

***Dissertation***

***KF7029***

***MSc Computer Science***

***and Digital Technologies Project***

Student Name: Elliot Anderson

Supervisor Name: Ian Watson

Second Marker Name: Naveed Anwar

Can recommendation-based machine learning models aid video game distribution platforms in user retention and the discoverability of games?2019/2020

Declaration

I declare the following:

(1) that the material contained in this dissertation is the end result of my own work and that due acknowledgement has been given in the bibliography and references to **ALL** sources be they printed, electronic or personal.

(2) the Word Count of this Dissertation is 14078

(3) that unless this dissertation has been confirmed as confidential, I agree to an entire electronic copy or sections of the dissertation to being placed on the eLearning Portal (Blackboard), if deemed appropriate, to allow future students the opportunity to see examples of past dissertations.  I understand that if displayed on eLearning Portal it would be made available for no longer than five years and that students would be able to print off copies or download.

(4) I agree to my dissertation being submitted to a plagiarism detection service, where it will be stored in a database and compared against work submitted from this or any other Department or from other institutions using the service.

In the event of the service detecting a high degree of similarity between content within the service this will be reported back to my supervisor and second marker, who may decide to undertake further investigation that may ultimately lead to disciplinary actions, should instances of plagiarism be detected.

(5) I have read the Northumbria University Policy Statement on Ethics in Research and Consultancy and I confirm that ethical issues have been considered, evaluated and appropriately addressed in this research.

**SIGNED:**

**DATE:**

Abstract

Increasingly machine learning is being used across many industries. This is due to machine learning’s ability to process vast amounts of data (such as user information like search history and content interactions) and then offer personalised data to the user based on these factors. One industry that could benefit from machine learning, is within the field of online video game distribution. The largest platform for this, Steam, does not currently use machine learning for its recommendation systems. This project aims to prove the effectiveness of this method within this industry by developing a recommendation model with machine learning and having participants evaluate its effectiveness in order to come to a conclusion whether it is better than current methods. The results of the tests proved that machine learning models do in fact aide video game distribution platforms in user retention and the discoverability of games. This was shown in the results, as an overwhelming majority of participants agreed that recommendations were accurate and felt the recommendations would be persuasive enough for them to buy the recommended games without doing further research for themselves. One thing presented that could be built upon after the conclusion of the project, was that a more thorough comparison of the older methods could be made, with more plans for users to test both recommendations side by side. Overall though, the project was a success.

Contents

[Declaration 2](#_Toc30110843)

[Abstract 3](#_Toc30110844)

[1 Introduction 6](#_Toc30110845)

[1.1 Background 6](#_Toc30110846)

[1.2 Aim 6](#_Toc30110847)

[1.3 Objectives 6](#_Toc30110848)

[1.4 Research Approach 8](#_Toc30110849)

[1.5 Legal, Social, Ethical and Professional considerations 8](#_Toc30110850)

[1.6 Structure of the Report 8](#_Toc30110851)

[2 Literature Review 9](#_Toc30110852)

[2.1 Machine learning 9](#_Toc30110853)

[2.2 Recommender Systems 9](#_Toc30110854)

[2.2.1 Current state of research and history 9](#_Toc30110855)

[2.2.2 How recommender systems work and choosing the best one 10](#_Toc30110856)

[2.2.3 Content-based recommendation models 10](#_Toc30110857)

[2.2.4 Collaborative filtering based recommendation models 11](#_Toc30110858)

[2.2.3 Evaluating the accuracy of recommendation systems 12](#_Toc30110859)

[2.2.4 Problems with recommendation systems 13](#_Toc30110860)

[2.3 Machine Learning Libraries 14](#_Toc30110861)

[2.4 How a games marketplace can benefit from a recommender system 14](#_Toc30110862)

[2.5 Summary 16](#_Toc30110863)

[3 Design and Implementation 17](#_Toc30110864)

[3.1 Legal, Social, Ethical and Professional considerations 17](#_Toc30110865)

[3.1.1 Ethics 17](#_Toc30110866)

[3.1.2 Legal 17](#_Toc30110867)

[3.1.3 Social 17](#_Toc30110868)

[3.1.4 Security 17](#_Toc30110869)

[3.1.5 Professional 17](#_Toc30110870)

[3.2 Model Development 18](#_Toc30110871)

[3.2.1 Tools and Software 18](#_Toc30110872)

[3.2.2 Chosen Model - Matrix Factorization Algorithm with Collaborative Filtering 18](#_Toc30110873)

[3.2.3 Requirements 18](#_Toc30110874)

[3.2.4 Design 19](#_Toc30110875)

[3.2.5 Implementation 22](#_Toc30110876)

[3.3 Testing the model and making changes 30](#_Toc30110877)

[3.3 Summary 33](#_Toc30110878)

[4 User Evaluation and Model Findings 34](#_Toc30110879)

[4.1 Did the model predict your favorite genre correctly? 34](#_Toc30110880)

[4.2 Would you play any of the games predicted for you? 35](#_Toc30110881)

[4.3 Overall would you agree the majority of the games are of interest to you? 36](#_Toc30110882)

[4.4 If these were to appear on the steam storefront would you consider checking one of these games out? 37](#_Toc30110883)

[4.5 Would you consider these recommendations any more accurate than steam’s current system? 38](#_Toc30110884)

[4.6 On games marketplaces do you usually listen to the recommendations? 39](#_Toc30110885)

[4.7 List the games that are of particular interest to you and explain why 40](#_Toc30110886)

[4.8 List the games that are not of interest to you and explain why 41](#_Toc30110887)

[4.9 Would you ever buy a game simply because it had been recommended to you and subsequently liked the look of it? 42](#_Toc30110888)

[4.10 Do you have any additional comments or suggestions for improvements on predictions? Or anything else you would like to elaborate on that may have been missed? 42](#_Toc30110889)

[4.11 Findings and Evaluation 43](#_Toc30110890)

[5 Project Evaluation 45](#_Toc30110891)

[5.1 Quality of Research 45](#_Toc30110892)

[5.2 Learning Journey 45](#_Toc30110893)

[5.3 Professional Approach 46](#_Toc30110894)

[5.4 Summary 46](#_Toc30110895)

[6 Conclusions and Recommendations 47](#_Toc30110896)

[6.1 Changes and Future Work 47](#_Toc30110897)

[Bibliography 48](#_Toc30110898)

[Appendix A. Research Proposal 52](#_Toc30110899)

[Appendix B. Ethics Approval 85](#_Toc30110900)

[Appendix C. Code 86](#_Toc30110901)

# Introduction

Increasingly machine learning is being used across many industries. This is due to machine learning’s ability to process vast amounts of data (such as user information like search history and content interactions) and then offer personalised data to the user based on these factors [1]. Companies such as Netflix utilise machine learning algorithms which have allowed them to save almost $1 billion, due to the ability to recommend personalised tv shows and movies [2].

One industry that could benefit from machine learning, is within the field of online video game distribution.

## Background

The largest platform for this, Steam, does not currently use machine learning for its recommendation systems. Though they do have plans for this in the future [3]. Research into utilising machine learning for this market would bring many benefits to the game industry, in the same way, it has helped entertainment platforms such as Netflix [2]. This project aims to prove the effectiveness of this method within this industry by developing a recommendation model with machine learning and having participants evaluate its effectiveness in order to come to a conclusion whether it is better than current methods.

## Aim

The exact aim of the project is:

* Can recommendation-based machine learning models aid video game distribution platforms in user retention and the discoverability of games?

## 1.3 Objectives

In order to achieve the aim “Can recommendation-based machine learning models aid video game distribution platforms in user retention and the discoverability of games?” the following objectives must be met.

**Objective 1:** Produce a literature review to compare the best recommendation-based machine learning models to find the most suitable to use with research

**Deliverable**: A background is needed first on the different recommendation models in order to begin development of a fitting model for use with a game’s marketplace. A literature review covering what machine learning is and its history, recommendation-based machine learning algorithms with a comparison of each one, how recommendation-based machine learning can be used in the industry and how this can keep user retention in a video game distribution platform and a conclusion which algorithm is best fit for the project.

**Quality Measure:**

* Number and quality of Sources used.
* Breadth and depth of the literature review and the amount of research undertaken for it.
* Has a solid conclusion on which the model will be developed.

**Objective 2:** Design the prototype software based on the chosen model and detail the research methodology such as how the tests will be carried out

**Deliverable:** A written chapter detailing the designs of the software. This will include a list of requirements that the design must adhere to based on the criteria set out by the literature review. Also, a design diagrams will be produced in the form of UML and finally a detailed plan on how the research and testing will be carried out with users.

**Quality Measure:**

* Workable designs that are fit for purpose.
* Detail covered of how research will be carried out.
* Diagrams can be followed and used easily for programming in the development stage.

**Objective 3:** Build a working prototype of the software with the functionality of being able to feed users profiles to get recommendations

**Deliverable:** The Final software in .exe format, it will also include all the code in C# and the files runnable from visual studio from source code. There will also be an explanation in written form about how each part of the code works.

**Quality Measure:**

* The software works and is ready to be tested with users.
* Software matches designs of the previous objective.
* Minimal bugs present in software.
* Code quality.
* Explanations of the code understandable to the average reader

**Objective 4:** Investigate the effectiveness of the ML model by using the software to perform tests on participants

**Deliverable:** Participant test results recorded presented as graphs and discussed, a discussion of the results of the test and an evaluation of what the participants thought overall, evaluating whether it was a success or failure based on if the participant answers were positive or negative.

**Quality Measure:**

* Questions are good at gauging the outcome of the test
* Results from users are extensive and good at evaluating the effectiveness of the software.
* Discussion of the results are conclusive and give a thorough overview of how it went.

**Objective 5:** Complete recommendations and conclusions based on the tests and an evaluation of the project overall

**Deliverable:** A written conclusion talking about if the objectives have been sufficiently met, an evaluation of the overall project, discussion about what went well and what could have been done differently.

**Quality Measure:**

• Justified recommendations and conclusions.

• Covers as much as the project as possible and talks about each objective.

• Explains in detail what went well and what went wrong throughout the project.

## 1.4 Research Approach

Qualitative research will be the main method used. This is due to the aim being very customer-centric as the measurement would be based on the opinions of users rather than a repeatable numerical result. The research will involve having participants complete surveys after sending them the result of what the recommendation model has predicted for them. Based on their answers an evaluation can be made on the success of the model and also, its ability to potentially keep user retention over other methods.

## 1.5 Legal, Social, Ethical and Professional considerations

The Northumbria University ethics guideline was adhered to throughout the project. In addition, before the project could even be started a peer reviewed ethical approval form was completed online to ensure that it was in line with it. This is discussed more in detail in chapter 3.1.

## 1.6 Structure of the Report

This dissertation is split into 6 chapters. The first chapter covers the introduction going over the basics of the project and the overall objective including how it will be achieved. After this, there is a chapter covering the literature review which will go over the background knowledge needed to complete this project including the technical and general background information needed to make a judgement on the effectiveness of the recommendation model.

The third chapter follows the implementation and design of the model including exactly how it was designed and subsequentially developed. Each line of code of the model will be explained when necessary to give an understanding of the reader how the model works.

After this, there is a chapter on user evaluation which involves analysing the answers given by users. This is the most important chapter as it provides the overall answer to the objective and aim of the project. Based on the answers of participants an evaluation can be made on the successfulness of the model.

Next, the whole project is evaluated with an analysis of the success of the dissertation as a whole and the learning journey throughout the project.

Finally, the last chapter covers the conclusion and future recommendations of what can be improved if this project were to be completed in the future and recommendations based on the findings of the project.

# 2 Literature Review

This literature review will cover what recommendation-based machine learning algorithms are including their history, current state of research, a comparison of the popular methods in order to choose the best one for use within this project, the challenges that may be faced plus how exactly they have benefited other companies and how this can be applied to a games marketplace.

## 2.1 Machine learning

Machine learning (a subset of artificial intelligence) is described as the study of algorithms and statistical models that computer systems can use to perform tasks without being given exact instructions, relying only on interference and patterns. Machine learning algorithms utilise a gathering of training data to make predictions without being programmed to perform the task [4].

Machine learning has many potential applications and can be tailored to each application to perform a specific task [4]. Its application can be used across many different fields including economics, marketing, linguistics and image recognition [5].

One of these applications, recommendation systems, has gained a lot of traction and has revolutionised the e-commerce market in many ways generating a lot of money for these markets. In the following sections, recommendation-based machine learning models will be looked at in-depth to see how they work and how it has helped other marketplaces.

## 2.2 Recommender Systems

Recommendation systems main goal is to generate recommendations for a group of users for products that may be of interest to them. For some real-world examples, this may include movies on Netflix or products to buy from eBay. The design of a recommendation system depends on the domain where the data falls into and the specific characteristics of the available data. An example of this is the fact Netflix datasets will include a rating scale from 1 to 5 based on whether a user likes or dislikes a movie. This kind of data source records the quality of interaction between the user and the item. [6]

Systems may also include additional attributes such as demographics, product descriptions and genres to provide an even more accurate recommendation. Recommender systems are effective due to their way of analysing data sources to develop affinity between the users and the products to identify pairs which are well matched. Two prominent systems are collaborative filtering systems and content-based filtering systems. Collab filtering analyses only historical interactions while content-based also use profile attributes. There also exist hybrid techniques which use techniques from both. Recommender systems and their evaluation of real-world problems are actively being researched [6].

### 2.2.1 Current state of research and history

Naturally, humans seek recommendations from trusted sources and are a natural process of making a decision. Due to the growth of online marketplaces, consumers are being shown an increasing amount of choices while the online sellers have the challenge of advertising their products to them. At the same time, online enterprises collect big quantities of transactional data. This can be used to analyse customers interactions with their site and products. Recommender systems have evolved and continue to evolve to fulfil the needs of buyers and sellers based on data analysis.

The first commercial recommender system named Tapestry introduced the term “Collaborative filtering.” [7] It was used to recommend documents to users from newsgroups. The motivation behind its creation was to prevent users from getting overwhelmed by massive amounts of documents. This method of analysing usage data across users to find the best matches in content is now often used in conjunction with an older method known as content filtering. This method was first created for information retrieval. With content filtering suggestions are not collaborative as they do not often use the entire user base to make a suggestion but focuses on a singular person and their attributes. An early example of collaborative filtering includes the GroupLens system [8]

Billsus and Pazzani [9] note that original formulas for recommender systems were basic correlation statistics and predictive modelling without utilising the wide range of practices in statistics and machine learning literature. As time went by in order to create a more accurate system efforts were made to combine content-based and collaborative based filtering together.

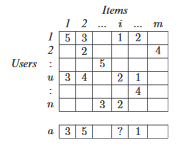
Further research was encouraged by the availability of public datasets freely available on the internet and by the growth of the e-commerce market. Companies such as Netflix released their own dataset to the public featuring 100 million ratings given by a half a million users and began a competition to find the best collaborative filtering algorithm in this domain.

Since then, Matrix Factorization [10] has appeared which uses complex techniques rooted in numerical linear algebra and statistical matrix analysis. Further research into this area is active, with applications being perused in a diverse range of domains. Including recommending music, books, movies and with this dissertation, video games.

### 2.2.2 How recommender systems work and choosing the best one

In its simplest form, most recommender systems function as shown in Figure 1 [6]. A matrix of n users and m items represent the known user preferences. Each cell represents the rating that the user has given to the item. The user rating matrix is usually sparse due to the fact most users do not rate every item. The task of the recommender is to give a prediction to a formerly unrated item. In most cases, ratings are predicted for every item that has not already been looked at by the user and the highest rated are presented to the user as recommendations.

Figure : User ratings matrix



There are many methods to create recommender systems but the approaches can be categorised these three ways: Content-Based Recommending – Where a user is recommended items that have similar attributes to items they already like or matched to predefined attributes of the user, Collaborative filtering – Where a user is recommended items based on all users collective past ratings and Hybrid: A combination of both. These methods will be looked at in detail and how they work exactly.

### 2.2.3 Content-based recommendation models

Content-based recommendation systems work in a much different way to pure collaborative filtering recommenders. As opposed to making predictions without regard to the specifics of users or their items, content-based tries to make recommendations based on the demographic information of users [11] or about an item, for example, the director and genre of a movie [12] An example of this if a user liked “Lord of the Rings” and “Harry Potter” the recommender may make a prediction for fantasy and could, therefore, recommend the film “The Chronicles of Narnia.” This approach provides recommendations by comparing representations of content describing an item to representations of content that interest the user.

Approaches to this method treat it as an information retrieval task wherein content which is associated with a user’s preference is treated as a query and the other items are scored with relevance and similarity to the query [13]

Within the domain of news filtering, NewsWeeder [14] had documents categorised by rating, converted into tf-idf word vectors then averaged to get a prototype vector of each category for a user. When a new document needed to be classified it would be compared with each of the prototype vectors and a rating was given to it based on the similarity to each category.

Another way this can be done without treating it as an information retrieval task is to treat it as a classification task. Using this method each example represents the content of an item and the labels represent the user’s past ratings. Within the domain of recommending books, Mooney and Roy [15] use text from the author, title, synopses, subject terms and reviews to train a naïve bayes classifier. The ratings are directly mapped to classes [12]

### 2.2.4 Collaborative filtering based recommendation models

User feedback is collected by collaborative filtering systems in the form of a rating for the item and then utilising the similar rating habits of users to determine the recommended item. Collaborative can be categorised further into other categories such as neighbourhood based and model-based approaches [16]

#### 2.2.4.1 Neighbourhood based collaborative filtering

In this technique, a group of users based on their similarity with the active user (user currently being recommended for) are chosen. A weighted combination of all their ratings is used to produce the recommendations for the user. In order to achieve this, every single user is given a similarity rating with respect to the active user. Then a group are selected that have the highest similarity, they are called the “neighbourhood.” These selected neighbours are used for the weighted combination [16]. However, neighbourhood-based CF is outdated and since then there have been many improvements made with newer methods.

#### 2.2.4.2 Item-based collaborative filtering

Neighbourhood based CF does not scale well when applied to millions of items and users due to the complexity and computational time it would take to find similar users. An alternative created by Linden [17] where instead of matching similar users, the user's rated items are matched with similar items. This leads to much faster systems and improved recommendations [18]

#### 2.2.4.3 Model-based Collaborative Filtering

These techniques work in much more complex ways but give a much faster and accurate recommendation than the rest of the CF techniques. They provide recommendations by estimating parameters of statistical models for user ratings. An early approach was described by Billsus and Pazzani [9] where a CF is mapped to a classification problem with a classifier built for the active user. The items are represented as features and the user’s ratings are represented as labels. These are used in addition to dimensionality reduction techniques to make data less sparse. Other models also exist such as latent factor and matrix factorisation models.

#### 2.2.4.4 Latent factor and matrix factorisation models

This is a recent state of the art method within this class of techniques [10]. Opposed to methods which recommend based on similar users, or items, these latent factor models assume that user similarity and item similarity is simultaneously induced by a hidden lower-dimensional structure within the data. An example of this is the rating given to a movie may be assumed to depend on a few factors which are not directly expressed such as their taste across multiple genres.

The million-dollar Netflix competition has thrown matrix factorisation to the top of recommendation system technologies in the setting of collaborative filtering [10]. Many different enhancements were found that lead to improvements to basic matrix factorisation.

This included using extra item and user-specific parameters to mitigate popular movies from always receiving higher ratings on average. Also having the ratings being time dependant as users’ opinions and tastes change over time.

Also, it is often the case that there is no like or dislike ratings available, only having a log of items a user may have bought without leaving any kind of review. Matrix factorisation techniques have evolved to handle these situations thus giving even more information to the recommender to make a better prediction.

#### 2.2.4.5 Hybrid Approaches

Content-based and collaborative based recommenders both have their strengths, so it is natural that both be combined to get the best of both. Several hybrid approaches have been proposed. A simple approach would be to allow both methods to create their own separate lists and then merge the results to produce a more accurate list [19]. Claypool, Gokhale, and Miranda [20] use adaptive weighted average to combine the two predictions, this increases the weight of the collaborative component as the number of users accessing the item is increased.

Melville et al. [12] have a method named “Content boosted collaborative filtering” where content-based predictions are used to convert sparse rating matrix into a full rating matrix and then collaborative filtering is used to give the recommendation. Specifically, a Naïve bayes classifier is trained on documents which describe item ratings of each user, and then replace the unrated items by the ones created by the classifier. This new predicted matrix is then used to find similar neighbours for the active user and produce further predictions using a method called Pearson correlation. This approach is shown to achieve more than just collab-filtering or just content-based systems.

#### 2.2.4.6 Chosen Approach for the project

Due to **matrix factorisation techniques** being cutting edge and effective as well as being very well researched it has been decided that this will be the technique used to create the game recommendation model for this project.

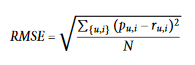
### 2.2.3 Evaluating the accuracy of recommendation systems

To gauge the quality of a recommendation system a test data set of known ratings is directly compared with the recommendation systems predicted results. Predictive accuracy metrics are used in most cases [21]. This encompasses many different kinds of metrics that can be used to measure accuracy. One very popular metric is the Mean Absolute Error (MAE) which is the average absolute difference between predicted ratings and actual ratings.



Pu,i is the predicted rating for the user (u) on item (i). ru,I is the real rating and N is the number of ratings within the test set.

Another popular and similar metric to MAE is the Root Mean Squared Error (RMSE) which gives emphasis to larger absolute errors.



These methods can be used to get an idea of how accurate it may be and so will be used with this project, however it cannot be used instead of testing on humans. It is possible that the accuracy tests will not yield the same results as asking a human participant of their opinion of the recommendation. Therefore, human testing will also be used.

### 2.2.4 Problems with recommendation systems

In order to be fully prepared, it is important to understand challenges and limitations with recommendation systems and how they can be addressed if any problem should arise.

#### 2.2.4.1 Sparsity

Sparsity can be a problem as most users will not have rated or watched every item, so due to this, the rating matrix can be very sparse. In particular, this can cause issues with collaborative filtering systems as the probability decreases in finding users with similar ratings. This problem commonly arises when the ratio of items to the user is very high. To best avoid this issue for this project a dataset should be found where the ratio of items to the user is not too high. If such a dataset cannot be found the issue can be mitigated by the use of additional domain information [22] or by finding a way of making assumptions about how the data is generated which can allow for high-quality imputation of the missing values [23], though this may not be possible in all cases.

#### 2.2.4.2 Cold Start Problem

The continuing addition of new users and items to a system can cause great challenges to recommendation systems. Together, these problems are called “the cold start problem” [24]. The first issue occurs in collaborative filtering systems since a new item cannot be recommended due to the fact no users have rated it. This can also be an issue with obscure items. Content-based recommendation [15] get around this problem as they do not rely on the ratings of users. Provided the item has attributes, recommendations can be produced from the start.

New users rather than new items is also a hard problem to tackle. Since having no previous ratings makes it impossible to find similar users for collaboration based or even for content-based. Research in this area has focused on ways to get the user to rate more items so that recommendation performance can improve [25].

These problems should not be an issue with this project as old datasets which have ratings for every item are being worked with and the aim is not to actively maintain a system where new items are being constantly added with new ratings as this is only a prototype.

#### 2.2.4.3 Fraud

With the increasing use of recommendation systems, the impact on the profitability of certain items can be felt by sellers, whether this is positively or negatively. This has led to dishonest vendors engaging in fraudulent behaviour to try game the recommendation systems for their benefit. They can do this by attempting to boost the perceived desirability of their own products (push attack) or by lowering the ratings of competitors products (nuke attack.) These attacks have been studied as profile injection attacks [26] or as shilling attacks [27]. These attacks usually involve dummy profiles being set up and assumptions on how the recommendation system works. For example, an attack known as the “average attack” [27] needs to assume what the average rating is for each item and then distributes random values across this average, however with high ratings for the item they want to push. An attack such as this one can impact a rating system greatly, however, studies have shown [27] that item-based collaborative filtering can be quite robust to these attacks. Content-based methods which only rely on the active users’ past ratings are generally unaffected by such an attack. Despite the fact pure content-based methods can be unaffected by most fraudulent behaviour, content filtering still has an overall advantage. For example, content filtering just performs much better where there is not many tags or attributes associated with an item, or where it is difficult for the computer to analyse certain items, ideas or opinions. It can also recommend items much more relevant to a user despite the user’s profile not containing much information and merely past ratings. This makes content-based worth it despite being more suspectable to attacks.

## 2.3 Machine Learning Libraries

Machine learning models can be implemented either using a machine learning algorithm from scratch or by using a prebuilt ML library. While building a model using a mathematical toolkit alone could prove educational, it would not be practical to do so. Many machine learning frameworks exist to cut out the development time of creating a neural network with a lot of the complexities out of the way [28].

When choosing a library, the following criteria must be considered, how easy it is to program, the speed at which it runs and how open the source code is and whether it is easy to make changes to the base code. Examples of good machine learning libraries include Tensorflow [29], PyTorch [30], Keras [31] and ML.NET [32].

After reviewing these criteria ML.NET was selected as the library of choice. This is due to the researcher’s proficiency with C# and existing .NET skills. The library was made to be easily integrated into .NET apps [32]. It is also open source and easy to make changes to along with the ability to extend its functionalities with other libraries such as TensorFlow if ML.NET does not include a specific feature.

## 2.4 How a games marketplace can benefit from a recommender system

According to Statistica, the number of games releasing on the Steam store increases year after year. [33] (See Figure 2). Machine learning is ideal for processing the vast libraries of video games available on these distribution platforms whilst still giving a fast recommendation. This will benefit both the storefront and the user. The storefront due to saving time and not having to hire as many employees and for the user as it becomes easier to find games they are interested in. If the store automatically recommends games based on what they already enjoy it will make it more likely for the user to buy the recommended game. Benefitting the storefront with a new sale and for the customer, as they discover a game they may enjoy.

Another benefit of a recommendation system is its ability to improve user retention [34]. Recommendations have the ability to continuously calibrate to a user’s preference. So as time goes on more and more preferred items will appear as it gets to know the user better. This results in the user less likely to switch to a competitor’s site since they know the current site will give them the best recommendations.

According to McKinsey & Company [35], 35% of amazon.com’s revenue is generated by its recommendation engine and 75% of what people watch on Netflix also comes from the recommendation system. A paper written by Netflix executives Carlos A Gomez and Neil Hunt [36] state that machine learning recommendation systems saved the company about $1 billion each year allowing them to invest more money on content. According to Spotify [37] their recommendation system has increased its number of monthly users from 75 to 100 million despite the competition from new rival streaming services such as Apple music. Recommendation systems also helped grow Best Buy. According to CNBC [38] there was a 23.7% growth in sales online. This growth was attributed to the companies improved recommendation system. Another successful implementation of a machine learning recommendation system comes from YouTube [39] where 60% of their video clicks from the homepage come from recommendation systems.

As well as improving the experience for customers, information can be easily gathered from recommendation systems. This information can be used to further improve ad targeting tools. For example, they can choose ad’s more relevant to the user which they are more likely to click on. This will also improve revenue for marketplaces. [34]

Knowing all this information of past success’s it is clear a machine learning recommendation system can vastly improve a games marketplace as it has done for all these other businesses.

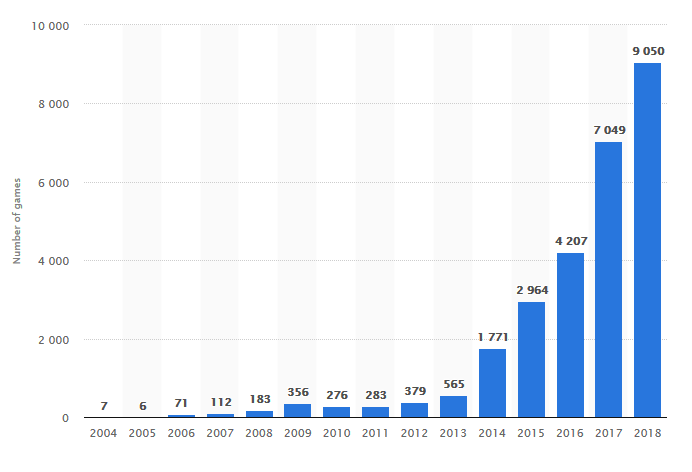


Figure Number of games released on Steam worldwide from 2004 to 2018 [33] (name source)

## 2.5 Summary

In the chapter many of the different recommendation based machine learning models were explained and compared coming to the conclusion that a matrix factorization technique would be best for a games marketplace due to it’s effectiveness over other methods based on recent studies and it’s overall accuracy. In addition which tools should be used to develop the model were discussed where ML.NET was decided to be the best library for the job. Discussions of how to evaluate the accuracy as well as potential issues were made which will prove useful in the following chapters. It was also discussed how a games marketplace could benefit from a machine learning based recommendation model forming the basis as to why this dissertation was started in the first place.

In the next chapter much of the practical designs are set out and subsequentially executed. Much of the lessons learned from the literature review will be used.

# Design and Implementation

In order to meet the aim of the project this section sets out to design and then implement the chosen recommendation model. It will explain how each area was designed and then implemented with each part of the code explained so there is an understanding of how it was achieved.

Before the designs and implementation begin there will be a review of the professional and ethical issues.

## Legal, Social, Ethical and Professional considerations

### 3.1.1 Ethics

Ethics refers to the standards of right and wrong that prescribes what humans are supposed to do in terms of rights, obligations, benefits to society, fairness or specific virtues. [40]

Participants will be involved in the testing of this project, however there will be no potentially harmful testing with users. Simply they will be asked to evaluate the software and answer questions relating to it. The project will adhere to Northumbria university ethics guideline policies. [41] A copy of the ethics form is available in the appendix.

### 3.1.2 Legal

Legal issues need to be considered while working on this project. All software used are legally obtained and have valid licenses. All other libraries such as ML.NET are completely open-source and free to develop with [32]. If any assets such as graphics are used, these will not have any former copyright applied to them. Assets will either be self-created or fair use to avoid legal issues with copyright.

### 3.1.3 Social

As the project would be intended for end-users of diverse backgrounds it is important that the software is appropriate for each person and to not cause offence. For example, if the user is under 18, games suitable for their age range will only be recommended. While collecting information on each game as part of the dataset it would also be good to not include anything that may be deemed too offensive. A safe mode could also be included to filter out such games. Testing on users will not interfere with participants personal lives therefore there is a low risk for any social issues [41].

### 3.1.4 Security

Security can be an issue during the course of this project as much of the data is stored in the cloud. It is possible for this data to be hacked or stolen, so using two-factor authentication for extra protection will be used. When transporting data using memory sticks or portable hard drives extra care will go into making sure these do not go missing. No sensitive data will be collected during the course of this project so that should also not be an issue [41].

### 3.1.5 Professional

It is important to have a level of professionalism throughout. For example, any communication with supervisors must be appropriate and respectful. If any use of copyright material is used the owner will be contacted first before use and credit properly given.

## Model Development

This section outlines the development and implementation of the model along with its requirements and result collection methodology.

### 3.2.1 Tools and Software

The following tools and software will be used throughout the implementation of this project.

* Git – Git will be used in order to keep the program files constantly backed up and to more easily keep track of versions and to go back to previous versions if needed. It also makes it easier to work across multiple computers.
* Dropbox – Dropbox will be used to keep files unrelated to the software backed up in the cloud. This would include documents and diagrams. This also helps in making work across multiple computers easier.
* Visual Studio – Visual Studio is the IDE of choice as the researcher has the most experience with this and works well in conjunction with C# and the following machine learning library.
* ML.NET – ML.NET is the machine learning library of choice. This is due to the researcher’s proficiency with C# and existing .NET skills. The library was made to be easily integrated into .NET apps [13]. It is also open source and easy to make changes to along with the ability to extend its functionalities with other libraries such as TensorFlow if ML.NET does not include a specific feature.
* MATLAB – MATLAB will be used in order to sort, organise and perform functions on the datasets ready for the machine learning to work on.
* Draw.io – Used to create any design diagrams. The software is free and easy to use and includes many of the features needed to create various programming diagrams such as UML.

### 3.2.2 Chosen Model - Matrix Factorization Algorithm with Collaborative Filtering

As discussed in the literature review, matrix factorisation algorithms with collaborative filtering are one of the most explored techniques in recommendation based machine learning algorithms [42]. It works by analysing the past behaviour of users in order to determine the connection they may have to a specific item. Based on the connection that is found between them, a recommendation can be given for comparable items the user may like [42]. Depending on the amount of data available to the developer, different approaches should be used. For example, within the context of movies, watch history may be the only data available. In other cases, some rating data may be available which could give a more accurate prediction when used in conjunction [43].

### 3.2.3 Requirements

In order to achieve the aim some basic requirements were formulated to exactly address whether the aim was met. These were formulated by breaking down the aim by asking questions such as WHAT the recommendation model should do, WHAT the end result should be and finally HOW the model can be proved to be better than previous models.

1. The recommendation model must retrieve recommendations for specified users
2. In order for the prototype to be successful testing participants should judge the recommendations to be accurate.
3. The recommendation-based machine learning model is overall better than the one currently employed by the steam marketplace with evidence to suggest it will improve the experience for both users and the storefront.

### 3.2.4 Design

#### 3.2.4.1 Datasets

In order to train the machine learning model, an appropriate dataset must be gathered containing user ids and the number of hours they have logged in each game and its respected genre. It is important that the dataset is large enough to be able to get accurate predictions. Initially, the idea was to use the steam API to gather the data without outside help however, this proved difficult as the API does not give access to any and all users. A custom program would need to be developed in order to crawl from user to user gathering data as it goes along. However, developing such a program would be a challenge in and of itself so instead, the internet was searched to find if anyone had already produced a dataset as needed.

Two datasets were found on Kaggle (a website where users create datasets) which would need to be combined to meet the needs of the project. These were a dataset which included user data with the hours played for each game but did not include features such as genre or the games ID. [44] And a dataset which included each game, their ID’s, genres, descriptions and more. Though it did not include user data. [45]

In order to meet the exact requirements of the project, these two datasets will need to be combined. This will be discussed in the implementation stage, how Matlab will be utilised to do this. At this stage, the format of the table will need to be designed to just the important features. It would need to include:

* UserID
* GameName
* GameID
* Genres
* Hours

#### 3.2.4.2 Model Design

To help with the implementation of the model, the model can be planned out using UML. It should show how each class relates to each other and show the specific functions and variables that would be needed to implement this. Once planned out in the form of a diagram it becomes much easier to implement each stage in code, as it gives a bigger picture of the whole model. It can be overwhelming to go straight into programming without an idea of the structure or how it can be implemented. Figure shows this UML diagram.

The main class “program” is where the bulk of the code is located. This is shown in the large box where the other classes are called from. The ML.NET library is accessed solely from this class as shown with the double-ended arrow to show both classes have access to each other. An instance of MLContext is created in this class. This is the object which allows the “program” class to call all the machine learning functions present in ML.NET.

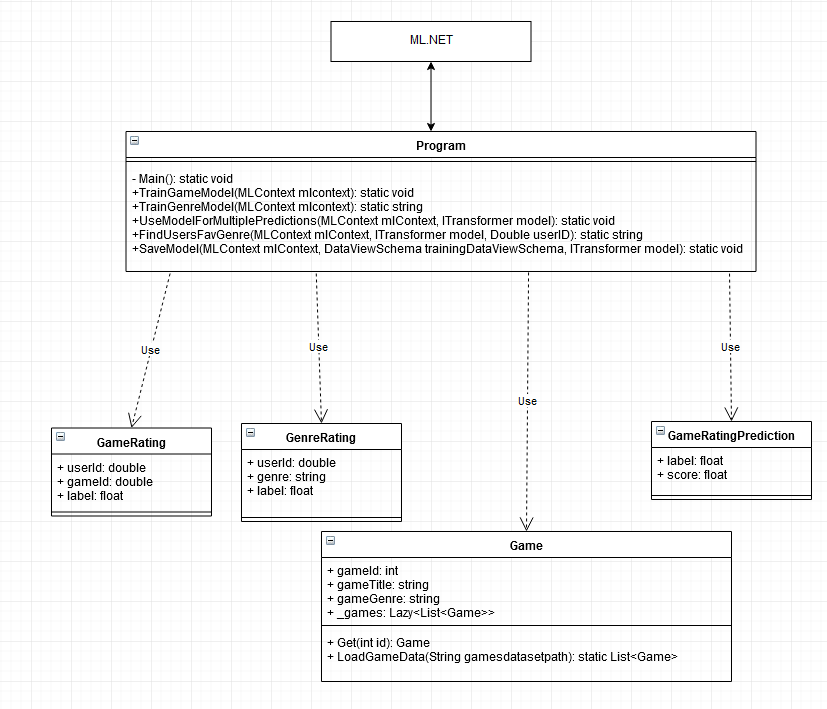


Figure UML Diagram

The diagram also shows the program class should have the following functions:

A function to train the model on the game data, as well as a second to train on the genre of the games.

There will be a function to use the trained models to make a prediction for a specified user. This is within the function “UseModelForMultiplePredictions” as it will give a list of multiple games the user may like within their predicted favourite genre and various other games outside of that genre.

In order for the ML.NET to work three different classes need to be created. These are shown in the boxes below the program class. The first two are GameRating and GenreRating. These classes specify which columns from the.CSV file (dataset) need to be loaded and the name of each column. These classes will be passed to the ML.NET context during the training process.

The third class “GameRatingPrediction” specifies which column is used to make a prediction. This would be named the “score” For example the score column in the case of this dataset would be the number of hours played. The higher the predicted number of hours played for a game the more recommended it is for that user. This is used during the prediction process.

Another class simply titled “Game” is used for programming purposes. It puts every game into a list and will allow the programmer to retrieve information on a game by passing it the game’s ID. When this class is created as an object, an automatic list will be created. Calling the “Get” function and passing it a game ID will retrieve the name of the game. This will also work for the genre and can be expanded in the future with other features such as retrieving a description.

Exactly how these classes will be implemented and programmed will be detailed in the implementation section.

#### 3.2.4.3 User Evaluation Plans

Before user evaluation can commence some plans need to be created on how exactly this should be carried out.

First of all, the participants who will be evaluating the software will need to be chosen. Known users of the steam marketplace, such as friends, acquaintances and mutual friends of the researcher will be the most likely candidates. Additionally, these users will need to have at least 40 games with more than one hour of gameplay as this is a good amount of data to work with in order to predict the most accurate results.

Once users are asked and agree to participate in the study their games will be checked to make sure they are suitable. After this the data will be logged in .csv datasets along with the rest of the data, so that when the machine learning training takes place their data is included in the model. If their data is not present during training, the model will not be able to make a prediction for them. How exactly this will take place will be detailed in the implementation section.

Each participant will be given a document showing which games the machine learning model recommended for them. After which they will be presented with a questionnaire on the following pages. The questions asked will need to be a good gauge of whether the project was a success by finding out if:

* The predicted games are something they would be interested in?
* Would they be likely to buy these games after being recommended?
* How does it compare to recommendation systems on other platforms and if they are likely to buy from them comparatively?

After the results of the questions are viewed, the model can be judged to decide whether it has met the aim of the project. The exact questions of the questionnaire will be shown in the User Evaluation and Model Findings section.

### 3.2.5 Implementation

#### 3.2.5.1 Preparing the datasets

The first step in the process is to prepare the datasets.

As explained in the chapter 3.2.4.1, two separate datasets were combined, as on their own they did not have the appropriate columns to be used with the model. Using the program MATLAB an inner join was performed on the tables in Figure 4 and Figure 6, keeping only the columns for userId, gameName, gameID, genres and hours played. This resulted in the table shown in Figure 7.

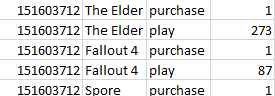


Figure Dataset contains only users and how many hours they play

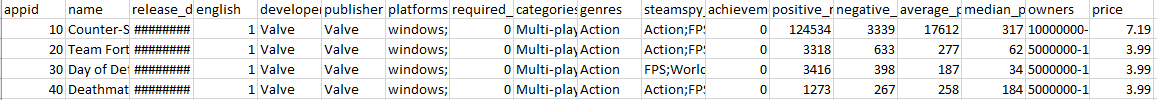


Figure Dataset contains only game data

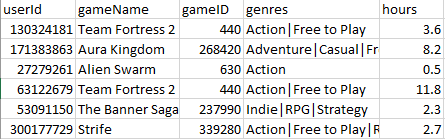


Figure The combined dataset with only relevant data

Once the new table is created, they are split into training and testing datasets. To fit the model the training dataset is used while the test dataset is used to make predictions with the trained model and used to grade the effectiveness. Usually the data within the data set is split 80% to train on and 20% for the testing. Put simply 20% of the items in the dataset are separated and made it’s own CSV file only for testing purposes. This is used as an accuracy test by seeing if the recommendation model can predict the results of the testing dataset based on the 80% training data.

These are both saved in .csv file format and imported into the visual studio project where they can be accessed via code.

#### 3.2.5.2 Programming

**GameRating and GenreRating Class**

As mentioned in chapter 3.2.4.2, a couple of csharp classes are needed in order to tell the machine learning library what each column of the dataset is and what kind of variable they contain.

Figure 7 shows what this looks like. A load column attributes is used to define which column is being loaded for each variable. To train the machine learning model, two features and a label are needed.

The features are the inputs that are given to the model to predict the label. The label is the output of the model.

If you look at Figure 7 you can see the userID is in the first column. This means to add this column [loadColumn(0)] is used because 0 is the first column. The gameID is in the third column so the number 2 is passed into LoadColumn. Finally, the label needs to be named “Label” regardless if it has another name in the table. The label for this model is the number of hours a user plays on a game. The idea is the model will predict the number of hours a user will play on a game and the higher this value is, the more they will be recommended it.

Since the model also needs to be trained on genre data, a class specifying the genre column instead of the gameId is also created. This looks exactly the same as figure , however it has gameID replaced with genre.



Figure GameRating class

**Game Retrieval Class**

Another class simply titled “Game” is used for programming purposes. It places every game into a list and will allow the programmer to retrieve information on any game by passing it the game’s ID. When this class is created as an object, an automatic list will be created. Calling the “Get” function and passing it a game ID will retrieve the name of the game. This will also work for the genre and can be expanded in the future with any number of game information.

Put simply, the class can search through the .csv files for this information. It also uses lazy initialisation to delay the task of having to create a list from the table until it is absolutely needed.

As seen in Figure calling the Get function will return a single instance of a game object from the lazy list. A single game object contains the games title and the games genre. The list is only created at the first reference to the list. So only when this get function is called is the list created. As shown in Figure as soon as the list is created, its contents is set to what is returned in the LoadGameData function which has the path of the database set inside it’s parameters.

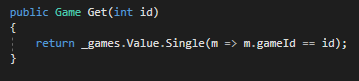
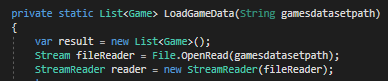


Figure Get Function



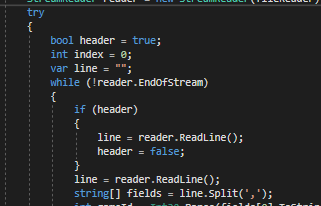
Figure Lazy List

Figure shows the load game data function which converts the .csv dataset to a list of games which includes the title and the genre. First the file is opened into stream and then read into a StreamReader. This allows the program to read each line of code.



Figure

As shown in figure , if there is a header for the table the header is ignored. This is the title of each column. In this case there is a header in the .csv, so the header is ignored. The line after the header is read in. Tables in CSV format separate each column with a ‘,’ so each of these items are split and put into an array of “fields”. This is done with “line.split(‘,’)” which separates each item on that line.



Figure

After this the program is told what each relevant field is as shown in figure . So the “gameID” is set to the first field, the title the second and genre the third. After this an object of type game is added to the ”\_games” list with all those specified fields included.



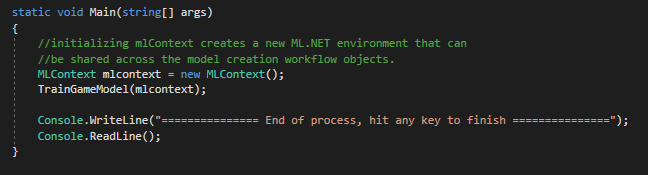
Figure

This will repeat for every single line of the table until the dataset has been completely read. After this, the list has been completely populated and the user will be returned with information for the game they are after.

**Program Class**

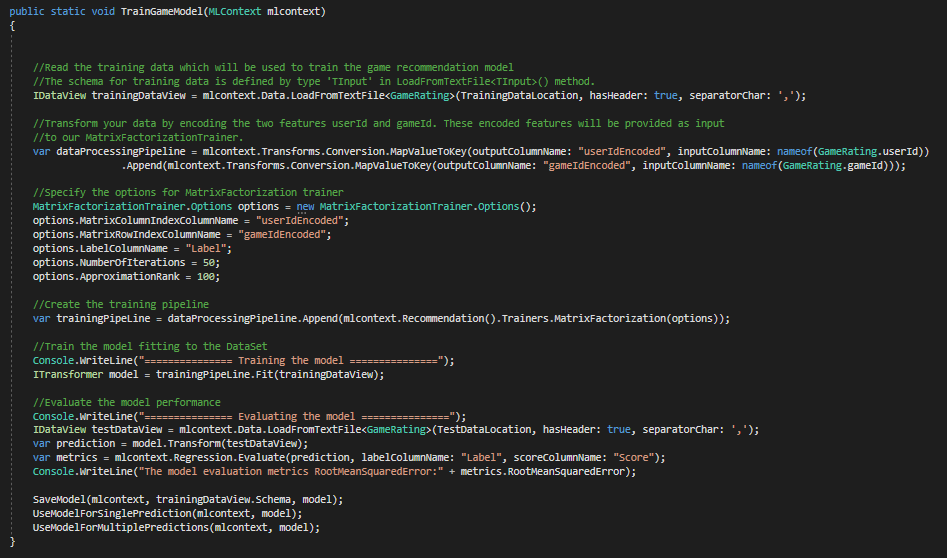
The program class is where most the code is, this is where the training predominantly happens and where all the important functions are located.

The class starts with the main function which contains very little. The machine learning context is created here which is the ML.NET environment needed for the machine learning training. This is passed to a function called “TrainGameModel” as shown in figure .



Figure

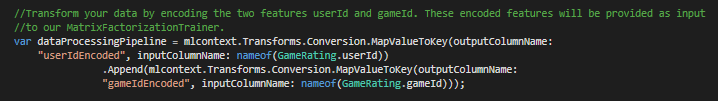
The “TrainGameModel()” function is where the model training takes place. First the training data needs to be read in. The schema is defined by the “GameRating” class that was shown earlier. This tells the schema which column is which.



Figure

It is required that data is in a certain format for it to work properly with machine learning algorithms. Transformers are used on tabular data to transform it into a compatible format. Transformers are created in ML.NET via the creation of estimators. These take in the data and return the transformers. Figure shows the “dataProcessingPipeline” this is the estimator. “UserID” and “GameID” are the defined data transformations. These encoded features are provided as input for the trainer.

The “MapValueToKey()” function is used on these features also in order to transform them into a numeric key type feature column as this is the only format accepted by recommendation algorithms.

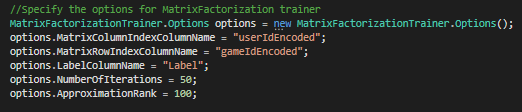


Figure

The recommendation training algorithm that is being implemented is the Matrix Factorization method. This method is explained much more in detail in the literature review. However put simply, it is a kind of “Collaborative Filtering” method which works on the assumption that if one user has the same opinion as a second user about something then that user is also more likely to have a similar opinion to the second user about that same item. In this case it would be different games.

The next thing that needs to be done is to define the options for the method. The Matrix Factorization trainer has several Options. Options such as “Number of Iterations” and “Approximation Rank” can be changed and altered until the best results are found. The number of iterations is the number of times the data is passed through the algorithm. Having more can make the results increasingly accurate, however too many can result in overfitting where it eventually becomes less accurate. It will be talked about in chapter 3.3, which will go over how many iterations were chosen in the end and how the testing lead to this.

The approximation rank is harder to explain but essentially this number can also give more or less accurate results. There is also the computation speed to consider as having too high a number can also take longer with diminishing returns.



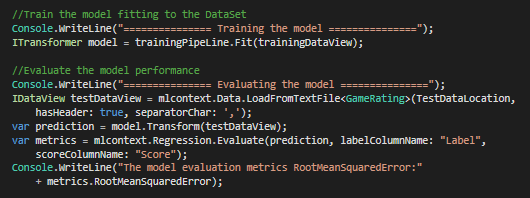
Figure

After this the options are appended to the pipeline to finalise these changes.



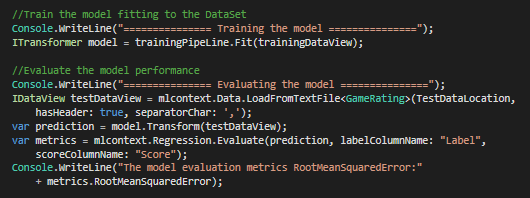
Figure

Finally, the model is trained using the estimator and the transformed dataset. The “Fit()” function executes the estimator definitions by transforming the data and applying the training and returns the trained model (which is a transformer.)



Figure

Now that the model is trained, the test dataset (explained in chapter 3.2.5.1) can now be used to evaluate the performance of the model. The “Transform()” function makes predictions for the many provided rows of the test set. Once the predictions are made the "evaluate()” method is used to give an assessment, which compares the predicted values with the actual labels in the test set. The metrics are returned on how close the results are. These evaluations are printed to the console. How well it does will be talked about in chapter 3.3.



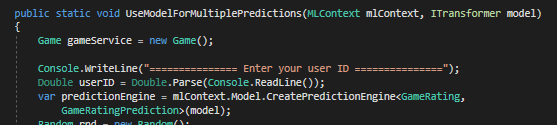
Figure

After this the training was completed and can now use the trained model to make the predictions.

**Making the Prediction**

Now that the model is trained the predictions can be made. A function was created “UseModelForMultiplePredictions()” this is where game predictions for a specified user are made. An instance of the “game” object is created, this is used to pull details about any game as explained earlier whilst talking about the game retrieval class.

The program reads the input of the user with “ReadLine()” the input they give will be the user ID of the user they want to have predictions for. The prediction engine is created in order to make the prediction. It is set using the “mlcontext”, and by passing it the “gamerating” and “gameratingprediction” classes to inform the prediction engine how it will make the prediction. As discussed earlier the “GameRating” class specifies which column of the dataset does what.



Figure

An instance of random is created, this can be created anywhere but is needed to be able to generate random numbers which will be used later. A blank list of strings called games is created, this is used to store the names of games, it’s full use will be explained later.

Next, another function is called “TrainGenreModel()” which will return the users favourite genre. This function also uses machine learning to figure this out in the same way the recommended games are predicted. Due to the similarity of how games are predicted there will be no need to explain exactly how this works.



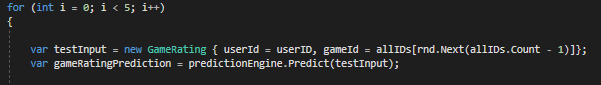
Figure

Next the user is told what the favourite genre predicted is through the console and what games they will enjoy within this genre. This prepares the user for the list they will receive.



Figure

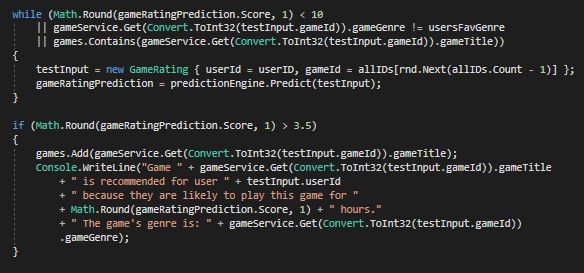
Next a for loop is initiated to perform the same task 5 times, in this case, 5 game predictions are going to be made. A GameRating object is created as input for the prediction. It defines the user as the selected userID and the game as any RANDOM game. This object is then fed into the prediction engine to make a prediction on the game.



Figure

While the result of the prediction is rated less than 10, or the game does not belong to the users favourite genre, or if the game has already been listed as a recommendation; then a new prediction is made on another random game and is repeated until it no longer matches those attributes.

Finally, if the prediction is a success it is checked again to make sure it has a good enough rating to make sure. The game is then consequentially added to the recommended games list and displayed to the user through the console. The user is told how long they are likely to play the game for (this is the predicted score.)



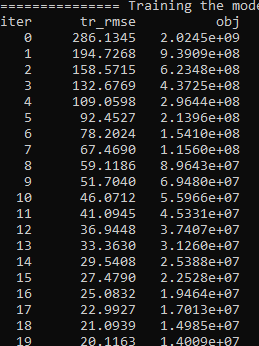
Figure

Once five of these games within the users favourite genre are predicted another loop is initiaited. This time for 10 games. These will be games not within the users favourite genre yet are still recommended for the user.

## 3.3 Testing the model and making changes

Once the program had been fully implemented it was time to test the program and make any necessary changes in order to make it as accurate as possible. To make the model as successful as possible a process of iteratively changing parameters, checking results then altering them again accordingly was used. One of these parameters is how many iterations the model should use. As explained earlier, iterations are the number of times the data is passed through the algorithm.

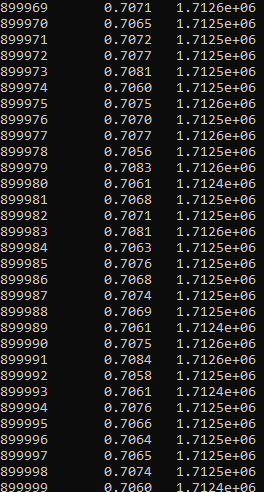
Figuring out how many iterations was a process of trial and error to figure out what gave the best results. To start with, 20 iterations were used. Figure shows the output to the console as the training takes places. It shows that after each iteration the measure of error (second column) decreases. The idea is to get this number as close to 0 as possible. As shown, the measure of error becomes 20.1163 after the 20 iterations is up. This is too high and will not be successful with any predictions thus the number of iterations must be changed.



Figure

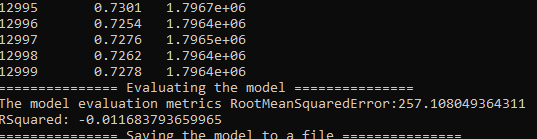
After this the number of iterations was increased to a very high number to observe and see the limits of how much accuracy is possible. 999999 iterations was the chosen number. This made the model train for a full hour, in comparison 20 iterations took 5 seconds. The output of this is shown in figure . Training for this many iterations brought the measure of error down to 0.7060 which is considerably more successful due to its proximity to 0.

However it was noted that when the model got to a measure of error of around 0.73 it became exponentially longer to drop by 0.01, in fact it took 6 minutes to get to 0.73, 15 minutes to get to 0.71 and a full hour to get to 0.70. Looking at this information it was surmised it may be possible that training for longer will not give much benefit. More iteration amounts were tested with additional accuracy tests, this time to fully understand how many iterations are best.

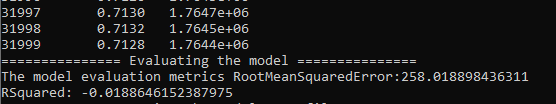


Figure

Looking through the output it was found that after around 13000 iterations the error would get to 0.72 for the first time and after around 32000 iterations it would get down to 0.71. So these two iterations were chosen to compare. The results of these are shown in figures and .



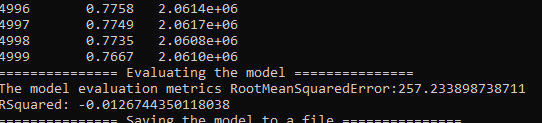
Figure



Figure

An additional metric that can be used to deduct accuracy is the root of mean squared error. It is used to measure the differences between the model predicted values and the observed values of the test dataset. Specifically, it is the square root of the average of the squares of the errors. The lower this value is the better the model is.

It was thought initially that the higher iteration with the lower measure of error would have a more accurate root mean squared error however this was not the case. 13000 had a RMSE of 257.1080 while 32000 had a RMSE of 258.0189. As having the lower RMSE usually denotes more accuracy it would appear that training it for much longer actually made it worse. After this was realised, another test was carried out with an even smaller number of iterations at 5000. As shown in figure , this resulted in a RMSE of 257.2339



Figure

Interestingly this is only slightly worse off in relation to the RMSE than that of the 13000 iterations. But in the end, it’s measure of error was 0.7667 which is quite a bit worse than the 13000 iterations. Due to all these factors it was the latter chosen to be used on the final model.

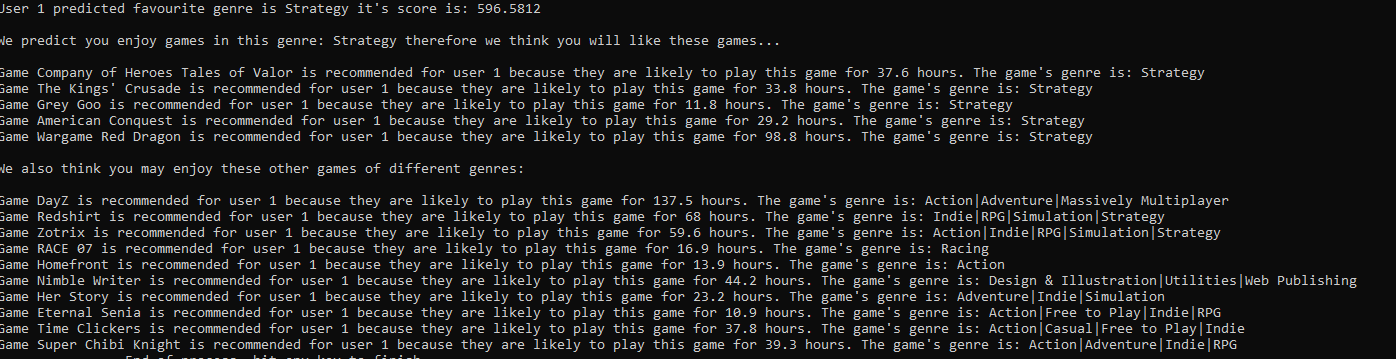
One thing worth mentioning is there was originally a second attribute used to test accuracy. This was R Squared. R squared indicates how well the data fits the model and has a result between 0 and 1. A value of 0 indicates that the data is random and is not fit to be a model. A value of 1 would mean the model matches exactly with the data.

Throughout testing with this, the value would always be very close to 0. This flagged up that something may be wrong with the model and that something must need to be changed. However, nothing would change this value. Through research it was determined this was due to the datasets being based on number of hours a player is likely to play rather than a rating of 1 to 10 which is what the model is usually trained with.

This means that the predicted value could be anything from 1 to 20000 hours. This makes testing accuracy much harder without human subjects as the predicted value is so much broader and therefore would never be dead on 100% when comparing the testing and training sets. Therefore, human participants are much more important for this project in order to truly gauge the accuracy and decide the successfulness of the model to the aim. This is discussed in chapter 4.

Figure shows what the model has output for user 1. This is how the model presents its recommendations to the user in the console.

### Summary



Figure

In this chapter the ethical, social, legal and professional considerations were outlined as well as the tools that were going to be used. After this the recommendation model was designed then implemented with explanations of how each element works. There was also some user testing by the researcher in order to increase the accuracy as much as possible in order to get ready for the user evaluations in the next chapter.

The next chapter will look at the results of the user evaluation and present a judgement of whether the recommendation-based machine learning model is in fact better than the one currently employed by steam.

# User Evaluation and Model Findings

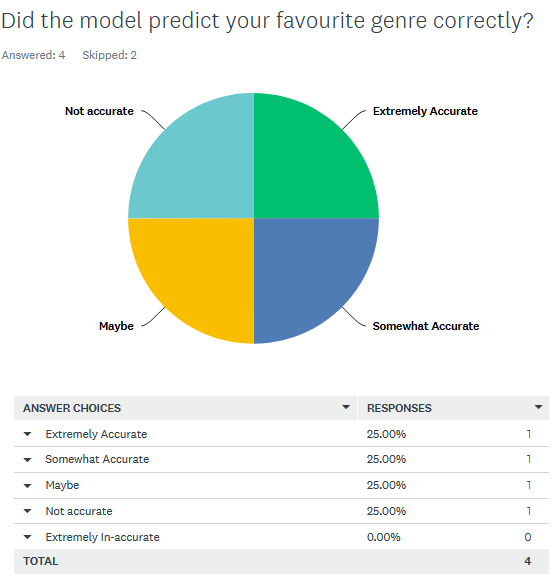
Read chapter 3.2.4.3 to understand how these tests were planned and how participants were chosen.

In order to answer the aim if recommendation-based machine learning models aid video game distribution platforms in user retention and the discoverability of games qualitative research was carried out. The website SurveyMonkey was used to create the questionnaire. After the survey was created a link to complete the survey was sent out to all participants. SurveyMonkey allows easy analysis of the results received from users and is particularly good for use over the internet as there is no need to see the participant in person.

Each participant was sent a pdf document explaining what exactly the recommendation algorithm has predicted for them and images and descriptions for each game predicted. This is to give them a good understanding of what each game is that is predicted for them, and to give them a greater idea if it is something they may enjoy. An example of one of these documents is shown in the appendix.

## 4.1 Did the model predict your favorite genre correctly?

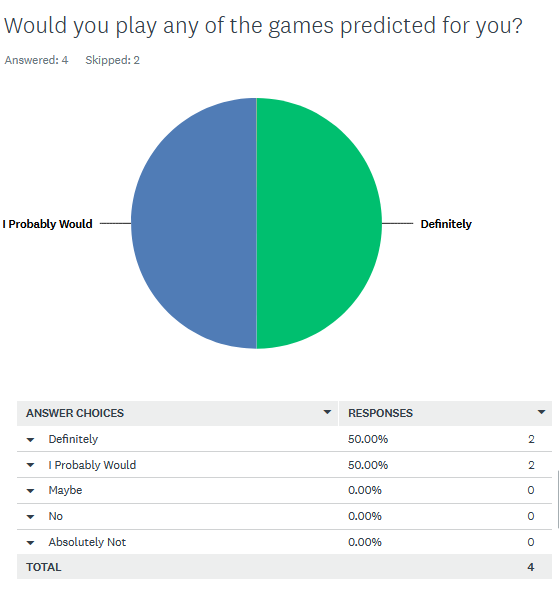
The responses to this question were all over the spectrum with no overall concrete answer. However, no participants felt it was “Extremely In-Accurate.” Overall 50% of participants felt positively on this, 25% unsure and 25% negative. Whether this part is a success is unclear and does not do much to prove the aim.



Figure

## 4.2 Would you play any of the games predicted for you?

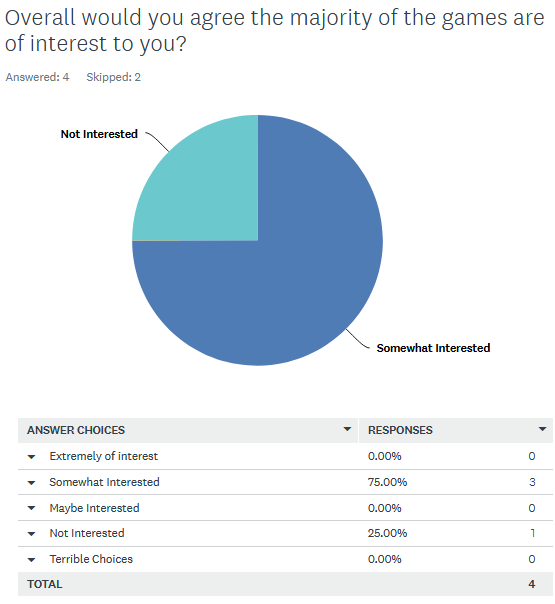
The answer to this question was much more conclusive as 100% of responses were positive with 50% feeling absolute certainty and 50% having a more reserved but probable outlook. This means the model must be predicting games accurately for the most part however more details are needed on how exactly this is and how much of it comes down to luck. The following questions can help answer this.



Figure

## 4.3 Overall would you agree the majority of the games are of interest to you?

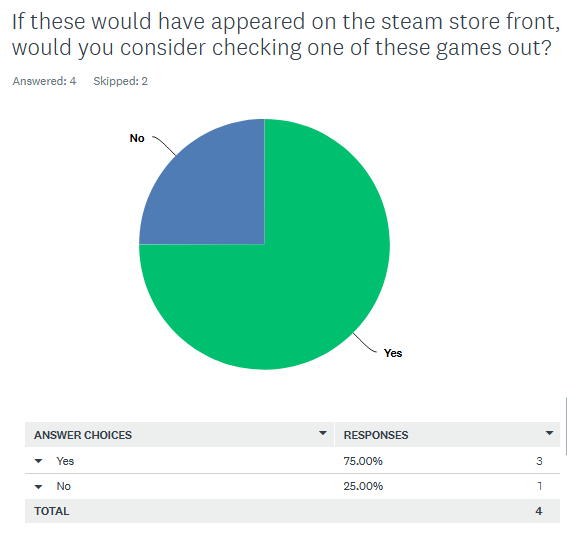
To this question 75% of the responses were positive and 25% were not. This shows that the model has done an excellent job in predicting games that people want to play. Further questions may be able to give insight into why 25% gave a negative response.



Figure

## 4.4 If these were to appear on the steam storefront would you consider checking one of these games out?

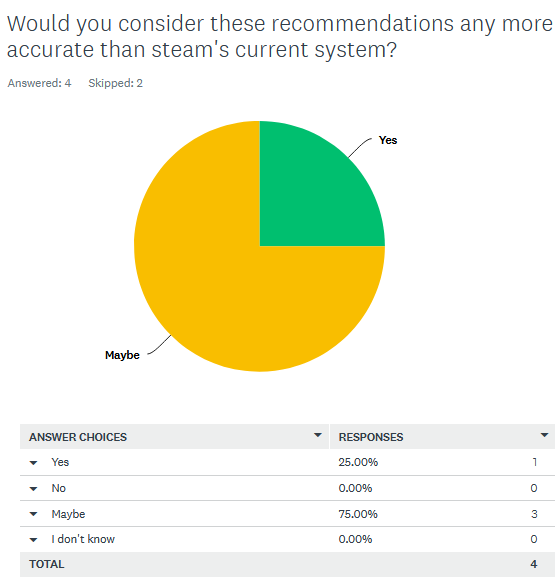
To this question 75% had a positive reaction to this comment and 25% did not. The 25% that said they were not interested expanded further by saying they do not usually look at the games recommended by steam and instead do their own research. This shines some light on the more negative responses from earlier questions. Some users will never be interested in algorithmic recommendations and therefore recommending to them would not be successful.



Figure

## 4.5 Would you consider these recommendations any more accurate than steam’s current system?

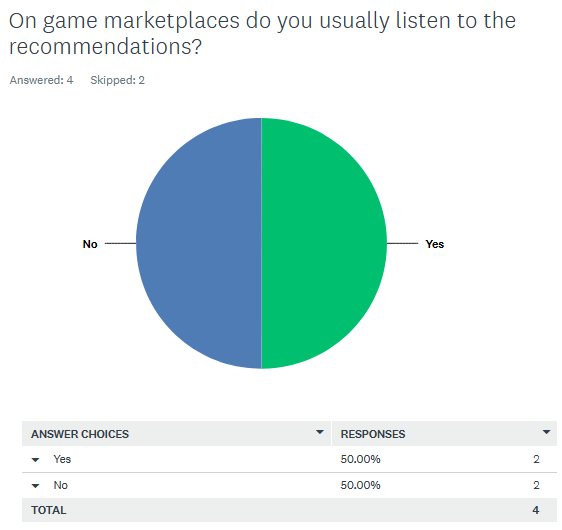
None of the responses to this question were negative however most participants were unsure due to the lack of knowledge of how the steam recommendation system works. And could not form a basis on what to compare the recommendations to. This is backed up by the comments made by the participants.



Figure

## 4.6 On games marketplaces do you usually listen to the recommendations?

The question aims to gauge interest from participants if they listen to marketplaces recommendations in general. This was split 50/50 with the people who do not, commenting that they’d prefer to do their own research and that normally they can find better games than what is recommended for them. This could suggest keeping users retentions on a marketplace will only be effective for half of all customers.

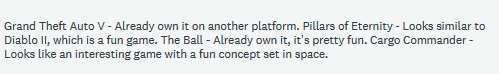


Figure

## 4.7 List the games that are of particular interest to you and explain why

This question helps gauge the accuracy of the recommendations; a substantial list will show the algorithm has done its job. Having the users describe why they are interested helps understand if the algorithm had made connections between the games. From the answers it showed that all users had found games of interest to them, meaning the model is accurate however some users are reluctant to allow an algorithm dictate what they should buy as evidenced from the negative reactions in previous questions.

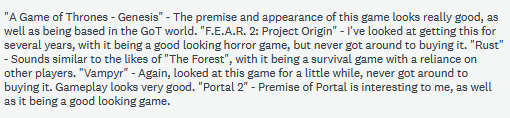
User 1:



User 2:



User 3:



User 4:

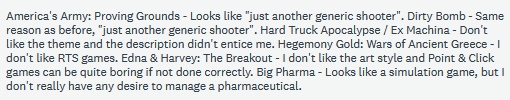


## 4.8 List the games that are not of interest to you and explain why

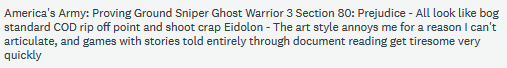
This question also gauges the accuracy of the recommendations. Ideally there should be as few games that are not of interest as possible. Though it unavoidable that a recommendation algorithm will get it wrong sometimes, it should still be kept to a minimum. This information can be cross referenced with the previous question to see with the users with most games in this category have fewer games recommended to them that they do enjoy and perhaps find a reason why the algorithm was having trouble.

From cross referencing it would appear that User 1 had the least successful results. They had the most amount of games that they were not interested in with a relative much less games that do interest them. This could be due to a number of reasons such as they may have put many hours into a game within a genre that they don’t usually like however do enjoy that game for whatever reason. This can skew the results into recommending these kind of games despite the fact that one game is an exception for them. These sorts of bugs can be ironed out with tweaking, however in general it seems it does a good job with 75% of users tested.

User 1:



User 2:



User 3



User 4:



## 4.9 Would you ever buy a game simply because it had been recommended to you and subsequently liked the look of it?

Only half of people surveyed would buy a game that had been recommended to them without any further research other than liking the look at it. This could suggest to make the recommendation more effective more emphasis can be made to show reviews rather than just showing pictures and a short description.



Figure

## 4.10 Do you have any additional comments or suggestions for improvements on predictions? Or anything else you would like to elaborate on that may have been missed?

This is to obtain feedback that could help the recommendation improve in the future or to find out the weaknesses of the algorithm. One user suggested that the algorithm might be looking at the games owned without looking at how many hours they put in. This is false since the algorithm does consider hours so another problem may have caused this user to feel like the algorithm is not accurate enough. Another user suggested that the model could say how strongly it recommends each game. This is absolutely possible, and may bring a big improvement if the user sees that the game has a strength of 95% for example which would compel them to check it out even more.

## 4.11 Findings and Evaluation

Some findings from this user evaluation have been compiled into a table of positive, negative and neutral responses.

|  |  |  |
| --- | --- | --- |
| Positive | Negative | Neutral |
| * 100% of people would play some of the games predicted for them. * 75% felt the majority of games predicted were of interest to them. * 75% would check out one of these games had it been recommended to them on the steam store front. * Half of people surveyed would buy a game that had been recommended to them without any further research other than liking the look at it (easy customers.) | * 75% of people did not know if it was any more accurate than steams current recommendations * Some users are reluctant to allow an algorithm dictate what they should buy. | * 50% were positive and 25% were unsure it predicted their favourite genre. * 50% of users usually listen to the recommendations on games marketplaces. |

To evaluate whether the model was a success let’s again look at the requirements set out in the design stage:

1. The recommendation model must retrieve recommendations for specified users.

2. In order for the prototype to be successful testing participants should judge the recommendations to be accurate.

3. The recommendation-based machine learning model is overall better than the one currently employed by game distribution platforms such as steam, with evidence to suggest it will improve user retention and game discoverability.

The first requirement, for the model to simply retrieve recommendations for specified users is true as it does in fact do this. 100 % of people would play at least **some** of the games predicted for them which shows recommendations are successful.

The second requirement asks if the recommendations are accurate. From looking at these results it would appear that the model has achieved this goal with 75% of participants feeling that the **majority** of games predicted for them were accurately predicted. Though this is not 100%, it still represents the overall majority experience people would have with this recommendation model. It’s to be expected that there would be some issues, as long as there is a high majority of positive results the requirement is successful.

The final requirement takes a lot more analysis of the evaluations to answer with certainty. This is “The recommendation-based machine learning model is overall better than the one currently employed by game distribution platforms such as steam, with evidence to suggest it will improve user retention and game discoverability.” The first thing of note is that 75% of people of people would check out these games had they been recommended to them on the steam marketplace. This shows it is well in-line with the sort of recommendations expected from current game marketplaces. So, this at least shows it is at least **as good** current game marketplaces but the question remains if it is better.

One thing that suggests that the recommender will improve user retention on these platforms is that half of all people surveyed would buy one of these games without any further research. This seems to show they trust the algorithm. 50% might not seem like a lot but the expected result was much lower because in general people do more research into buying games. This keeps users on the marketplace as they insinuate, they have all the information they want in one place. This is a benefit to the marketplace and proves that having a machine recommender-based machine learning algorithm in place can greatly present users with games they would not usually see and 50% of customers are likely to make a purchase at some point due to it.

The part of the question whether it is better than the current system seems to be inconclusive which further tests could help prove. 75% of people answered “maybe” to if it was anymore accurate to steams current recommendations. Although this is not completely negative as no person felt it was less accurate and 25% feeling it was definitely more accurate it still leaves much to be decided. This could have been mitigated by also having participants evaluate steams current system and then have them compare.

Overall the aim of the project “Can recommendation-based machine learning models aid video game distribution platforms in user retention and the discoverability of games?” was arguably a success however it could be argued more testing is needed by comparing it with current systems in order to have a more complete conclusion.

# Project Evaluation

Many challenges were faced through the development of this project and many new skills were developed as well as requirements learned. This section will look at the quality of research and the degree to which objectives were achieved. There will also be discussion on the potential improvements to the project and the learning journey of the author.

## Quality of Research

The literature review was highly important to the rest of the project as it gave the author a greater understanding of the topic as it was being conducted and laid the foundations that the rest of the project can be constructed from. The many different approaches to recommendation algorithms were discussed along with their effectiveness, potential issues and benefits to a marketplace leading to finally being able to choose the best method. This research enabled much greater insight into machine learning and recommendation algorithms to be able to effectively carry out the project into the practical stage.

The literature review seemed daunting at first as figuring out where to start and how to tie all the research together seemed difficult. Initially the literature review felt disconnected with no focus. It eventually became easier after making more notes and having a plan together to give it direction. When the basic points of what the literature review was supposed to achieve were set out the structure of the review came naturally and kept the review relevant to the project.

Finding journals and other sources for recommendation algorithms was not an issue and could be found with relative ease. This is due to the topic being well researched and due to its recent growing popularity across the industry and online marketplaces in general.

The main limiting factor of the research was the researchers lack of understanding of the deep mathematical inner workings of the machine learning models. These were a challenge to understand and present in a way that could be understood on the literature review.

## Learning Journey

A key skill learned during this project was the ability to always have an objective in mind and keeping the focus on that objective. Often during past projects there would not be much direction with many topics simply thrown together to attempt to create a cohesive project. It was unclear to the author where the trouble was lying until this project. When difficulties with having to explain or know what to do next the author would simply think back to what this project was trying to achieve which would bring the explanations back on track.

For example, when forming the testing questions, it was initially difficult to think of the questions and what was needed out of it. When the author thought back to the objective and what needed to be achieved it became much easier to form questions that would be useful. After this realisation the author formed objectives for each section and as a result was much more focused.

Many technical skills were also learned during the project. In particular any kind of work with machine learning and the use of the machine learning library ML.NET. These are excellent skills to have and considering their increased use in the industry, will be largely beneficial to the author in future endeavours.

One are that went particularly well was the programming and implementation of the recommendation model. Despite not having any experience with machine learning, all areas were research thoroughly and understood which enabled the researcher to implement it as smooth as possible with no issues. This was completed before much of the written work was completed.

However, an area which can be improved is time management. The project could have been benefited by sticking to a tighter deadline. For example, the Gantt chart as set out in the research proposal was not adhered to. This resulted in the researcher spending too much time in some areas and neglecting responsibilities in others. This may have led to some chapters feeling more rushed than others due to the mismanagement of time. After having completed this project it is obvious to the researcher that this skill must be developed in order to be more successful in the future.

Despite that, this hasn’t had an overall impact on the projects findings and a conclusion was still drawn, however more time could have been allocated to the planning and performing of the user analysis tests in order to come to a more clear conclusion by performing further tests on unrelated recommendation algorithms for comparison.

## Professional Approach

Ethical issues were completely avoided as there were no vulnerable groups of people involved through the project. All appropriates steps were taken regarding consent and privacy. Those that were involved gave consent and understood exactly what they were doing. Participants were given all the information they needed and as the testing was taking place online, participants digitally agreed to continue with the test by selecting a box essentially giving a digital signature. All data collected was stored securely and no participants names were recorded in order to maintain confidentiality. No names of the participants were used when referring to the tests instead referring to them as User 1, User 2, etc. All communication made as part of the project was conducted via university mail and complied with Northumbria University code of conduct.

## Summary

All together the project has been a productive experience with findings that could benefit the games marketplace’s and research communities based around this. Though challenges were encountered, the projects objective was still met with only slight compromises made.

# Conclusions and Recommendations

The results of this project support the hypothesis that recommendation-based machine learning models aid video game distribution platforms in user retention and the discoverability of games. This was shown by the overall majority of participants agreeing that the recommendations were accurate. This of course will help users discover new games that they would never have played otherwise.

Additionally, research showed that it would also benefit the storefront due to the fact it will improve user retention on these platforms as half of all people surveyed would buy one of these games without any further research. This seems to show they trust the algorithm. 50% might not seem like a lot but the expected result was much lower because in general people do more research into buying games. This keeps users on the marketplace as they insinuate, they have all the information they want in one place. This is a benefit to the marketplace and proves that having a machine recommender-based machine learning algorithm in place can greatly present users with games they would not usually see and 50% of customers are likely to make a purchase at some point due to it. Chapter 4.11 gives a greater analysis of the findings.

Overall, the machine learning algorithm has proved the hypothesis correct, making the project a resounding success.

## Changes and Future Work

Many areas were not fully realised or cut due to time constraints. These were originally proposed in the research proposal. A user interface developed using .NET was the original plan, with users being able to directly use the model themselves. However, after considering the time this was cut out of the final product. Developing the user interface, while useful, was deemed unimportant to the overall aim of the project as this could be answered without needing a user interface. This can be developed further in a future project showing it directly integrated in a steam-like interface.

The most important area which could be developed in a future project, is having extra tests where the same users testing the machine learning model would also test the one used on the steam store. This was never thought until after the user evaluations had ended. This would give a much more concrete answer as the accuracy could be directly compared and users would be ale to make a judgement on which they prefer. This was a common problem in the testing as many of the users did not have much of a basis to compare the machine learning model to a none-machine learning model.

Another thing that was changed from the original plans was that the steam API was going to be used to create the datasets. However, after researching this, it was found to be not possible to do as steam does not give access of this kind of information to regular users. Instead two datasets compiled by another company or person, were found on the internet and combined together using MATLAB in order to create a new dataset which could be used with this project. The only problem with this was the datasets only showed games released up to the year 2015. In the future, a new dataset which is more current could be created. This could be made by developing a custom script which would scrape through the steam servers for as much information as possible. This kind of script would not be able to be developed with the timeframe of this project.

In the end, the work done for this project should form a solid basis for research relating to recommendation-based machine learning models and their applications within the game industry and games marketplaces.

# Bibliography

|  |  |
| --- | --- |
| [1] | H. Deng, "Recommender Systems in Practice," towardsdatascience, 13 Feb 2019. [Online]. Available: https://towardsdatascience.com/recommender-systems-in-practice-cef9033bb23a. [Accessed 22 08 2019]. |
| [2] | B. Kour, "The Rise of Machine Learning and AI is Improving Lives in 2018," SmartDataCollective, 05 01 2018. [Online]. Available: https://www.smartdatacollective.com/rise-of-machine-learning-ai-improving-lives/. [Accessed 31 03 2019]. |
| [3] | A. Smith, "Steam Will Use Machine Learning for Game Recommendations," pcmag, 15 01 2019 . [Online]. Available: https://www.pcmag.com/news/365987/steam-will-use-machine-learning-for-game-recommendations. [Accessed 31 03 2019]. |
| [4] | J. Koza, F. Bennett, D. Andre and M. Keane, "Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming," *Artificial Intelligence in Design ’96,* p. 151–170, 1996. |
| [5] | Daffodil Software, "9 Applications of Machine Learning from Day-to-Day Life," 31 07 2017. [Online]. [Accessed 08 07 2019]. |
| [6] | P. Melville and V. Sindhwani, "Recommender Systems," *Encyclopedia of Machine Learning,* no. 338, pp. 1-8, 2010. |
| [7] | D. Goldberg, D. Nichols, B. Oki and D. Terry, "Using collaborative filtering to weave an information tapestry," *Communications of the Association of Computing Machinery,* no. 35, pp. 61-70, 1992. |
| [8] | P. Resnick, I. Neophytos, P. Bergstrom, S. Mitesh and J. Riedl, "Grouplens: An open architecture for collaborative filtering of netnews," *CSCW94 – Conference on computer supported cooperative work,* pp. 175 - 186, 1994. |
| [9] | M. Pazzani and D. Billsus, "Learning and revising user profiles: The identification of interesting web sites," *Machine Learning,* vol. 27, pp. 313 - 331, 1997. |
| [10] | R. Bell, Y. Koren and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Computer,* vol. 42, pp. 30 - 37, 2009. |
| [11] | M. Pazzani, "A framework for collaborative, content-based and demographic filtering," *Artificial Intelligence Review,* vol. 27, pp. 393 - 408, 1999. |
| [12] | P. Melville, R. Mooney and R. Nagarajan, "Content-boosted collaborative filtering for improved recommendations," *Proceedings of the eighteenth national conference on artificial intelligence,* pp. 187 - 192, 2002. |
| [13] | M. Balabanovic and Y. Shoham, "Fab: Content-based, collabo-rative recommendation," *Communications of the Association for Computing Machinery,* vol. 40, pp. 66 - 72, 1997. |
| [14] | K. Lang, "NewsWeeder: Learning to filter netnews," *Proceedings of the twelfth international conference on machine learning,* pp. 331 - 339, 1995. |
| [15] | R. Mooney and L. Roy, "Content-based book recom-mending using learning for text categorization.," *Proceedings of the fifth ACM conference on digital libraries, San Antonio, Texas,* pp. 195 - 204, 2000. |
| [16] | J. Breese, D. Heckerman and C. Kadie, "Empirical analysis of predictive algorithms for collaborative filtering," *Proceedings of the fourteenth conference on uncertainty in artificial intelligence, Madison, Wisconsin,* 1998. |
| [17] | G. Linden, B. Smith and J. York, "Recommendations: Item-to-item collaborative filtering," *IEEE Internet Computing, ,* vol. 7, pp. 76 - 80, 2003. |
| [18] | B. Sarwar, G. Karypis, J. Konstan and J. Reidl, "Item-based collaborative filtering recommendation algorithms," *WWW'01: Proceedings of the tenth international conference on World Wide Web,* pp. 285-295, 2001. |
| [19] | P. Cotter and B. Smyth, "PTV: Intelligent personalized TV guides," *Twelfth conference on innovative applications of artificial intelligence,* pp. 957 - 964, 2000. |
| [20] | M. Claypool, A. Gokhale and T. Miranda, "Combining content-based and collaborative filters in an online newspaper," *Proceedings of the SIGIR-99 workshop on recommender systems: algorithms and evaluation,* 1999. |
| [21] | J. Herlocker, J. Konstan, L. Terveen and J. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems,* vol. 22, pp. 5-53, 2004. |
| [22] | X. Su, T. Khoshgoftaar, R. Greiner and X. Zhu, "Hybrid collaborative filtering algorithms using a mixture of experts," *Web intelligence,* pp. 645-649, 2007. |
| [23] | X. Su, T. Khoshgoftaar, X. Zhu and R. Greiner, "Imputation-boosted collaborative filtering using machine learning classifiers," *SAC ’08: Proceedings of the 2008 ACM symposium on applied computing,* pp. 949 - 950, 2008. |
| [24] | A. Schein, A. Popescul, L. Ungar and D. Pennock, "Methods and metrics for cold-start recommendations," *SIGIR’02: Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval,* pp. 253 - 260, 2002. |
| [25] | A. Harpale and Y. Yang, "Personalized active learning forcollaborative filtering," *SIGIR ’08: Proceedings of the 31st annual international ACM SIGIR conference on research anddevelopment in information retrieval,* pp. 91 - 98, 2008. |
| [26] | R. Burke, B. Mobasher, R. Bhaumik and C. Williams, "Segment-based injection attacks against collaborative filtering recommender systems," *ICDM ’05: Proceedings of the fifth IEEE international conference on data mining,* pp. 577 - 580, 2005. |
| [27] | S. Lam and J. Riedl, "Shilling recommender systems for fun and profit," *WWW ’04: Proceedings of the 13th international conference on World Wide Web,* pp. 393 - 402, 2004. |
| [28] | V. Amilkanthawar, "Choosing a Deep Learning Framework," Medium, 28 02 2019. [Online]. Available: https://medium.com/in-pursuit-of-artificial-intelligence/choosing-a-deep-learning-framework-5669a85ebc3f. [Accessed 09 07 2019]. |
| [29] | TensorFlow, "TensorFlow," Google Brain Team, 01 04 2019. [Online]. Available: https://www.tensorflow.org/. [Accessed 01 04 2019]. |
| [30] | A. Paszke, S. Gross, S. Chintala and G. Chanan, "PyTorch," [Online]. |
| [31] | François Chollet, "Keras," [Online]. |
| [32] | Z. Ahmed, S. Amizadeh, M. Bilenko, R. Carr, W. Chin, Y. Dekel, X. Dupre, V. Eksarevskiy, E. Erhardt, C. Eseanu and S. Filipi, "Machine Learning at Microsoft with ML. NET," *arXiv preprint arXiv:1905.05715,* 2019. |
| [33] | Statista , "Number of games released on Steam worldwide from 2004 to 2018," 2018. [Online]. Available: https://www.statista.com/statistics/552623/number-games-released-steam/. [Accessed 28 03 2019]. |
| [34] | C. Underwood, "Use Cases of Recommendation Systems in Business – Current Applications and Methods," emerj, 20 May 2019. [Online]. Available: https://emerj.com/ai-sector-overviews/use-cases-recommendation-systems/. [Accessed 16 December 2019]. |
| [35] | I. MacKenzie, C. Meyer and S. Noble, "How retailers can keep up with consumers," McKinsey & Company, October 2013. [Online]. Available: https://www.mckinsey.com/industries/retail/our-insights/how-retailers-can-keep-up-with-consumers. [Accessed 16 December 2019]. |
| [36] | C. Gomez and N. Hunt, "The Netflix Recommender System: Algorithms, Business Value, and Innovation," *ACM Transactions on Management Information Systems ,* vol. 6, no. 4, 2016 . |
| [37] | D. Leonard, "Spotify Is Perfecting the Art of the Playlist," Bloomberg, 21 September 2016. [Online]. Available: https://www.bloomberg.com/news/articles/2016-09-21/spotify-is-perfecting-the-art-of-the-playlist. [Accessed 16 December 2019]. |
| [38] | K. Gustafson, "Best Buy’s surge in online sales shows it won’t be toppled by Amazon," CNBC, 23 August 2016. [Online]. Available: https://www.cnbc.com/2016/08/23/best-buys-surge-in-online-sales-shows-it-wont-be-toppled-by-amazon.html. [Accessed 16 December 2019]. |
| [39] | J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, Y. He, M. Lambert, B. Livingston and D. Sampath, "The YouTube video recommendation system," *Proceedings of the fourth ACM conference on Recommender systems,* pp. 293-296, 2010 . |
| [40] | M. Velasquez, C. Andre, T. Shanks and M. Meyer, "Issues in Ethics," *IIE ,* 1987. |
| [41] | Northumbria University, "Ethics and Governance," March 2019. [Online]. Available: https://www.northumbria.ac.uk/research/ethics-and-governance/. [Accessed March 2019]. |
| [42] | D. Bokde, S. Girase and D. Mukhopadhyay, "Matrix Factorization Model in Collaborative Filtering Algorithms: A Survey," *Procedia Computer Science,* vol. 49, pp. 136-146, 2015. |
| [43] | Microsoft, "Tutorial: Build a movie recommender using matrix factorizaton with ML.NET," Microsoft, 06 05 2019. [Online]. Available: https://docs.microsoft.com/en-gb/dotnet/machine-learning/tutorials/movie-recommendation#other-recommendation-algorithms. [Accessed 09 07 2019]. |
| [44] | Tamber, "Recommend video games from 200k steam user interactions.," Kaggle, 01 01 2016. [Online]. [Accessed 09 11 2019]. |
| [45] | N. Davis, "Steam Store Games (Clean dataset)," Kaggle, 1 May 2019. [Online]. Available: https://www.kaggle.com/nikdavis/steam-store-games/version/3. [Accessed 7 November 2019]. |
| [46] | S. Clayton. [Online]. Available: https://www.codeproject.com/Articles/1232150/Building-a-Recommendation-Engine-in-Csharp. |
| [47] | "Tutorial: Build a movie recommender using matrix factorizaton with ML.NET," [Online]. Available: https://docs.microsoft.com/en-us/dotnet/machine-learning/tutorials/movie-recommmendation. |
| [48] | IEEE, "Transactions on Neural Networks and Learning Systems," 2019. [Online]. Available: https://cis.ieee.org/publications/t-neural-networks-and-learning-systems. [Accessed 31 03 2019]. |
| [49] | M. Pazzani and D. Billsus, "Content-Based Recommendation Systems," *The Adaptive Web,* vol. 4321, pp. 325-341, 2007. |
| [50] | A. Sunita , "Combination of machine learning algorithms for recommendation of courses in E-Learning System based on historical data," *Knowledge-Based Systems,* vol. 51, pp. 1-14, 2013. |
| [51] | L. Barrington, D. Turnbull and G. Lanckriet, "Game-powered machine learning," *PNAS,* 2012. |
| [52] | Steam, "Steam Web API Documentation," Valve, 01 04 2019. [Online]. Available: https://steamcommunity.com/dev. [Accessed 01 04 2019]. |
| [53] | J. Koza and H. Forrest . [Online]. Available: https://link.springer.com/chapter/10.1007%2F978-94-009-0279-4\_9. |
| [54] | Google Brain Team, "TensorFlow," [Online]. |

1. Research Proposal



Faculty of Engineering and Environment

**KF7028 Research and Project Management**

**Assignment: *Research Proposal***

*Can recommendation-based machine learning models aid video game distribution platforms in user retention and the discoverability of games?*

Module: KF7028 – Research and Project Management

Module tutor: Shelagh Keogh

Supervisor: Ian Watson

Second marker name: Naveed Anwar

Assignment title: MSc Research Proposal'

Student name and your university identifier: Elliot Anderson 15007132

Programme or course of study: Advanced Computer Science

Confirmation that you have completed the ethical approval process: Yes/No

Date arranged by you with your supervisor and second marker for the review: Pending (Monday 2nd or Tuesday 3rd September)

Word Count: 3032

Contents

[1. Aim 3](#_Toc17405939)

[2. Background, Motivation and Relevance –literature review 3](#_Toc17405940)

[2.1. Background 3](#_Toc17405941)

[Matrix Factorization Algorithm with Collaborative Filtering 3](#_Toc17405942)

[One Class Matrix Factorization 3](#_Toc17405943)

[Field Aware Factorization Machines 3](#_Toc17405944)

[Machine Learning Libraries 3](#_Toc17405945)

[Relevance 4](#_Toc17405946)

[2.2. Motivation 5](#_Toc17405947)

[2.3. Sources and use of Knowledge 5](#_Toc17405948)

[3. Scope, Objectives and Risk 7](#_Toc17405949)

[3.1. Scope 7](#_Toc17405950)

[3.2. Objectives 7](#_Toc17405951)

[3.3. Task List 8](#_Toc17405952)

[3.4. Risk Log 9](#_Toc17405953)

[4. Ethics, Legal, Social, Security and Professional Issues 12](#_Toc17405954)

[4.1. Ethics 12](#_Toc17405955)

[4.2. Legal 12](#_Toc17405956)

[4.3. Social 12](#_Toc17405957)

[4.4. Security 12](#_Toc17405958)

[4.5. Professional 12](#_Toc17405959)

[5. SCHEDULE OF ACTIVITIES 12](#_Toc17405960)

[Bibliography 13](#_Toc17405961)

[Appendices 15](#_Toc17405962)

[Appendix A – Task List 15](#_Toc17405963)

[Appendix B - Work Breakdown Structure 21](#_Toc17405964)

[Appendix C - Gantt chart 25](#_Toc17405965)

[Appendix D - Ethics Form 26](#_Toc17405966)

# 1. Aim

Can recommendation-based machine learning models aid video game distribution platforms in user retention and the discoverability of games?

# 2. Background, Motivation and Relevance –literature review

## 2.1. Background

Machine learning (a subset of artificial intelligence) is described as the study of algorithms and statistical models that computer systems can use to perform tasks without being giving exact instructions, relying only on interference and patterns. Machine learning algorithms utilise a gathering of training data to make predictions without being programmed to perform the task [1].

Machine learning has many potential applications and can be tailored to each application to perform a specific task [1]. Its application can be used across many different fields including economics, marketing, linguistics and image recognition [2]. In this study recommendation-based machine learning models will be looked at as these are the most suitable for the purpose of giving recommendations of video games based on other games the player is interested in and the genres in which a specific game belongs to [3].

### Matrix Factorization Algorithm with Collaborative Filtering

Matrix factorization algorithm with collaborative filtering is one of the most explored techniques in recommendation algorithms [4]. It works by analysing the past behaviour of users in order to determine the connection they may have to a specific item. Based on the connection that is found between them, a recommendation can be given for comparable items the user may like [4]. Depending on the amount of data available to the developer, different approaches should be used. For example, within the context of movies watch history may be the only data available. In other cases, some rating data may be available which could give a more accurate prediction when used in conjunction [5].

### One Class Matrix Factorization

One class matrix factorisation is the technique used when there is no rating data available [5]. The only features available may only be the ID of the user and the ID of the item they like or used. This style of recommendation is based upon the co-purchase scenario, or products frequently bought together, which means it will recommend to customers a set of products based upon their own purchase order history [5].

### Field Aware Factorization Machines

Field Aware Factorization Machines is the more accurate of these methods as it takes into account multiple features to make a recommendation [5]. This could be a mix of features such as user ID, product ID, description, price and tags. This would be the best method to use for a recommendation system in a games marketplace as all these features will be available to the software developer using the chosen marketplace API.

### Machine Learning Libraries

Machine learning models can be implemented either using an ML algorithm from scratch or by using a prebuilt ML library. While building a model using a mathematical toolkit alone could prove educational, it would not be practical to do so. Many machine learning frameworks exist to cut out the development time of creating a neural network with a lot of the complexities out of the way [6].

When choosing a library, the following criteria must be considered, how easy it is to program, the speed at which it runs and how open the source code is and whether it is easy to make changes to the base code. Examples of good machine learning libraries include Tensorflow [7], PyTorch [8], Keras [9] and ML.NET [10].

After reviewing these criteria ML.NET was selected as the library of choice. This is due to the researcher’s proficiency with C# and existing .NET skills. The library was made to be easily integrated into .NET apps [10]. It is also open source and easy to make changes to along with the ability to extend its functionalities with other libraries such as TensorFlow if ML.NET does not include a specific feature.

## Relevance

Increasingly machine learning is being used across many industries. This is due to machine learning’s ability to process vast amounts of data (such as user information like search history and content interactions) and then offer personalised data to the user based on these factors [3]. Companies such as Netflix utilise machine learning algorithms which allowed them to save almost $1 billion, due to the ability to recommend personalised tv shows and movies [11].

One industry that could benefit from machine learning, is within the field of online video game distribution. The largest platform for this, Steam, does not currently use machine learning for its recommendation systems. Though they do have plans for this in the future [12]. Research into utilising machine learning for this market would bring many benefits to the game industry, in the same way it has helped entertainment platforms such as netflix [11].

According to Statistica, the number of games releasing on the Steam store increases year after year. [13] (See Figure 1). Machine learning is ideal for processing the vast libraries of video games available on these distribution platforms whilst still giving a fast recommendation. This will benefit both the storefront and the user. The storefront due to saving time and not having to hire as many employees and for the user as it becomes easier to find games they are interested in. If the store automatically recommends games based on what they already enjoy it will make it more likely for the user to buy the recommended game. Benefitting the storefront with a new sale and for the customer as they discover a game they may enjoy.

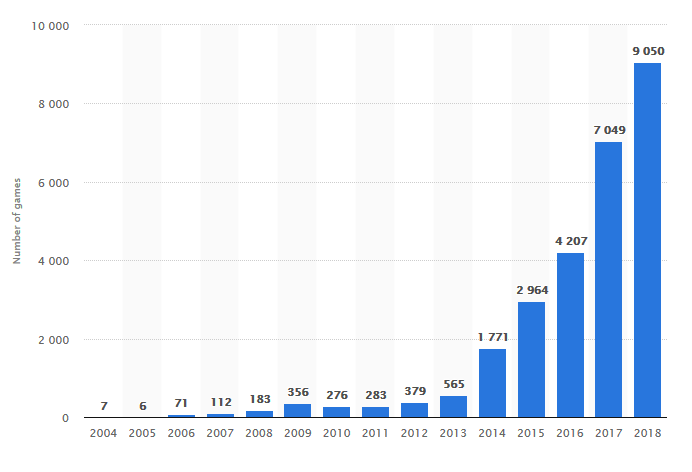


Figure Number of games released on Steam worldwide from 2004 to 2018 [13]

## 2.2. Motivation

The number of video games are increasing on a daily basis. This encompasses many different genres, where it becomes easier for the user to miss out on games that may have interested them. As the researcher has an interest in games and is always looking out for new releases that may be obscure or hard to find, research into ways such games can be discovered seemed suitable. In addition, the researcher has been following the advances of machine learning for many years and is something he is deeply interested in. Combining these two interests and creating something that would also benefit not only the researcher but the many people who play games is a big part of why this project was decided.

## 2.3. Sources and use of Knowledge

A relevant journal for this research is “Transactions on Neural Networks and Learning Systems published” by IEEE. This project fits within the aims and scope of the journal in many ways. The journal states it deals with the theory, design, and applications of neural networks and related learning systems [14]. This fits perfectly with the project as it aims to apply neural networks (machine learning) to a content-based recommendation system.

The project will fit within the body of knowledge pertaining to machine learning, and content-based recommendation systems. There are several studies relating to this such as one by M. Pazzani and D. Billsus which gives a general overview about how a system could work [15]. It does not focus on using machine learning or within the field of video games. This project will go more in depth to have it tailored for use in games media. To go deeper into what this project aims to achieve another study by Sunita talks about combining machine learning algorithms for recommendation of courses in E-learning systems based on historical data [16]. Knowledge from this study can assist further with this project as it is a much more in-depth look on using machine learning for recommendation systems in particular. However, the subject matter does not relate to the recommendation of video games though it still can be applied in a similar way to this project.

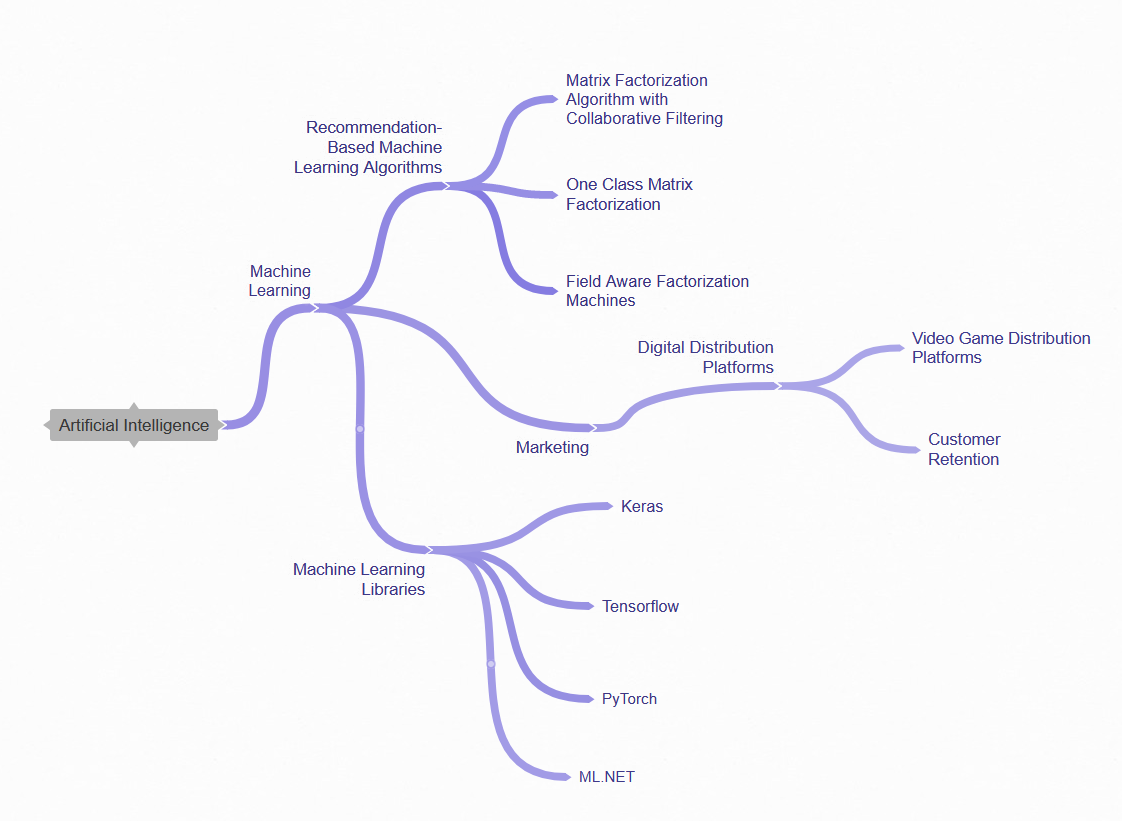


Figure Relevant body of knowledge Diagram

# 3. Scope, Objectives and Risk

## 3.1. Scope

The scope of the project is to research how using machine learning can benefit video game distribution platforms. This will involve developing a ML model which will be integrated into a prototype program which will allow users to input their Steam ID to receive recommendations based on their profile. The ML model will learn what kinds of games a user typically plays based on tags and assigned genres. The model will look at what other users play of similar game preferences, then make a prediction based on this. A machine learning framework ML.NET will be used to provide the machine learning capabilities and the steam API will give access to the game database which will be used as training data for the algorithm [17]. WinForms and C# will be used to develop the user interface. Black box tests will be performed on participants who have previously used steam and will be asked how accurate the system is in choosing games they would enjoy. The results will be graded to decide whether the project was a success based on the participants feedback.

## 3.2. Objectives

In order to achieve the aim “Can recommendation-based machine learning models aid video game distribution platforms in user retention and the discoverability of games?” the following objectives must be met.

**Objective 1:** Produce a literature review to compare the best recommendation-based machine learning models to find the most suitable to use with research

**Deliverable**: A background is needed first on the different models in order to begin development of a fitting model for the software. A literature review covering what machine learning is and its history, recommendation-based machine learning algorithms with a comparison of each one, how recommendation-based machine learning can be used in the industry and how this can keep user retention in a video game distribution platform and a conclusion which algorithm is best fitted for the project.

**Quality Measure:**

* Number and quality of Sources used.
* Breadth and depth of literature review and the amount of research undertaken for it.
* Has a solid conclusion on which the model will be developed upon.

**Objective 2:** Design the prototype software based on the chosen model and detail the research methodology such as how the tests will be carried out

**Deliverable:** A written chapter detailing the designs of the software. This will include a list of requirements that the design must adhere to based on the criteria set out by the literature review. Also design diagrams will be produced in the form of UML and ERD’s and finally a detailed plan on how the research and testing will be carried out with users.

**Quality Measure:**

* Workable designs that are fit for purpose.
* Detail covered of how research will be carried out.
* Diagrams can be followed and used easily for programming in the development stage.

**Objective 3:** Build a working prototype of the software with the functionality of being able to feed users profiles to get recommendations

**Deliverable:** The Final software in .exe format, it will also include all the code in C# and the files runnable from visual studio from source code.

**Quality Measure:**

* The software works and is ready to be tested with users.
* Software matches designs of the previous objective.
* Minimal bugs present in software.
* Code quality.

**Objective 4:** Investigate the effectiveness of the ML model by using the software to perform tests on participants

**Deliverable:** Participant test results recorded in tabular format, a discussion of the results of the test and an evaluation of what the participants thought overall, evaluating whether it was a success or failure based on if the participant answers were positive or negative.

**Quality Measure:**

* Questions are good at gauging the outcome of the test
* Results from users are extensive and good at evaluating the effectiveness of the software.
* Discussion of the results are conclusive and give a thorough overview of how it went.

**Objective 5:** Complete recommendations and conclusions based on the tests and an evaluation of the project overall

**Deliverable:** A written conclusion talking about if the objectives have been sufficiently met, an evaluation of the overall project, discussion about what went well and what could have been done differently.

**Quality Measure:**

• Justified recommendations and conclusions.

• Covers as much as the project as possible, talks about each objective.

• Explains in detail what went well and what went wrong throughout the project.

## 3.3. Task List

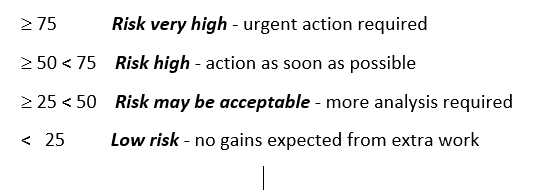
The task list in tabular format is shown in Appendix A, this goes much more in detail of each task and their deliverables, resources required and skills.

## 3.4. Risk Log

Key

Risk types – F (Financial), T (Technology), P (People), E(Environmental), S(Security) Add as necessary

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Risk Type | Risk Event | Likelihood  (1-10) | Impact  (1-10) | Risk Value  (1-100) | Risk Monitoring/Control Flag | Risk Management Strategy | Risk Review date | Risk owner | Commentary |
| T | Software does not yield positive results in relation to the aim | 4 | 4 | 16 | Reviewed after testing with participants | In the event the software does not prove the aim, talk about why it fails to do so and what has been learned from the project. |  | Researcher | As the aim is a question, the answer to this may well be a “no.” Therefore it would be acceptable if the software did not yield positive. |
| T | Unfamiliarity with chosen libraries and API’s such as ML.NET or the steam API | 2 | 7 | 14 | Reviewed before the development stage has commenced | Time will be needed to develop skills with these libraries and to be familiarised enough with them in order to develop the software effectively. This will be done before the development stage of the project. |  | Researcher | Failure to properly learn these libraries can lead to a prototype that does not work at all. This could mean the failure of the project. Therefore, it is important to learn how to use these libraries adequately. |
| F | Funds unavailable to travel to university | 1 | 7 | 7 | This should be reviewed from the beginning | The researcher will set aside a certain amount of funds needed to travel into university well in advance so that the researcher will always be able to go in for important occasions such as meetings or testing with users |  | Researcher | Without the required funds the project would be much harder to carry out as it is sometimes needed to meet with supervisors for meetings or to meet participants for testing purposes. |
| T | Loss of data and work | 2 | 9 | 18 | Review every time backed up and saving of work | Backup work regularly, have copies on the cloud and multiple devices. Don’t keep everything in one place. |  | Researcher | It is the duty of the researcher to make sure all work completed is always available. Losing data can be disastrous to the project and can delay work further. |
| S | Data stolen/ Device Hack | 1 | 8 | 8 | Review services of stored data | Do not store all data online, keep files locally available, use strong passwords and two factor authentication. |  | Researcher | Trusting other services with data is not always wise so having local backups is also important. |
| P | Lack of planning | 4 | 6 | 24 | Reviewed when project work is commencing | Make sure planning has been completed to an adequate level before the project has started. Gantt charts and work breakdown structures are useful for keeping on track. |  | Researcher | Failure to adequately plan can lead to a disorganised dissertation without direction thus making the entire project poor in quality. |
| P | Availability of student and supervisor for meetings | 5 | 4 | 20 | Should be reviewed whenever a meeting is needed | In order to keep up good feedback it is important the researcher and supervisor are up to date on when they are available in order to plan meetings or work reviewals. |  | Researcher & Supervisor | It is important to inform on availability and to keep up communications with the supervisor as feedback will greatly improve the overall quality of the dissertation. |



# 4. Ethics, Legal, Social, Security and Professional Issues

## 4.1. Ethics

Ethics refers to the standards of right and wrong that prescribes what humans are supposed to do in terms of rights, obligations, benefits to society, fairness or specific virtues. [17]

Participants will be involved in the testing of this project, however there will be no potentially harmful testing with users. Simply they will be asked to test the software and answer questions relating to it. The project will adhere to Northumbria university ethics guideline policies. [18] A copy of the ethics form is available in appendix D.

## 4.2. Legal

Legal issues need to be considered while working on this project. All software used are legally obtained and have valid licenses. All other libraries such as ML.NET are completely open source and free to develop with [10]. If any assets such as graphics are used, these will not have any former copyright applied to them. Assets will either be self-created or fair use to avoid legal issues with copyright.

## 4.3. Social

As the project would be intended for end users of diverse backgrounds it is important that the software is appropriate for each person and to not cause offense. For example, if the user is under 18, games suitable for their age range will only be recommended. While collecting information on each game as part of the dataset it would also be good to not include anything that may be deemed too offensive. A safe mode could also be included to filter out such games. Testing on users will not interfere with participants personal lives therefore there is low risk for any social issues [18].

## 4.4. Security

Security can be an issue during the course of this project as much of the data is stored in the cloud. It is possible for this data to be hacked or stolen, so using two factor authentication for extra protection will be used. When transporting data using memory sticks or portable hard drives extra care will go into making sure these do not go missing. No sensitive data will be collected during the course of this project so that should also not be an issue [18].

## 4.5. Professional

It is important to have a level of professionalism throughout. For example, any communication with supervisors must be appropriate and respectful. If any use of copyright material is used the owner will be contacted first before use and credit properly given.

# 5. SCHEDULE OF ACTIVITIES

Appendix B shows the work breakdown structure and Appendix C shows the Gantt chart. These identify key dates for each task and their deliverables.

# RESEARCH PROPOSAL Bibliography

|  |  |
| --- | --- |
| [1] | J. Koza, F. Bennett, D. Andre and M. Keane, “Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming,” *Artificial Intelligence in Design ’96,* p. 151–170, 1996. |
| [2] | Daffodil Software, “9 Applications of Machine Learning from Day-to-Day Life,” 31 07 2017. [Online]. [Accessed 08 07 2019]. |
| [3] | H. Deng, “Recommender Systems in Practice,” towardsdatascience, 13 Feb 2019. [Online]. Available: https://towardsdatascience.com/recommender-systems-in-practice-cef9033bb23a. [Accessed 22 08 2019]. |
| [4] | D. Bokde, S. Girase and D. Mukhopadhyay, “Matrix Factorization Model in Collaborative Filtering Algorithms: A Survey,” *Procedia Computer Science,* vol. 49, pp. 136-146, 2015. |
| [5] | Microsoft, “Tutorial: Build a movie recommender using matrix factorizaton with ML.NET,” Microsoft, 06 05 2019. [Online]. Available: https://docs.microsoft.com/en-gb/dotnet/machine-learning/tutorials/movie-recommendation#other-recommendation-algorithms. [Accessed 09 07 2019]. |
| [6] | V. Amilkanthawar, “Choosing a Deep Learning Framework,” Medium, 28 02 2019. [Online]. Available: https://medium.com/in-pursuit-of-artificial-intelligence/choosing-a-deep-learning-framework-5669a85ebc3f. [Accessed 09 07 2019]. |
| [7] | TensorFlow, “TensorFlow,” Google Brain Team, 01 04 2019. [Online]. Available: https://www.tensorflow.org/. [Accessed 01 04 2019]. |
| [8] | A. Paszke, S. Gross, S. Chintala and G. Chanan, “PyTorch,” [Online]. |
| [9] | François Chollet, “Keras,” [Online]. |
| [10] | Z. Ahmed, S. Amizadeh, M. Bilenko, R. Carr, W. Chin, Y. Dekel, X. Dupre, V. Eksarevskiy, E. Erhardt, C. Eseanu and S. Filipi, “Machine Learning at Microsoft with ML. NET,” *arXiv preprint arXiv:1905.05715,* 2019. |
| [11] | B. Kour, “The Rise of Machine Learning and AI is Improving Lives in 2018,” SmartDataCollective, 05 01 2018. [Online]. Available: https://www.smartdatacollective.com/rise-of-machine-learning-ai-improving-lives/. [Accessed 31 03 2019]. |
| [12] | A. Smith, “Steam Will Use Machine Learning for Game Recommendations,” pcmag, 15 01 2019 . [Online]. Available: https://www.pcmag.com/news/365987/steam-will-use-machine-learning-for-game-recommendations. [Accessed 31 03 2019]. |
| [13] | Statista , “Number of games released on Steam worldwide from 2004 to 2018,” 2018. [Online]. Available: https://www.statista.com/statistics/552623/number-games-released-steam/. [Accessed 28 03 2019]. |
| [14] | IEEE, “Transactions on Neural Networks and Learning Systems,” 2019. [Online]. Available: https://cis.ieee.org/publications/t-neural-networks-and-learning-systems. [Accessed 31 03 2019]. |
| [15] | M. Pazzani and D. Billsus, “Content-Based Recommendation Systems,” *The Adaptive Web,* vol. 4321, pp. 325-341, 2007. |
| [16] | A. Sunita , “Combination of machine learning algorithms for recommendation of courses in E-Learning System based on historical data,” *Knowledge-Based Systems,* vol. 51, pp. 1-14, 2013. |
| [17] | M. Velasquez, C. Andre, T. Shanks and M. Meyer, “Issues in Ethics,” *IIE ,* 1987. |
| [18] | Northumbria University, “Ethics and Governance,” March 2019. [Online]. Available: https://www.northumbria.ac.uk/research/ethics-and-governance/. [Accessed March 2019]. |
| [19] | L. Barrington, D. Turnbull and G. Lanckriet, “Game-powered machine learning,” *PNAS,* 2012. |
| [20] | Steam, “Steam Web API Documentation,” Valve, 01 04 2019. [Online]. Available: https://steamcommunity.com/dev. [Accessed 01 04 2019]. |
| [21] | J. Koza and H. Forrest . [Online]. Available: https://link.springer.com/chapter/10.1007%2F978-94-009-0279-4\_9. |
| [22] | Google Brain Team, “TensorFlow,” [Online]. |

# 

# Appendices

## RESEARCH PROPOSAL Appendix A – Task List

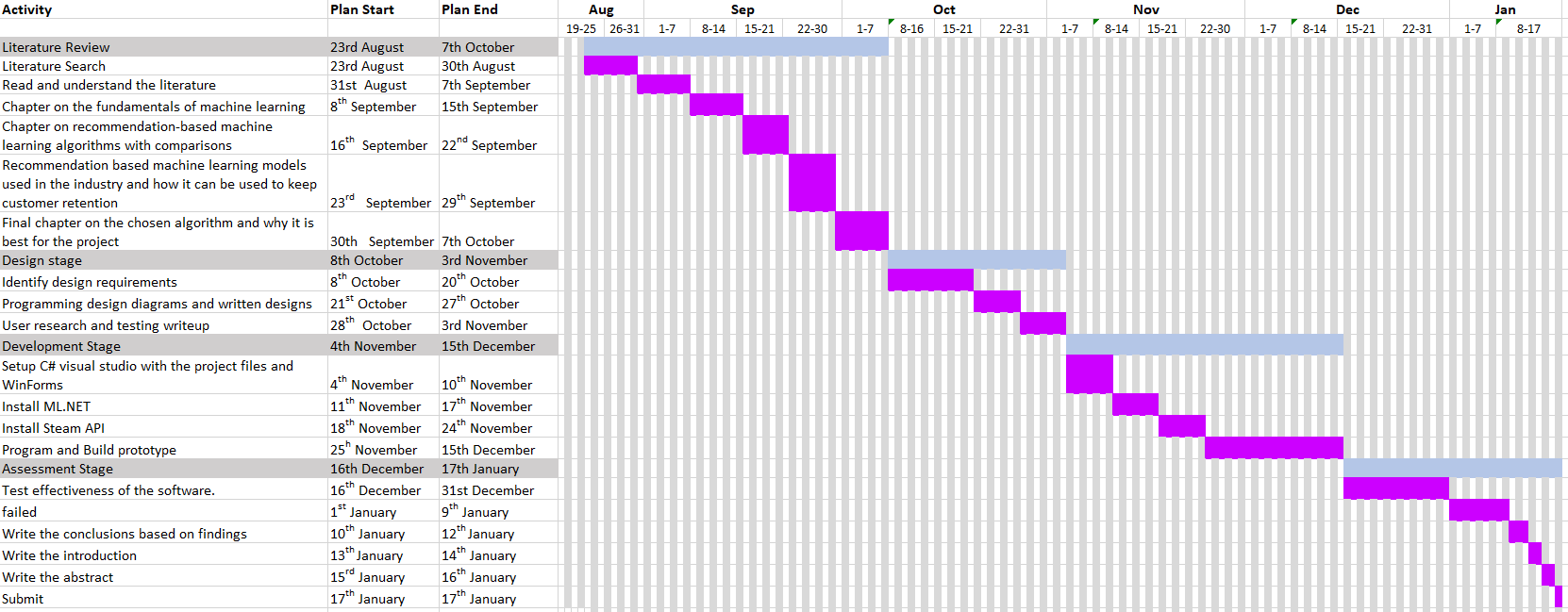
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Task | Deliverable / outcome | Resources required | Skills required | Time required |
| Literature Review | Literature Search | An extensive body of relevant literature that can be used in the literature review, encompassing journals, websites and books. | The internet and Northumbria university library. Google scholar will be used to find literature and IEEE journal will be looked at in particular. Specific literature to investigate include those to do with machine learning and recommendation algorithms. | To be able to assess quality of the literature.  Can easily find suitable literature within the chosen field. | 1 week |
| Read and understand the literature | Notes to use for review, and a greater understanding of the subject and its contents and how it can be applied to the project. | Computer, books, journals and literature that were found during the previous task. | Ability to identify suitable information from literature and retain this information.  Note taking on what has been read | 1 week |
| Chapter on the fundamentals of machine learning | A chapter within the literature review covering what machine learning is and its history. | Computer, notes created while reading through the literature and additional literature may be required | Ability to write in an academic manner.  Be able to articulate previous research in a way that is relevant to the project. | 1 week |
| Chapter on recommendation-based machine learning models with comparisons | A chapter within the literature review covering recommendation-based machine learning algorithms and comparing each one | Computer, notes created while reading through the literature and additional literature may be required | Ability to write in an academic manner.  Be able to articulate previous research in a way that is relevant to the project. | 1 week |
| Recommendation based machine learning models used in the industry and how it can be used to keep customer retention | A chapter within the literature review covering how recommendation-based machine learning can be used in the industry and how this can keep user retention in a video game distribution platform. | Computer, notes created while reading through the literature and additional literature may be required | Ability to write in an academic manner.  Be able to articulate previous research in a way that is relevant to the project. | 1 week |
| Final chapter on the chosen algorithm and why it is best for the project | A chapter within the literature review concluding which algorithm is best fitted for the project. | Computer, notes created while reading through the literature and additional literature may be required | Ability to write in an academic manner.  Be able to articulate previous research in a way that is relevant to the project. | 1 week |
| Design stage | Identify design requirements | A list of requirements that the design must encompass. This will be based on the criteria set out by the literature review. Such as the prototype must take user input to retrieve recommendations on what games they may like and how it can benefit a distribution platform the most. | The literature review needs to be completed before designs start, a computer. | Be able to identify what is required of the designs in order to complete the aim.  Be able to understand what is important and achievable over things that may not be required. | 2 weeks |
| Programming design diagrams and written designs | Designs which are both diagram and in written form. Diagrams can include UML and ERD’s for example. All additional details that cannot be expressed in diagrams will be set out in writing. | Computer, the design requirements of the previous task, diagramming software (draw.io) | Experience in diagram creation software.  Academic writing ability to articulate the designs elegantly.  Programming ability is needed in order to plan out the architecture of the software. | 1 week |
| User research and testing methodology writeup | A detailed writeup on how the research and testing will be carried out with users. Including, how participants will be chosen, questions asked and how information will be recorded in tables. | Computer, literature on how others have achieved similar research | An understanding on what exactly will need to be tested to achieve the aim.  Good planning abilities and to articulate the testing methodology informatively and clearly. | 1 week |
| Development Stage | Setup C# visual studio with the project files and WinForms | Project files set up and ready to be used so that programming can commence with C# and WinForms. | Computer, an IDE (visual studio) | Programming ability and experience with visual studio, c# and winforms. | 1 week |
| Install ML.NET | ML.NET installed into the visual studio program files, this will allow the use of machine learning | Computer, an IDE (visual studio), access to NuGet to install ML.NET | Knowledge on how to use NUget within visual studio to set up the ML.NET library, research should also go into learning how to use ML.NET | 1 week |
| Install Steam API | Steam API installed into visual studio, this will allow the retrieval of game and user data in order to complete the software | Computer, an IDE (visual studio), access to the steam api | Knowledge on how to obtain the steam API, research will need to go into learning how to use the steam API. | 1 week |
| Program and Build prototype | The final build of the prototype completed. This will be completely usable and ready for testing with users. | The designs, a computer and programming tools. Visual studio will be most likely used for programming. The language choice will be C# | Programming skills in C#, Winforms, ML.NET and the steam API. | 3 weeks |
| Assessment Stage | Test effectiveness of the software with participants | Completed tests data with the outcomes.  The criteria participants will be tested on will be in line with the aim of the project. Data retrieved from these tests will include if recommendations made were accurate, and if given such recommendations how likely you would then be to buy the game? | Computer to run the software, Participants within the target audience (steam users) | Leadership qualities in order to organise and plan the testing with participants. | 2 weeks |
| Evaluate effectiveness/ where it succeeded and failed | An evaluation of how the method succeeded and failed based on whether the participants answers to questions were positive or negative | Computer, the previous tests to evaluate in tabular format | Be able to interpret the tests and evaluate the outcome. | 1 week |
| Write the conclusions based on findings | The conclusion will go over if the objectives of the project have been sufficiently met. | All Work Completed / PC | Academic Writing | 2 days |
| Write the introduction | The Introduction which will give a good general overview of the whole project | All Work Completed / PC | Academic Writing | 2 days |
|  | Write the abstract | The Abstract which will be a concise yet comprehensive reflection of what is in the project. | All Work Completed / PC | Academic writing | 2 days |

## RESEARCH PROPOSAL Appendix B - Work Breakdown Structure

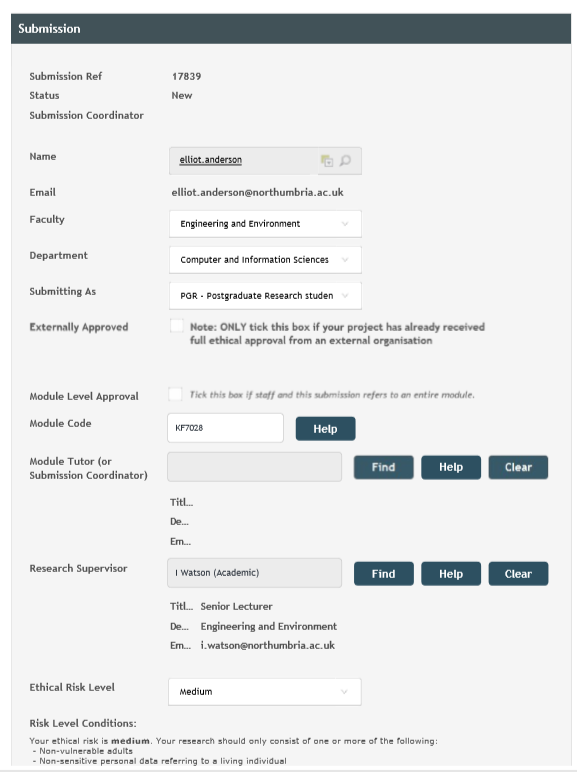
## 

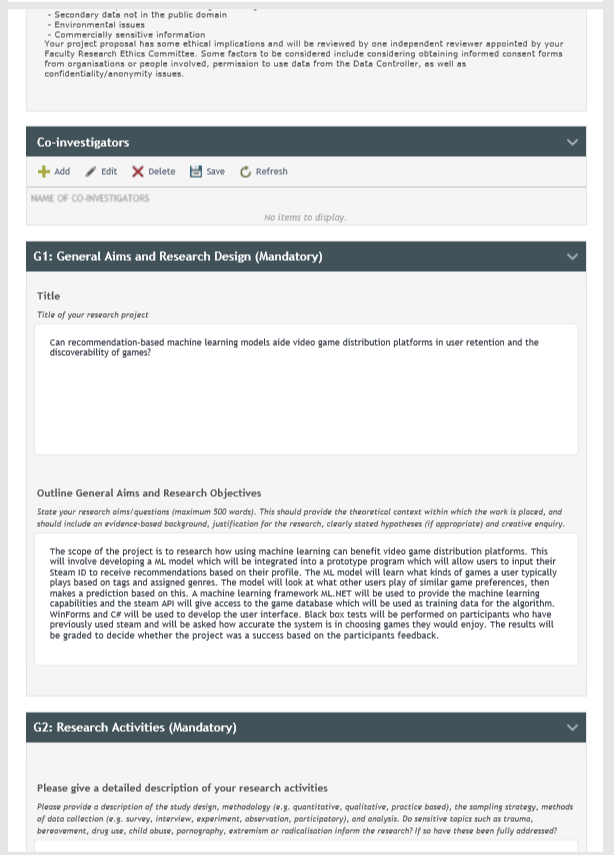
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Objective/Milestones Deliverable** | **Planned**  **start** | **Planned**  **end** | **Actual start** | **Actual end** | **Deliverable** | **Reflections** |
| Literature Review | 23rd August | 7th October |  |  | Literature Review and a clear understanding of how to approach the designing of the software. It should include a detailed explanation of machine learning and it’s applications, an explanation of recommendation-based machine learning algorithms, how it can be used in the industry for marketing purposes especially within the game industry, and finally an explanation on which algorithm will be chosen for the prototype. |  |
| Literature Search | 23rd August | 30th August |  |  | An extensive body of relevant literature that can be used in the literature review, encompassing journals, websites and books. |  |
| Read and understand the literature | 31st August | 7th September |  |  | Notes to use for review, and a greater understanding of the subject and its contents and how it can be applied to the project. |  |
| Chapter on the fundamentals of machine learning | 8th September | 15th September |  |  | A chapter within the literature review covering what machine learning is and its history. |  |
| Chapter on recommendation-based machine learning models with comparisons | 16th September | 22nd September |  |  | A chapter within the literature review covering recommendation-based machine learning models and comparing each one |  |
| Recommendation based machine learning models used in the industry and how it can be used to keep customer retention | 23rd September | 29th September |  |  | A chapter within the literature review covering how recommendation-based machine learning can be used in the industry and how this can keep user retention in a video game distribution platform. |  |
| Final chapter on the chosen algorithm and why it is best for the project | 30th September | 7th October |  |  | A chapter within the literature review concluding which algorithm is best fitted for the project. |  |
| Design stage | 8th October | 3rd November |  |  |  |  |
| Identify design requirements | 8th October | 20th October |  |  | A list of requirements that the design must encompass. This will be based on what the literature review decided would benefit a distribution platform the most. |  |
| Programming design diagrams and written designs | 21st October | 27th October |  |  | Designs which are both diagram and in written form. Diagrams can include UML and ERD’s for example. All additional details that cannot be expressed in diagrams will be set out in writing. |  |
| User research and testing methodology writeup | 28th October | 3rd November |  |  | A detailed writeup on how the research and testing will be carried out with users. Including, how participants will be chosen, questions asked and how information will be recorded in tables. |  |
| Development Stage | 4th November | 15th December |  |  |  |  |
| Setup C# visual studio with the project files and WinForms | 4th November | 10th November |  |  | Project files set up and ready to be used so that programming can commence with C# and WinForms. |  |
| Install ML.NET | 11th November | 17th November |  |  | ML.NET installed into the visual studio program files, this will allow the use of machine learning |  |
| Install Steam API | 18th November | 24th November |  |  | Steam API installed into visual studio, this will allow the retrieval of game and user data in order to complete the software |  |
| Program and Build prototype | 25h November | 15th December |  |  | The final build of the prototype completed. This will be completely usable and ready for testing with users. |  |
| Assessment Stage | 16th December | 17th January |  |  |  |  |
| Test effectiveness of the software. | 16th December | 31st December |  |  | Completed tests data with the outcomes.  The criteria participants will be tested on will be in line with the aim of the project. Data retrieved from these tests will include if recommendations made were accurate, and if given such recommendations how likely you would then be to buy the game? |  |
| Evaluate effectiveness/ where it succeeded and failed | 1st January | 9th January |  |  | An evaluation of how the method succeeded and failed based on whether the participants answers to questions were positive or negative |  |
| Write the conclusions based on findings | 10th January | 12th January |  |  | The conclusion will go over if the objectives of the project have been sufficiently met. |  |
| Write the introduction | 13th January | 14th January |  |  | The Introduction which will give a good general overview of the whole project |  |
| Write the abstract | 15rd January | 16th January |  |  | The Abstract which will be a concise yet comprehensive reflection of what is in the project. |  |
| Submit | 17th January | 17th January |  |  | The dissertation completely submitted to blackboard as well as the program files. |  |

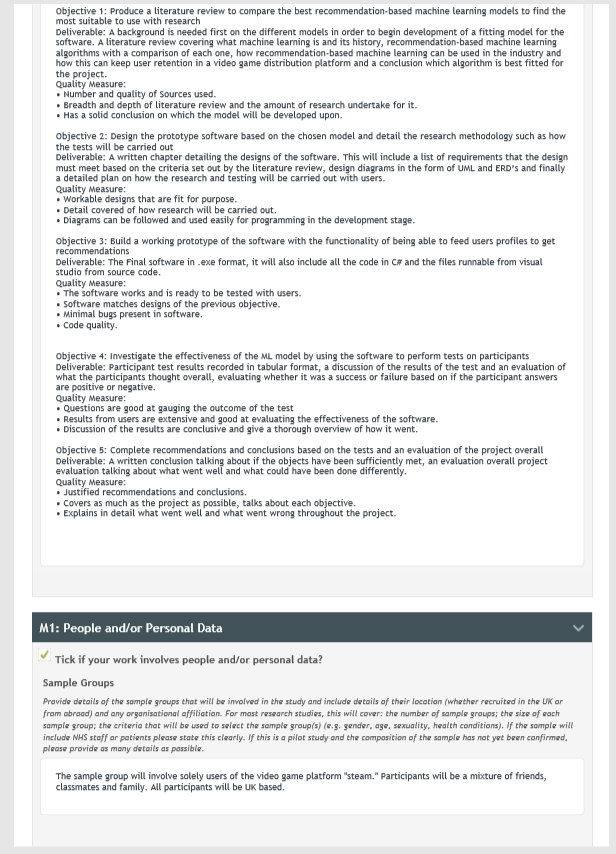
## RESEARCH PROPOSAL Appendix C - Gantt chart



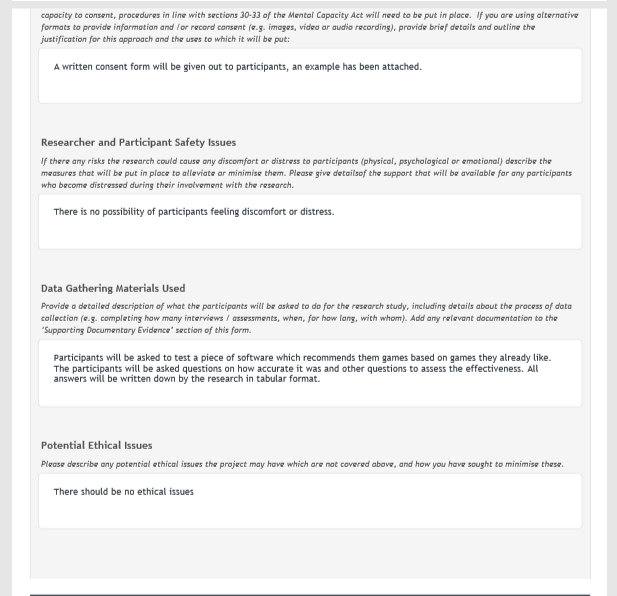
## RESEARCH PROPOSAL Appendix D - Ethics Form

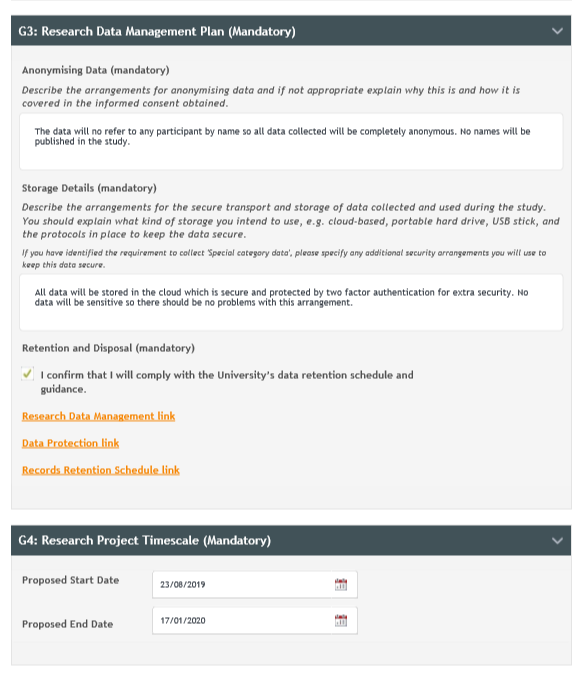


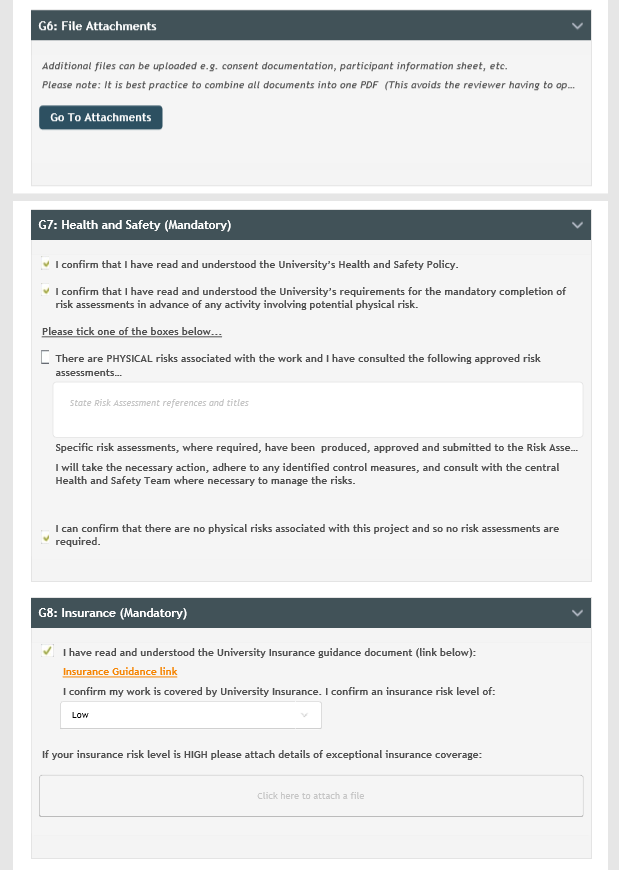


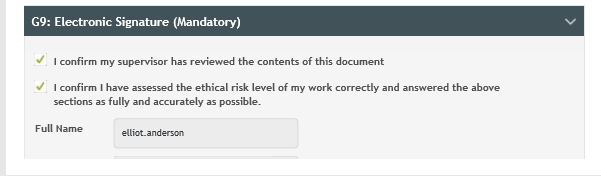












1. Ethics Approval

Ethics approval submitted and approved online in accordance with Northumbria University ethical guidance.

1. Code

