcloud for Sentiments (Self-Created)**

The word clouds visualize the most frequent terms in positive, negative, and neutral reviews, revealing user sentiment trends (Figure.17).

1. **Positive Reviews**: Highlight words like "fun," "great," "gameplay," and "experience," emphasizing enjoyment, quality, and engagement in the games. Positive sentiment reflects user satisfaction with gameplay and design.
2. **Negative Reviews**: Words such as "time," "horror," and "new" are prominent, often pointing to frustrations or unmet expectations. Negative sentiment highlights areas for improvement in design or performance.
3. **Neutral Reviews**: Words like "one" and "game" dominate, suggesting a lack of substantive feedback. This highlights the prevalence of reviews without clear opinions.

These insights guide game development and marketing strategies.



**Figure 18 Frequency of words**

The frequent words in positive reviews, such as "game," "like," and "fun," highlight user appreciation for gameplay and enjoyment (Figure.18). In contrast, negative reviews focus on words like "horror" and "one," indicating dissatisfaction or specific issues. Symbols and punctuation suggest a need for text cleaning in further analysis.

A graph with a blue line

Description automatically generated

**Figure 19 Distribution of Game Prices (Self-Created)**

The price distribution of games is highly skewed, with most games priced under $20. A steep drop is observed as prices increase, with very few games priced above $100. This indicates a concentration of affordable games in the market, catering to a larger audience, while premium-priced games are rare (Figure.19).

A green and red bar graph

Description automatically generated

**Figure 20 Sentiment Distribution of Reviews (Self-Created)**

The sentiment distribution shows that the majority of reviews are Neutral, followed by a smaller proportion of Positive reviews, and very few Negative reviews. This suggests that user feedback is predominantly neutral or balanced, with a slight inclination towards positive sentiments, indicating overall moderate satisfaction among the user base (Figure.20).

A graph of a distribution of a number of individuals

Description automatically generated with medium confidence

**Figure 21 Distribution of Estimated Owners (Self-Created)**

The distribution of estimated owners is highly skewed, with most games having a small number of owners, as reflected by the steep peak near zero. A few games achieve significantly higher ownership levels, indicating a long-tail distribution where a limited number of popular games dominate the market share (Figure.21).

A graph with orange bars

Description automatically generated

**Figure 22 Most Popular Game Genres (Self-Created)**

The bar chart highlights the top 10 most popular game genres. Indie games dominate, significantly surpassing other genres, followed by Casual, Action, and Adventure (Figure.22). These genres attract a wide audience due to creativity, accessibility, and diverse gameplay. Genres like Sports and Free to Play are less frequent, reflecting niche market interest or development focus. This distribution emphasizes user preferences for diverse, innovative, and story-driven gaming experiences.

A diagram of heat map

Description automatically generated

**Figure 23 Correlation Map (Self-Created)**

The correlation heatmap illustrates the relationships between numerical features in the dataset (Figure.23):

1. **Positive and Recommendations**: Strong positive correlation (0.90) suggests that higher positive reviews strongly influence game recommendations.
2. **Positive and Negative**: Moderate correlation (0.78) may indicate overlap in review counts but with different sentiments.
3. **DLC Count and Other Features**: Weak correlations suggest minimal impact of downloadable content on sentiment or recommendations.
4. **Achievements and Sentiment**: Negligible correlations imply that the number of achievements does not significantly affect reviews or recommendations.

Overall, positive reviews and recommendations are highly interlinked, highlighting user satisfaction as a critical factor in driving game popularity.

A graph with numbers and lines

Description automatically generated

**Figure 24 Distribution of Average Playtime Forever (Self-Created)**

The distribution of Average Playtime Forever is heavily skewed, with most users spending a relatively small number of hours playing games (Figure.24). A sharp peak near zero indicates many games with minimal engagement, while a long tail suggests a few games with significantly higher playtime. This reflects varying levels of user engagement across games, with most titles seeing limited playtime, potentially due to niche appeal, quality issues, or competition in the market. Games with higher playtimes may represent highly engaging or popular titles. Understanding these patterns can help identify factors influencing player retention and game success.

A screenshot of a computer

Description automatically generatedA screenshot of a video game

Description automatically generated

**Figure 25 Missing Values (Self-Created)**

The tables provide a summary of missing values for each column in the dataset (Figure.25):

1. **Significant Missing Values**: Columns like Reviews (75,360), Notes (72,082), Score Rank (85,059), and Metacritic URL (81,191) have substantial missing data, potentially impacting the analysis. These gaps suggest the need for data cleaning or imputation strategies.
2. **Moderate Missing Values**: Features like Website, Support URL, Support Email, and Movies also contain notable missing data, reflecting incomplete metadata for games.
3. **No Missing Values**: Essential features like AppID, Name, Price, and Recommendations have no missing values, ensuring reliable input for analysis.

Addressing these missing values is critical for accurate insights and model performance.

A black text on a white background

Description automatically generated

A number and numbers on a white background

Description automatically generated

**Figure 26 ML model performance Before and After Balancing Data (Self-Created)**

Figure 26 summarizes the performance of three models—Random Forest, Decision Tree, and Gradient Boosting—evaluated using Mean Squared Error (MSE) and R² score:

All three approaches predict well before balancing, but Gradient Boosting outperforms. Gradient Boosting has the lowest MSE (1.989983×1011) and highest R² score (0.839287), capturing 84% of target variable variation. Although Random Forest had a similar R² score of 0.779140, Decision Tree did poorly with 0.706426. The highly skewed dataset, especially Neutral sentiment's dominance, gave models tonnes of training data. After dataset balancing, all models failed. All three algorithms' MSE climbed dramatically, with Random Forest and Decision Tree reaching 1.4×1012, while Gradient Boosting had a somewhat lower figure of 1.250.476×1012. Gradient Boosting had the greatest R² value of 0.494795, followed by Random Forest and Decision Tree with 0.435584 and 0.425431, respectively. Due to the lesser representation of the Neutral majority model and the introduction of synthetic or oversampled minority data, the decline in model performance implies that balancing the dataset affected model generalisation.

Before balancing, the residual plots for the three models highlight varied performance trends, with Gradient Boosting emerging as the most effective. Random Forest demonstrates some heteroscedasticity, as residuals become more dispersed with higher fitted values, indicating difficulties in predicting larger values accurately. Despite this, residuals cluster near the zero line for smaller fitted values, suggesting reasonable performance for lower predictions. Decision Tree, however, shows a greater spread of residuals and inconsistent patterns around the zero line, revealing limited reliability in its predictions (Figure 26). The imbalanced dataset, dominated by the Neutral sentiment, likely exacerbates these inconsistencies, as the model has fewer examples from minority classes to learn from. Gradient Boosting, in contrast, has residuals more tightly concentrated around the zero line, with fewer extreme outliers. While it still shows slight heteroscedasticity, it handles the imbalanced dataset more effectively, making it the most stable model prior to balancing.

After balancing, the residual plots reveal a clear decline in the performance of all three models. Random Forest exhibits a significant increase in residual dispersion, with values deviating further from the zero line, indicating that the model struggles to adapt to the new balanced distribution. The Decision Tree model shows even broader residual variance and multiple outliers, further highlighting its reduced predictive accuracy. This decline can be attributed to the loss of the majority class’s dominance, which previously allowed the model to perform well in specific areas.

Gradient Boosting, while still outperforming the other models, shows a slight drop in performance, with residuals remaining closer to the zero line but showing increased outliers compared to its pre-balancing performance (Figure 27). These results demonstrate that balancing improved class fairness but introduced challenges for the models, requiring additional optimization strategies such as hyperparameter tuning or advanced balancing techniques like SMOTE. The trade-offs between fairness and predictive accuracy underscore the complexity of balancing dataset biases while maintaining robust model performance.

**Before Data Balancing:**

A graph with blue dots and a red line

Description automatically generatedA graph with blue dots and a red line

Description automatically generatedA graph with blue dots and red line

Description automatically generated

**(Random Forest) (Decision Tree) (Gradient Boosting)**

**After Data Balancing:**

A graph with blue dots

Description automatically generated A graph with blue dots and a red line

Description automatically generatedA graph with blue dots and red line

Description automatically generated

**(Random Forest) (Decision Tree) (Gradient Boosting)**

**Figure 27 Residual Vs Fitted Value Plots Before and After data balancing (Random Forest, Decision Tree and Gradient Boosting) (Self-Created)**

A screen shot of a computer

Description automatically generated

**Figure 28 LSTM model Performance before class balancing (Self-Created)**

LSTM model demonstrated a continuous decline in training and validation loss over time before balancing, indicating good learning. Training loss remained high at 4709415911424.0 and validation loss at 303831678976.0. MAE was significant at 142635.8 for the training set and 122857.8 for the validation set (Figure 28). High values indicate that the model has problems predicting, maybe due to the dataset's class imbalance. Neutral model performance hindered minority class learning and prediction. Training reduced loss and MAE, but imbalanced data impaired model accuracy and reliability.

A screenshot of a computer screen

Description automatically generated

**Figure 29 LSTM model Performance After Class Balancing (Self-Created)**

After balancing, LSTM model performance improved with lower loss and MAE. The model fitted the balanced dataset better, reducing training loss to 108562898944.0 and validation loss to 111778398208.0. The MAE dropped to 66609.4 for training and 64045.5 for validation (Figure 29), improving prediction accuracy across all classes. We found that balancing improved the model's generalisation by better representing each class. Eliminating data imbalances reduces training and validation errors, improving deep learning model performance. However, the high loss and MAE values suggest hyperparameter modification or advanced architectures may enhance results.

**A white background with black and white text

Description automatically generated**

**Figure 30 CNN Model Performance before class balancing (Self-Created)**

The CNN model's loss and MAE decline before balancing (figure.30), showing constant performance improvement. The validation loss begins at 296206925824.0 with an MAE of 120598.6 and the training loss at 3259579498496.0 with 165365.7. Epochs lower training loss to 334199816192.0 and validation loss to 164870029312.0, with MAE values of 92322.4 and 77099.6. Due to the imbalanced dataset, the model learns well from training data but may overfit to the majority class. The model's high validation MAE suggests it cannot predict minority class outcomes, a typical imbalanced dataset problem.

A white background with black and white text

Description automatically generated

**Figure 31 CNN Model Performance After Class Balancing (Self-Created)**

After balancing, CNN model performance significantly improves, as indicated by lower loss and MAE values (figure.31). Training loss starts at 365225803776.0 with an MAE of 111398.2 and drops to 115418988544.0, whereas validation loss starts at 130169618432.0 and decreases to 109159628800.0. The validation MAE dropped from 69842.9 to 60591.9 in the final epoch. These findings demonstrate that balancing improves model generalisation and prediction by representing all classes equally. For further optimisation, validation MAE scores recommend changing hyperparameters or adding data. Overall, balancing has improved the model's ability to learn from and predict across all datasets, demonstrating the necessity to address class imbalance for better model performance.

## Reason For Using Positive/Negative Sentiment Instead Of Raw Polarity Scores

**1. Simplified Interpretation:** The categorization into positive, neutral, and negative sentiment simplifies the analysis and makes the results easier to interpret for non-technical stakeholders. This is especially important when presenting actionable insights to developers and publishers.

**2. Noise Reduction:** Raw polarity scores can sometimes be noisy, especially when the differences are minor and fall within an acceptable range of neutrality. Categorization helps mitigate such noise and ensures a cleaner representation of sentiment trends.

**3. Aligning with Industry Practices:** Many industry analyses use categorical sentiment as it provides a more practical and comprehensible overview. By using these categories, the results are more aligned with industry expectations and standards.

**4. Highlighting Patterns:** While raw polarity scores provide granularity, the broader sentiment categories allow for the identification of overarching patterns and trends that might be obscured by the nuances of raw scores.

**5. Computational Efficiency:** When building machine learning models, using sentiment categories can simplify the feature space, improving computational efficiency and potentially reducing the risk of overfitting.

**6. Enhanced Visualisation:** Visualizing data (e.g., in charts or graphs) is more intuitive and impactful with discrete categories like positive, neutral, and negative sentiment, as opposed to a continuous range of raw scores.

# Project Management

## Project Schedule

Below figure is the Gantt Chart consisting tasks and sub tasks with deadlines. This helps to complete the project within defined timelines –

A screenshot of a project

Description automatically generated

**Figure 32** **Gantt Chart (Self-Created)**

**Project Schedule and Execution**

the project followed a Gantt chart timetable with stages and milestones. Five phases are Initial Setup and Data Preparation, Model Development, Model Evaluation, Result Analysis and Reporting, and Final Review.

**Initial Setup and Data Preparation (26/09/24 – 07/10/24)**

This initial step required planning, data collection, investigation, and data cleansing and preparation. These processes prepared the dataset for modelling. The team attended academic writing workshops to prepare dissertation chapters.

**Model Development (08/10/24 – 22/10/24)**

LSTM, CNN, Gradient Boosting, and Decision Trees were used to create models. We extracted and chose key features, then optimised hyperparameters for model performance. Hyperparameter tuning took longer than planned, but weekly reviews and realigning overlapping tasks like the literature review and dissertation Chapter 2 helped. This repetitive, systematic approach advanced academics and technology.

**Model Evaluation (01/11/24 – 13/11/24)**

The models' prediction power was rigorously tested throughout this phase. Regression performance was assessed using cross-validation. These evaluations were reported in Chapters 4 and 5 of the dissertation. Regular supervisory meetings led and aligned projects despite the difficulty of interpreting evaluation criteria.

**Result Analysis and Reporting (14/11/24 – 29/11/24)**

Actionable insights and context were extracted from model evaluation outcomes. Write the dissertation's last chapters to document and visualise results. Time from previous buffer periods was used to correct small delays to keep this stage on schedule.

### Project Management and Adaptability

Task schedules and progress were carefully tracked in Gantt charts. Like Agile scrums, weekly reviews found bottlenecks and reorganised tasks. Write the literature review to avoid hyperparameter tuning delays during Model Development.

The hardest jobs were hyperparameter tweaking, which required numerous rounds to improve performance, and Chapters 4 and 5, which required thorough evaluation metric interpretation. These tasks were completed with weekly monitoring and participation.

The project timeline contained buffers for unexpected events. Minor adjustments prevented technical job delays from slowing progress. Academic writing workshops early in the project accelerated dissertation writing, especially later.

The Gantt chart helped manage time and prioritise projects. To complete the project and dissertation, technical and academic writing were combined. Regular reviews, supervisory feedback, and job flexibility helped overcome obstacles and meet project goals.

### Project Supervision and Feedback on Drafts

The feedback provided by my supervisor, **Mark Johnston**, has been incredibly valuable in shaping the direction and quality of this project. His detailed insights on each draft helped me refine the methodology and enhance the clarity of my research objectives. Mark’s suggestions were particularly useful in addressing key gaps in my analysis, ensuring the project maintained its focus and academic rigor. His guidance enabled me to better structure the project, making it more impactful and aligned with its intended goals.

## Risk Management

The most important risks identified were related to quality, that is, missing values or inconsistent data formats inside the dataset. Skewed results are due to poor data quality and can impact model accuracy and reliability. To overcome this, a preliminary data cleaning and preprocessing phase was designed by using robust methods of handling missing data, removing outliers, and standardizing data formats.

The second risk was driven by limitations of model performance. The goal of this project was to develop various algorithmic models like Random Forest, Decision Trees, Gradient Boosting, along with sophisticated features toward CNN and LSTM. This place it at risk of potential underperformance, due to overfitting, insufficiency of the data representation or selection of insufficient number of features. The mitigation approach consisted of the application of cross validation techniques; regularization techniques; and the thorough tuning of hyperparameters to increase model generalization and accuracy.

Another risk was that computational resource constraints would limit what can be analysed through the computers, particularly for the more highly personalised models such as deep learning techniques. This risk was mitigated by adopting efficient coding practices, making use of better libraries such as (Scikit-learn and TensorFlow), and cloud computing resources in accordance.

Then another risk was identified, timeline adherence due to the huge scope of the project and the necessity of having an overlap between many tasks. This was managed by a very detailed Gantt chart that divided each phase the project into easy to manage actionable tasks with defined start and end dates. That made it possible to follow the progress closely and adjust to achieve the desired schedule on the project.

### Risk Materialization and Response

Risks did materialize during the project at a manageable level. The biggest problem was related to data quality. The first dataset had a huge quantity of missing and inconsistent entries that required a lot more time and resource to be properly preprocessed. These methods of imputation and categorical encoding proved to work; but they worked, and the project carried on without major disruption.

In the model training phase, another challenge emerged largely due to sign of the overfitting of certain models particularly CNNs and LSTM during early iterations. All this risk was mitigated by using techniques such as dropout and data augmentation and adjusting the learning rate. This leads to improved performance on these measures and with expected results on unseen data.

## Quality Management

Data science project like Predicting Sales Performance of Steam Games Using Machine Learning is always a high-quality work. It fell within several of quality standards and used techniques to monitor project progress and to assess results.

### Standards Adopted

The project followed industry standards of the data science and machine learning practices including applying the CRISP-DM (Cross Industry Standard Process for Data Mining) principles. By following this framework, each phase of data understanding and preparation, modelling, evaluation, and deployment were systematic and efficient (Liu & Huang, 2017). Furthermore, coding practices also followed the PEP 8 style guide for python so as to produce readable and maintainable code. Best practices like data normalization, feature scaling, and cross validation were taken to enhance the model development and make sure that models are following the best data science standards.

### Progress Review Techniques

Keeping project quality was accomplished by regular progress reviews. A Gantt chart was used so that status meetings could be held weekly to see how progress was against the Gantt chart milestones, and where needed, identify potential delays or other issues as they were happening. Code reviews, as well as checks of the data pre-processing steps to make sure that they comply with the initial data plan were a part of these meetings.

In addition, iterative testing and validation were performed during the stage of model development. Model stability as well as performance over different subsets of the data were assessed by techniques such as K-fold cross validation. This approach served as an early metric for overfitting or potentially underperforming, and exercised model structure and hyperparameters as necessary.

### Methods for Outcomes Evaluations

The performance metrics were designed to help us evaluate the effectiveness of the project outcomes. Then regression metrics (MSE and R2) were used to assess the machine learning models. By thus unifying these metrics, we created a multi-dimensional view into how the model performed, because the predictions were both accurate and balanced among different classes.

A final evaluation of the model's generalization capabilities was achieved after post modelling on a separate test dataset. This step proved that the models didn’t simply generalize to the training data, but could generalize to test data as well. To ensure that feature importance analysis confirmed that the most important features were being used in the predictions, this confirmed that the models provided actionable insights that were aligned with the project’s objectives.

## Social, Legal, Ethical and Professional Considerations

### Professional Considerations

Data science and programming best practices were followed, and professional standards were stuck to, this project was. Responsibility in research and analysis was maintained by satisfying the office of the associations such as the ACM’s Code of Ethics and IEEE’s Code of Conduct. This included transparent documentation, maintaining data integrity and a commitment to unbiased results to draw on reliable decision making.

### Social Considerations

The project acknowledges the social responsibility in using the predictive models within gaming industry. Developers and publishers can use accurate sales prediction to help inform their strategies, but inaccurate prediction can also create risk by influencing the market competition and small developer visibility. An important consideration was how to balance such insights into fostering innovation without simultaneously disadvantaging smaller creators who worked so diligently and passionately to achieve industry growth.

### Legal Considerations

The General Data Protection Regulation (GDPR) and other data protection and privacy laws were very much considered. The dataset used did not consist of any personal user data, so to adhere to the relevant legal standards we needed to make sure any data usage met the standards. Agreements, and terms of usage, for using publicly available data sources were strictly abided by to maintain laws.

### Ethical Considerations

The main ethical concerns were related to fairness and transparency of the model. It was important to make sure that the machine learning algorithms were not letting the bias be propagated, especially when it comes to developers, publishers. Training data potential biases, as well as model results, were scrutinized to put in measures of biases, and the responsibility for presenting the findings. There were also ethical guidelines that covered not manipulating the results for commercial or competitive gain.

### Privacy Considerations

Though the project did not use user specific data, user privacy was still a top priority. For confidentiality of user information, any data related to user reviews or engagement metric was anonymized and used only for aggregation analysis. It had a supportive role in terms of privacy laws and underlined ethical data usage.

# Critical Appraisal

The project's outcomes are critically evaluated by appraising both the strengths and limitations of the methodologies and results.

## Positive Aspects

A machine learning and deep learning approach was applied successfully to predict a game’s sales performance with a variety of game features. One of the biggest positives was that multiple models Random Forest, Decision Tree, Gradient Boosting, LSTM and CNN were chosen and a comparison could be made as to how each one of them performs given the same dataset. In particular, the Random Forest model was largely more accurate in predicting than others in MSE and R² score. However, the sentiment analysis also helped enrich the dataset and enabled users to see meaningful user opinions, one of the key factors affecting sales results. By performing the sentiment classification, we were able to identify some key trends in the user feedback; most specifically, that nearly 84 percent of feedback was neutral, which can be useful information for a game developer to use in their development and marketing strategy.

Furthermore, examining various visualisations, for instance word clouds and sentiment distributions, not only provided additional value in offering intuitive, actionable insights into player satisfaction and dissatisfaction, but further directed attention to the areas in which particular design changes may result in improvement. While, these visual aids helped me understand the major factors that affected user engagement and sales potential, both of which could be immensely helpful during decision making for developers and publishers.

## Negative Aspects and Limitation

Although the results were positive, the project suffered from a number of limitations. A big problem was the vast amounts of missing data throughout different tables, for example, Reviews, Notes, and Metacritic URL. Although, there were rich sources of information in the dataset, the presence of a few missing values may set a toll on the accuracy & reliability of the models. However, since getting a full data set was problematic, a few imputation techniques were considered, but sometimes a lack of full data could bias the results, in particular with regards to predictive power.

A limitation was that the dataset was skewed in terms of price distribution as well as estimated owners. Training of these models was challenging due to long tail distribution – a small number of games constitute the majority of the market share, and the predictive models perhaps over leveraged on these high performing few games, and thus may have introduced a bias in predicting the games that are less popular.

Moreover, the Random Forest model did outperform the rest but the overall performance of the Gradient Boosting and the Decision Tree models are not in line with expectations. In particular, the Gradient Boosting model has high MSE and lower R² score, which implies that it cannot perform capturing patterns from the dataset as well as other models. This means that either the features that were selected were not good for this model or more tuning and preparation on the data may be needed to get the performance better.

The results from deep learning models LSTM and CNN also look promising, CNN being more accurate and consistent than LSTM. Both models, however, had relatively high mean absolute error (MAE) and validation loss, which shows that additional model refinement is required. These techniques are hyperparameter tuning, more complex network architectures, or extra training data to help make the models generally applicable.

## Knowledge Gained

This project led to gaining a thorough understanding of how machine learning models, especially Random Forest and deep learning models such as LSTM and CNN, could be implemented in the realm of real-world problems in gaming industry. Feature engineering was emphasized and the importance of exposing these engineered features to predictive modelling. The project also demonstrated the problems associated with working with real world data, including missing data, skewed distributions and model overfitting. Through this experience, we were re‑emphasized just how important it is to properly clean and pre-process data to make sure that models are trained on the best possible, most representative data.

In conclusion, the project allowed for learning regarding predictive factors of game sales performance, and indicating ways to strengthen the predictive power. But their work also illustrated what machine learning excels at capturing complex patterns in data as well as challenges making more accurate and robust predictions that still need to be solved.

# Conclusions

This project proved that machine learning and deep learning models can help predicting Steam games sales performance given different game attributes such as user sentiment, pricing, playtime and owner data. The project achieved this by analyzing the dataset with multiple models such as Random Forest, Decision Tree, Gradient Boosting, LSTM, and CNN and the resulting insights help determine what factors impact game success where the Random Forest model was found to offer the highest level of predictive accuracy. Furthermore, the sentiment analysis was also performed on user reviews showing player feedback effect on game performance. Results show that while the project met its objectives, improvements can be made such as by including additional data and the optimization of the model. The contributions include the understanding of game market dynamics and lay groundwork for future studies and applications in game analytics and prediction.

## Achievements

The main goal of this project which was predicting Steam game sales performance using machine learning techniques was achieved. We were able to infer the factors which help make a game successful by using various models such as Random Forest, Decision Tree, Gradient Boosting, LSTM, and CNN. The project successfully pre-processes and analysis data including important game features such as reviews, price, play metrics, and ownership data. The models showed different degrees of predictive accuracy using sentiment analysis and feature engineering with Random Forest being the most effective model using Mean Squared Error and R² scores. Besides that, the deep learning models, specifically the CNN, had also been promising in capturing complex patterns and perform more generalization than other models. With this work, we help to understand the factors that affect game sales, and give the base to make future improvements and uses for gaming analytics.

## Future Work

Several areas for future work surface in this project. It will be suggested that improving data quality, specifically reducing the effects of missing values and more granular game metadata (including developer reputation & advertising spend), would improve prediction performance. Then there is further tuning of hyperparameters for both machine learning and deep learning models, and further exploration of additional algorithms such as XGBoost or ensemble methods might improve accuracy. Furthermore, the dataset can be expanded to include additional games, fresh data can be added from recent time, and external factors such as market trends and game updates can be investigated to perform a more complete study. This future work could also include real time predicting the price adjustment based on user’s price of acceptance or marketing strategy which is built on the basis of sentiment analysis to optimize the promotions and sales for the game.

# Student Reflections

This project was definitely a very rewarding learning experience to think about and a great chance to apply theoretical concepts in practice. I was mostly responsible for conducting data analysis, implementing machine learning models on top of that data and evaluating their performance. In retrospect, I was able to identify numerous areas where I performed well personally, but also other spots that were challenging, and areas for further development.

## Achievements and Strengths

They key achievement for me was developing and fine-tuning machine learning models efficiently. I began by gaining an understanding of what the dataset looked like, and its structure, allowing me to form a plan on what pre-processing steps were necessary, e.g., handling missing data, encoding categorical variables. Assessing machine learning algorithm performance on different algorithms such as Random Forest and Gradient Boosting allowed me to try some real-world comparisons and select the best performing models. Also, this process helped me reinforce my technical knowledge of machine learning and refined my problem-solving skills as I made and tested different settings to enhance the accuracy of the model.

I felt I did well in the feature engineering domain specifically, such as sentiment analysis. Integrating textual data into the models gave more depth to the analysis, giving a more holistic view of the factors that contribute to games sales than could simply be quantified by numerical metrics. This was an area I was new, so it was a big learning milestone to successfully implement sentiment analysis.

## Challenges Encountered

This project also posed a few challenges. The biggest of these was the missing data in the dataset. There were quite a few columns missing values and this feature, if not handled properly could have had an adverse effect on the quality of the models. I first tried to just drop rows that had missing data, but this made the dataset size very small, which could harm model performance. I conducted some research and experimented with imputation techniques, and finally decided to use median imputation for numerical features as it improved the integrity of the dataset without losing too much data. However, I think I could have looked at other imputation methods, like predictive modelling or multiple imputation, for missing values.

Second thing I struggled with the unbalanced distribution of the size of sales and reviews on the dataset. There was a few of those high performing games that skewed the dataset high in sales overall and with low sales. The accuracy of the predictive models was skewed by this. At first, I dismissed this as an issue but then went back to analyse the model performance, whereby I saw that something like resampling or class imbalance correction would lead to better results. Looking back, perhaps some successes might be obtained were these techniques applied earlier in the data preparation phase.

## Lessons Learned and Areas for Improvement

I learned lessons along the way in data pre-processing, model selection and evaluation on this project. I also learnt that data cleaning and feature engineering are not just pre requisite steps to the modelling process. Input data quality and relevance will greatly affect the success of machine learning models. Further, I understood the value of model interpretability and recognizing the assumptions being made with these algorithms. Random Forest and Gradient Boosting models were good, but I should have tuned their hyperparameters more, which should have improved the performance more.

Secondly, I also learned that machine learning projects always need to be iterated on. Initially my models were not perfect, but with testing and refining of the models, I was able to improve the models. We can observe from this that the importance of patience and persistence in model optimization was evident, as was the need to tweak the models by making refactors based on real time feedback from the evaluation metrics.

These models were promising, though I could have spent more time tuning their parameters and trying out other architecture to improve performance. Given that, further experimenting with transfer learning and ensemble methods on available computational resources could have produced better outcomes.

# Bibliography and References

1. Andrews, M., Luo, X., Fang, Z., & Aspara, J. (2014). Cause Marketing Effectiveness and the Moderating Role of Price Discounts. Journal of Marketing, 78(6), 120–142. <https://doi.org/10.1509/jm.14.0003>
2. Abonazel, M.R. (2020). Handling Outliers and Missing Data in Regression Models Using R: Simulation Examples. Academic Journal of Applied Mathematical Sciences, (68), pp.187–203. doi: <https://doi.org/10.32861/ajams.68.187.203>.
3. Brunt, C. S., King, A. S., & King, J. T. (2019). The influence of user-generated content on video game demand. Journal of Cultural Economics. <https://doi.org/10.1007/s10824-019-09349-0>
4. Clement, J. (2024, January 10). Steam gaming platform - Statistics & Facts. Www.statista.com; Statista. <https://www.statista.com/topics/4282/steam/>
5. Gangwar, A., Singh, S., Mishra, R. and Prakash, S. (2023). The State-of-the-Art in Air Pollution Monitoring and Forecasting Systems Using IoT, Big Data, and Machine Learning. Wireless Personal Communications, 130(3), pp.1699–1729. doi: <https://doi.org/10.1007/s11277-023-10351-1>.
6. Florez, M. A., & Zuluaga, C. C. (2023). Interpreting direct sales’ demand forecasts using SHAP values. Production, 33. <https://doi.org/10.1590/0103-6513.20220035>
7. Flunger, R., Mladenow, A., & Strauss, C. (2021). Game Analytics—Business Impact, Methods and Tools. Studies in Systems, Decision and Control, 601–617. <https://doi.org/10.1007/978-3-030-77916-0_19>
8. Finch, G., & Buchmesse, F. (2019). The Future Impact of Artificial Intelligence on the Publishing Industry. Livre blanc.
9. Hamarashid, H. K., Saeed, S. A., & Rashid, T. A. (2022). A comprehensive review and evaluation on text predictive and entertainment systems. Soft Computing. <https://doi.org/10.1007/s00500-021-06691-4>
10. Hukal, P., Kanat, I., & Ozalp, H. (2022). Different Strategy Playbooks for Digital Platform Complementors. MIS Quarterly Executive, 1–15. <https://doi.org/10.17705/2msqe.00056>
11. Josef, A., Van Lepp, A., & Carper, M. D. (2022). The Business of Indie Games: Everything You Need to Know to Conquer the Indie Games Industry. CRC Press.
12. Joshi, A., Saggar, P., Jain, R., Sharma, M., Gupta, D. and Khanna, A. (2021). CatBoost- An Ensemble Machine Learning Model for Prediction and Classification of Student Academic Performance. Advances in Data Science and Adaptive Analysis. doi: <https://doi.org/10.1142/s2424922x21410023>.
13. Lee, J., Jung, O., Lee, Y., Kim, O., & Park, C. (2021). A Comparison and Interpretation of Machine Learning Algorithm for the Prediction of Online Purchase Conversion. Journal of Theoretical and Applied Electronic Commerce Research, 16(5), 1472–1491. <https://doi.org/10.3390/jtaer16050083>
14. Liu, M. C., & Huang, Y. M. (2017). The use of data science for education: The case of social-emotional learning. Smart Learning Environments, 4, 1-13.
15. Lin, D., Bezemer, C.-P., Zou, Y., & Hassan, A. E. (2018). An empirical study of game reviews on the Steam platform. Empirical Software Engineering, 24(1), 170–207. <https://doi.org/10.1007/s10664-018-9627-4>
16. McKaughan, J. G., Zoch, L. M., Bowers, K. W., & Smith, M. J. (2017). Video Game Values: A qualitative study of organizational values in the video game industry. Radford University.
17. Muhammed, Liong, C.-Y., & Gunasekaran, H. (2023). Visualisation and prediction of user ratings on video games using web analytics. AIP Conference Proceedings, 2500, 020010–020010. <https://doi.org/10.1063/5.0128976>
18. Ozalp, H. (2024). Heterogeneous Development Paths to Growth and Innovation : The Evolution of the Video Game Industry across Four Hubs. Social Science Research Network. <https://doi.org/10.2139/ssrn.4816101>
19. Panwar, A., & Bhatnagar, V. (2020). Sentiment Analysis of Game Review Using Machine Learning in a Hadoop Ecosystem. Advances in Computer and Electrical Engineering Book Series, 145–165. <https://doi.org/10.4018/978-1-7998-2772-6.ch008>
20. Parlan, B. (2017). INDEPENDENT GAME DEVELOPMENT: PRICE OF FREEDOM. International Journal of Social Sciences and Interdisciplinary Studies, 2(2), 57-69.
21. Politowski, C., Petrillo, F., Ullmann, G. C., & Guéhéneuc, Y.-G. (2021). Game industry problems: An extensive analysis of the gray literature. Information and Software Technology, 134, 106538. <https://doi.org/10.1016/j.infsof.2021.106538>
22. PraveenaSri, P., Prasuna, V. N. P., Murugesan, R., & Usha, S. P. (2023, May). Business Challenges of Forecasting Sales in Bakery Industry: Applications of Machine Learning Algorithms. In Proceedings of the International Conference on Emerging Trends in Business & Management (ICETBM 2023) (Vol. 242, p. 335). Springer Nature.
23. Rahman, S. A., Ali, Bakar, A. A., & Mutalib, S. (2024). Enhancing churn forecasting with sentiment analysis of steam reviews. Social Network Analysis and Mining, 14(1). <https://doi.org/10.1007/s13278-024-01337-3>
24. Raman, D.R., Jayalakshmi, S., Arumugam, K., Raj, A.V.A., Balaji, D. and Brightsingh, R. (2022). Implementation of Data Analysis and Document Summarization in Social Media Data Using R and Python. 2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA). doi: <https://doi.org/10.1109/icirca54612.2022.9985479>.
25. Reardon, D. C., & Wright, D. (2021). The Digital Role-Playing Game and Technical Communication: A History of Bethesda, BioWare, and CD Projekt Red. Bloomsbury Publishing USA.
26. Rizani, M. N., Khalid, M. N. A., & Iida, H. (2023). Application of meta-gaming concept to the publishing platform: Analysis of the steam games platform. Information, 14(2), 110.
27. Santos, N. (2022). Assassins and the Creed: A look at the Assassin’s Creed series, Ubisoft, and women in the video games industry. Women in Historical and Archaeological Video Games, 9, 25.
28. Sattar, R. A., Wang, S., Muqadas, M., Ashraf, M. F., & Tahir, M. N. (2017). Qualitative and quantitative approaches to study adoption of sustainable agricultural practices: a research-note on mixed method approach. International Journal of Agricultural Extension and Rural Development, 5(2), 539-544.
29. Shih, J. Y., & Wang, J. H. (2024, April). Application of Latent Dirichlet Allocation Algorithm on User Review Analysis: A Case Study on Online Games. In 2024 IEEE 4th International Conference on Electronic Communications, Internet of Things and Big Data (ICEIB) (pp. 251-256). IEEE.
30. Syam, N., & Sharma, A. (2018). Waiting for a sales renaissance in the fourth industrial revolution: Machine learning and artificial intelligence in sales research and practice. Industrial Marketing Management, 69(0019-8501), 135–146. <https://doi.org/10.1016/j.indmarman.2017.12.019>
31. Teja, A. S., Muhammad, & Qomariyah, N. N. (2023). Predicting Steam Games Rating with Regression. E3S Web of Conferences, 388, 02001–02001. <https://doi.org/10.1051/e3sconf/202338802001>
32. Tobon, S., & Abril, C. (2024). Game on: curbing impulse buying and returns in apparel e-tailers. Review of Managerial Science. <https://doi.org/10.1007/s11846-024-00808-3>
33. Tong, X. (2021). Positioning game review as a crucial element of game user feedback in the ongoing development of independent video games. Computers in Human Behavior Reports, 3, 100077. <https://doi.org/10.1016/j.chbr.2021.100077>
34. Visconti, R. M. (2021). Video Games and Digital Media: Business Models and Valuation Approaches. SSRN Electronic Journal. <https://doi.org/10.2139/ssrn.4132429>
35. Varghese, R. R., Aiswarya, R., Roy, A., Muraly, V., & Renjith, S. (2022). A Novel Approach to Predict Success of Online Games Using Random Forest Regressor for Time Series Data. In International Conference on Advances in Electrical and Computer Technologies, 27–40.
36. Wang, D., Moh, M., & Moh, T.-S. (2020). Using Deep Learning and Steam User Data for Better Video Game Recommendations. Proceedings of the 2020 ACM Southeast Conference. <https://doi.org/10.1145/3374135.3385283>
37. Williams, R. (2021). Kids can gamble too: Loot boxes and predatory video game practices. Ariz. St. U. Sports & Ent. LJ, 11, 93.
38. Ao, Y., Li, H., Zhu, L., Ali, S., & Yang, Z. (2018). The linear random forest algorithm and its advantages in machine learning assisted logging regression modeling. *Journal of Petroleum Science and Engineering*, *174*, 776–789. <https://doi.org/10.1016/j.petrol.2018.11.067>
39. Guitart, A., Chen, P. P., Bertens, P., & Periáñez, Á. (2018). Forecasting Player Behavioral Data and Simulating In-Game Events. In *Advances in intelligent systems and computing* (pp. 274–293). <https://doi.org/10.1007/978-3-030-03402-3_19>
40. Suthaharan, S. (2015). Decision Tree Learning. In *Integrated series on information systems/Integrated series in information systems* (pp. 237–269). <https://doi.org/10.1007/978-1-4899-7641-3_10>
41. Zhang, Y., & Haghani, A. (2015). A gradient boosting method to improve travel time prediction. *Transportation Research Part C Emerging Technologies*, *58*, 308–324. <https://doi.org/10.1016/j.trc.2015.02.019>
42. Zhou, X., Li, Y., & Liang, W. (2020). CNN-RNN Based Intelligent Recommendation for Online Medical Pre-Diagnosis Support. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *18*(3), 912–921. <https://doi.org/10.1109/tcbb.2020.2994780>

# Appendix A – Project Source Code

import nltk

from nltk.corpus import stopwords

from nltk.tokenize import word\_tokenize

from nltk.stem import WordNetLemmatizer

import string

from nltk.sentiment import SentimentIntensityAnalyzer

from wordcloud import WordCloud

import matplotlib.pyplot as plt

import pandas as pd

from collections import Counter

import numpy as np

from sklearn.model\_selection import train\_test\_split, GridSearchCV, cross\_val\_score

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.ensemble import RandomForestRegressor, GradientBoostingRegressor

from sklearn.tree import DecisionTreeRegressor

from sklearn.metrics import mean\_squared\_error, r2\_score

from sklearn.pipeline import Pipeline

from sklearn.compose import ColumnTransformer

from sklearn.impute import SimpleImputer

import matplotlib.pyplot as plt

import seaborn as sns

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, LSTM, Dropout

from tensorflow.keras.optimizers import Adam

from sklearn.preprocessing import StandardScaler

from sklearn.model\_selection import train\_test\_split

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Conv1D, Flatten

# Download necessary NLTK resources

nltk.download('punkt')

nltk.download('stopwords')

nltk.download('wordnet')

nltk.download('vader\_lexicon')

nltk.download('punkt\_tab')

# Load the dataset

data = pd.read\_csv('/content/drive/My Drive/Data Science Project/games.csv')

data.head()

# Initialize the stopwords, lemmatizer, and sentiment analyzer

stop\_words = set(stopwords.words('english'))

lemmatizer = WordNetLemmatizer()

sia = SentimentIntensityAnalyzer()

# Function to preprocess the reviews

def preprocess\_review(review):

# Tokenization

tokens = word\_tokenize(review.lower())

# Remove punctuation and stop words, and apply lemmatization

tokens = [lemmatizer.lemmatize(word) for word in tokens if word not in stop\_words and word not in string.punctuation]

return ' '.join(tokens)

# Apply the preprocessing function to the 'Reviews' column

data['Cleaned\_Reviews'] = data['Reviews'].apply(lambda x: preprocess\_review(str(x)))

data.head()

# Sentiment analysis function using VADER sentiment intensity analyzer

def get\_sentiment\_score(review):

return sia.polarity\_scores(review)['compound']

# Apply sentiment analysis to the cleaned reviews

data['Sentiment\_Score'] = data['Cleaned\_Reviews'].apply(lambda x: get\_sentiment\_score(x))

data.head()

# Categorize reviews as positive, negative, or neutral based on the sentiment score

def sentiment\_category(score):

if score > 0.05:

return 'Positive'

elif score < -0.05:

return 'Negative'

else:

return 'Neutral'

data['Sentiment'] = data['Sentiment\_Score'].apply(lambda x: sentiment\_category(x))

data.head()

# Display the sentiment distribution

print(data['Sentiment'].value\_counts())

data.columns

# Drop rows with null values in the 'Reviews' column

data.dropna(subset=['Reviews'], inplace=True)

# Generate word clouds for positive, negative, and neutral reviews

def generate\_wordcloud(sentiment\_category):

# Filter the cleaned reviews by sentiment category

reviews = ' '.join(data[data['Sentiment'] == sentiment\_category]['Cleaned\_Reviews'])

# Generate a word cloud

wordcloud = WordCloud(width=800, height=400, background\_color='white').generate(reviews)

# Plot the word cloud

plt.figure(figsize=(10, 5))

plt.imshow(wordcloud, interpolation='bilinear')

plt.axis('off')

plt.title(f'Word Cloud for {sentiment\_category} Reviews')

plt.show()

# Generate word clouds for positive, negative, and neutral reviews

generate\_wordcloud('Positive')

generate\_wordcloud('Negative')

generate\_wordcloud('Neutral')

# Further analysis: most frequent words in positive and negative reviews

def most\_frequent\_words(sentiment\_category):

reviews = ' '.join(data[data['Sentiment'] == sentiment\_category]['Cleaned\_Reviews'])

words = reviews.split()

word\_freq = Counter(words)

return word\_freq.most\_common(10)

# Most common words in positive reviews

print("Most frequent words in positive reviews:", most\_frequent\_words('Positive'))

# Most common words in negative reviews

print("Most frequent words in negative reviews:", most\_frequent\_words('Negative'))

# 1. Distribution of Game Prices

plt.figure(figsize=(10, 6))

sns.histplot(data['Price'], bins=30, kde=True, color='blue')

plt.title('Distribution of Game Prices')

plt.xlabel('Price ($)')

plt.ylabel('Count')

plt.grid(True)

plt.show()

# 2. Sentiment Analysis of Reviews

plt.figure(figsize=(8, 6))

data['Sentiment'].value\_counts().plot(kind='bar', color=['green', 'red', 'gray'])

plt.title('Sentiment Distribution of Reviews')

plt.xlabel('Sentiment')

plt.ylabel('Number of Reviews')

plt.grid(True)

plt.show()

# 3. Distribution of Estimated Owners

data['Estimated owners'] = data['Estimated owners'].apply(lambda x: int(x.split('-')[0]) if '-' in str(x) else int(x))

plt.figure(figsize=(10, 6))

sns.histplot(data['Estimated owners'], bins=30, kde=True, color='purple')

plt.title('Distribution of Estimated Owners')

plt.xlabel('Estimated Owners')

plt.ylabel('Frequency')

plt.show()

# 4. Popular Genres

plt.figure(figsize=(10, 6))

genres = data['Genres'].str.split(',').explode()

genres.value\_counts().head(10).plot(kind='bar', color='orange')

plt.title('Top 10 Most Popular Game Genres')

plt.xlabel('Genre')

plt.ylabel('Count')

plt.xticks(rotation=45)

plt.show()

# 5. Correlation Heatmap

numerical\_features = ['Price', 'DLC count', 'Positive', 'Negative', 'Achievements', 'Recommendations']

plt.figure(figsize=(10, 8))

sns.heatmap(data[numerical\_features].corr(), annot=True, cmap='coolwarm', fmt='.2f')

plt.title('Correlation Heatmap of Numerical Features')

plt.show()

# 6. Average Playtime Distribution

plt.figure(figsize=(10, 6))

sns.histplot(data['Average playtime forever'], bins=30, kde=True, color='teal')

plt.title('Distribution of Average Playtime Forever')

plt.xlabel('Average Playtime (hours)')

plt.ylabel('Frequency')

plt.grid(True)

plt.show()

data.isna().sum()

data.info()

# Data Preprocessing

# 1. Handle non-numeric values in 'Estimated owners' by converting to numeric (removing ranges)

data['Estimated owners'] = data['Estimated owners'].apply(lambda x: np.mean([int(i) for i in str(x).split('-')]) if '-' in str(x) else int(x))

# 2. Fill missing values in categorical columns with 'Unknown' or appropriate default

data.fillna({'Name': 'Unknown', 'About the game': 'Unknown', 'Developers': 'Unknown',

'Publishers': 'Unknown', 'Categories': 'Unknown', 'Genres': 'Unknown',

'Tags': 'Unknown'}, inplace=True)

# 3. Drop columns with significant missing data or irrelevant to analysis

data.drop(columns=['Reviews', 'Website', 'Support url', 'Support email', 'Score rank', 'Notes', 'Metacritic url'], inplace=True)

data.columns

# 2. Encoding categorical variables

# Identify object data type columns

object\_cols = data.select\_dtypes(include=['object']).columns

# Initialize LabelEncoder

label\_encoder = LabelEncoder()

# Iterate through object columns and apply Label Encoding

for col in object\_cols:

# Handle potential errors during label encoding

try:

data[col] = label\_encoder.fit\_transform(data[col])

except TypeError as e:

print(f"Error encoding column '{col}': {e}")

# Optionally, handle the error more specifically, e.g., by filling missing values

# or dropping the problematic column

data[col] = data[col].astype(str) # Try converting to string first

data[col] = label\_encoder.fit\_transform(data[col])

data.head()

data.info()

"""# Original Data"""

# Define target and features

X = data[['Developers', 'Publishers', 'Categories', 'Genres', 'Price', 'DLC count', 'Positive', 'Negative']]

y = data['Estimated owners']

# Split the dataset into training and testing

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Feature Scaling

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

# Machine Learning Model Development

models = {

'Random Forest': RandomForestRegressor(),

'Decision Tree': DecisionTreeRegressor(),

'Gradient Boosting': GradientBoostingRegressor()

}

# Train and evaluate each model

results = {}

for model\_name, model in models.items():

model.fit(X\_train\_scaled, y\_train)

y\_pred = model.predict(X\_test\_scaled)

# Evaluation metrics

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

results[model\_name] = {'MSE': mse, 'R2 Score': r2}

residuals = y\_test - y\_pred

plt.figure(figsize=(8, 6))

plt.scatter(y\_pred, residuals)

plt.xlabel("Fitted Values")

plt.ylabel("Residuals")

plt.title("Residuals vs. Fitted Values")

plt.axhline(y=0, color='r', linestyle='--') # Add a horizontal line at y=0

plt.show()

# Display results

results\_df = pd.DataFrame(results).T

print("Model Performance:\n", results\_df)

# LSTM Model Development

# Select features and target

features = ['Positive', 'Negative', 'Price', 'Achievements', 'DLC count']

target = 'Estimated owners'

X = data[features].values

y = data[target].values

# Split the data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Scale the features

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

# Reshape data for LSTM (3D input: [samples, time steps, features])

X\_train\_scaled = np.expand\_dims(X\_train\_scaled, axis=1)

X\_test\_scaled = np.expand\_dims(X\_test\_scaled, axis=1)

# LSTM Model

lstm\_model = Sequential([

LSTM(64, activation='relu', input\_shape=(X\_train\_scaled.shape[1], X\_train\_scaled.shape[2])),

Dropout(0.2),

Dense(32, activation='relu'),

Dropout(0.2),

Dense(1) # Output layer

])

lstm\_model.compile(optimizer=Adam(learning\_rate=0.001), loss='mse', metrics=['mae'])

# Train the model

lstm\_model.fit(X\_train\_scaled, y\_train, epochs=50, batch\_size=32, validation\_split=0.2)

# Evaluate the model

loss, mae = lstm\_model.evaluate(X\_test\_scaled, y\_test)

print(f"LSTM Model - Loss: {loss}, MAE: {mae}")

# CNN Model Development

# CNN requires 3D input: [samples, time steps, features]

# Reshape data for CNN

X\_train\_cnn = np.expand\_dims(X\_train, axis=1)

X\_test\_cnn = np.expand\_dims(X\_test, axis=1)

# CNN Model

cnn\_model = Sequential([

Conv1D(filters=64, kernel\_size=1, activation='relu', input\_shape=(X\_train\_cnn.shape[1], X\_train\_cnn.shape[2])), # Adjusted kernel size to 1

Dropout(0.2),

Flatten(),

Dense(32, activation='relu'),

Dropout(0.2),

Dense(1) # Output layer

])

cnn\_model.compile(optimizer=Adam(learning\_rate=0.001), loss='mse', metrics=['mae'])

# Train the model

cnn\_model.fit(X\_train\_cnn, y\_train, epochs=50, batch\_size=32, validation\_split=0.2)

# Evaluate the model

loss, mae = cnn\_model.evaluate(X\_test\_cnn, y\_test)

print(f"CNN Model - Loss: {loss}, MAE: {mae}")

"""# Balanced Data"""

# Data Balancing

from imblearn.over\_sampling import SMOTE

# Separate features (X) and target variable (y)

X = data.drop(['Sentiment\_Score', 'Sentiment'], axis=1)

y = data['Sentiment']

# Initialize SMOTE

smote = SMOTE(random\_state=42)

# Apply SMOTE to balance the classes

X\_resampled, y\_resampled = smote.fit\_resample(X, y)

# Create a new balanced DataFrame

balanced\_data = pd.DataFrame(X\_resampled, columns=X.columns)

balanced\_data['Sentiment'] = y\_resampled

# Now balanced\_data contains the balanced dataset

print(balanced\_data['Sentiment'].value\_counts())

# Balanced Data

data=balanced\_data.copy()

data.columns

# Define target and features

X = data[['Developers', 'Publishers', 'Categories', 'Genres', 'Price', 'DLC count', 'Positive', 'Negative']]

y = data['Estimated owners']

# Split the dataset into training and testing

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Feature Scaling

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

# Machine Learning Model Development

models = {

'Random Forest': RandomForestRegressor(),

'Decision Tree': DecisionTreeRegressor(),

'Gradient Boosting': GradientBoostingRegressor()

}

# Train and evaluate each model

results = {}

for model\_name, model in models.items():

model.fit(X\_train\_scaled, y\_train)

y\_pred = model.predict(X\_test\_scaled)

# Evaluation metrics

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

results[model\_name] = {'MSE': mse, 'R2 Score': r2}

residuals = y\_test - y\_pred

plt.figure(figsize=(8, 6))

plt.scatter(y\_pred, residuals)

plt.xlabel("Fitted Values")

plt.ylabel("Residuals")

plt.title("Residuals vs. Fitted Values")

plt.axhline(y=0, color='r', linestyle='--') # Add a horizontal line at y=0

plt.show()

# Display results

results\_df = pd.DataFrame(results).T

print("Model Performance:\n", results\_df)

# LSTM Model Development

# Select features and target

features = ['Positive', 'Negative', 'Price', 'Achievements', 'DLC count']

target = 'Estimated owners'

X = data[features].values

y = data[target].values

# Split the data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Scale the features

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

# Reshape data for LSTM (3D input: [samples, time steps, features])

X\_train\_scaled = np.expand\_dims(X\_train\_scaled, axis=1)

X\_test\_scaled = np.expand\_dims(X\_test\_scaled, axis=1)

# LSTM Model

lstm\_model = Sequential([

LSTM(64, activation='relu', input\_shape=(X\_train\_scaled.shape[1], X\_train\_scaled.shape[2])),

Dropout(0.2),

Dense(32, activation='relu'),

Dropout(0.2),

Dense(1) # Output layer

])

lstm\_model.compile(optimizer=Adam(learning\_rate=0.001), loss='mse', metrics=['mae'])

# Train the model

lstm\_model.fit(X\_train\_scaled, y\_train, epochs=50, batch\_size=32, validation\_split=0.2)

# Evaluate the model

loss, mae = lstm\_model.evaluate(X\_test\_scaled, y\_test)

print(f"LSTM Model - Loss: {loss}, MAE: {mae}")

# CNN Model Development

# CNN requires 3D input: [samples, time steps, features]

# Reshape data for CNN

X\_train\_cnn = np.expand\_dims(X\_train, axis=1)

X\_test\_cnn = np.expand\_dims(X\_test, axis=1)

# CNN Model

cnn\_model = Sequential([

Conv1D(filters=64, kernel\_size=1, activation='relu', input\_shape=(X\_train\_cnn.shape[1], X\_train\_cnn.shape[2])), # Adjusted kernel size to 1

Dropout(0.2),

Flatten(),

Dense(32, activation='relu'),

Dropout(0.2),

Dense(1) # Output layer

])

cnn\_model.compile(optimizer=Adam(learning\_rate=0.001), loss='mse', metrics=['mae'])

# Train the model

cnn\_model.fit(X\_train\_cnn, y\_train, epochs=50, batch\_size=32, validation\_split=0.2)

# Evaluate the model

loss, mae = cnn\_model.evaluate(X\_test\_cnn, y\_test)

print(f"CNN Model - Loss: {loss}, MAE: {mae}")

**Appendix B – Certificate of Ethics Approval**

A certificate of ethical approval

Description automatically generated