Final Year Project Report

Machine Learning for Outcome Prediction in Rainbow Six

Scott Kelly

A thesis submitted in part fulfilment of the degree of

BSc. (Hons.) in Computer Science

Supervisor: Pádraig Cunningham

UCD School of Computer Science

University College Dublin

April 2018

Project Specification

General Information:

The objective of this project is to create a machine learning (ML) system that will predict the outcome of a round in the game Tom Clancy's Rainbow Six: Siege. The game puts two teams composed of five players each against each other, on either the attacking or defending team, who will fight over a number of rounds to decide which team has won the match. After being randomly selected as attacker or defender in the first round, the teams switch roles each round until the game ends.

The match can take place on a variety of "maps", for example a building complex or something similar, and within the larger map, the defenders have a choice of objective location, ie a small place within the map where they must defend the objective.

Each player has a choice of choosing which "operator" or character they wish to play as each round. There are different operators for attacking and defending, and each operator can only be chosen by one player per team per round. They also have the choice of choosing certain equipment that they can use for the upcoming round

For the defenders to win a round, they must either eliminate the attackers or cause them to run out of time, and for the attackers to win, they must eliminate the defenders or complete the objective.

There are three main objectives; rescue the hostage, disarm the bombs, or secure the biohazard container. All objectives can be played on all maps, but vary in objective location, which could influence the attackers ability to approach the objective and complete it.

Players are ranked on an Elo-based system, where the winning team takes Elo points from the losing team, and the team members who contribute more towards the win gain more of the points.

This project is possible because Ubisoft, the publisher of the game, has released a large dataset of online matches. The data is in csv format and it has a row for each player per round per match.

Core:

The core objective is to develop an ML system that will take the available data on each round (map, objective, player operators, etc.) and learn to predict which team will win the round. This will be done in a supervised ML framework.

This system will be tested in a hold-out validation framework with a fraction of the available data held back for testing.

Advanced:

The advanced objectives would be to make the bot more accurate by considering other factors, such as the objective location, the equipment the players choose, and each player's rank, either individually or the average of each team. This phase of the project work will provide insight on effective player strategies and on what features can be derived from the data to enhance the ML performance.

Abstract

This project involves investigating if it is possible to accurately predict the outcome of a round in the video game Tom Clancy's: Rainbow Six: Siege (referred to as Siege from here on).

Siege is a competitive online tactical first person shooter, where two teams of five players battle it out over a series of rounds, defending or attacking the current objective, to see who is the victor. The players choose from a list of "operators", which are player characters all with a specific skill, as well as from a list of weapons and equipment for them to use throughout the round.

The project was completed using supervised machine learning algorithms. This will be done by using a combination of players' in-game factors such as operator choice, map choice or equipment, as well as indivdual player skill, in terms of Elo rank and account experience level.

Upon testing, the signal presented by in-game factors for prediction was weak, suggesting that individual player skill will be a larger deciding factor on accurate prediction. Multiple tests were performed on these player skills to analyse the signals for prediction and see possible improvements to the model.

Table of Contents

1 Introduction

In this report we will discuss progress made on the project, as well as relevant papers researched to gather ideas and direction on creating a machine learning framework to predict the outcome of a round in Siege. The purpose of the project is to create an accurate model that will predict the outcome of a round well based on a variety of factors.

A round of Siege consists of five attacking players versus five defending players. Players must first choose which operator they wish to play as each round (see Figure 1.1). They then battle it out over a series of rounds to determine the victor.

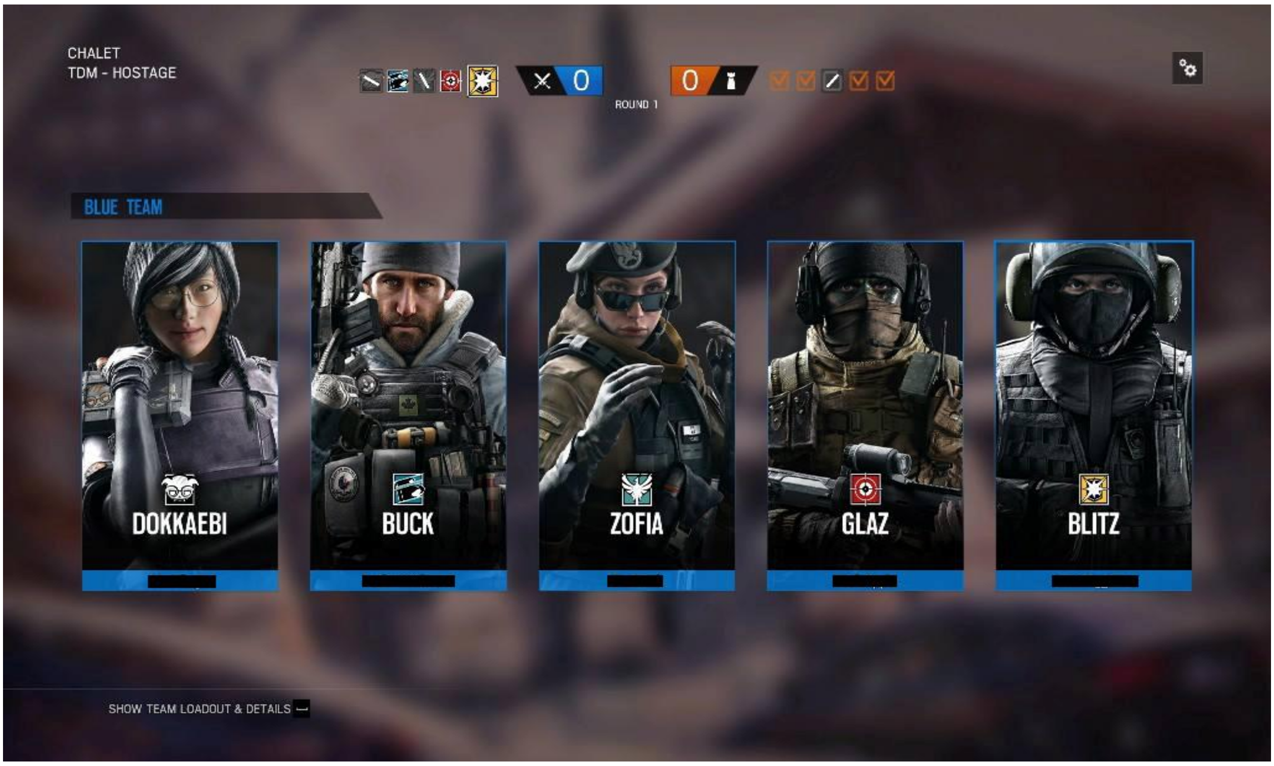

Figure 1.1: A screenshot from the game of 5 players choosing their attacking operators

The game is competitive and tactical, featuring an online Elo ranking system[1]. Therefore the most basic objective of the game, besides for fun, would be to consistently win matches, to improve and move up the ranks. While being competitive, it is important to work as a team. Each player should not only develop their own skill and develop tactics with their team members, but also compose a coordinated team composition. The composition of operators and of operator equipment and weapons can be as essential as player skill.

From a machine learning standpoint, we decided to view all of these possible signals for prediction, such as player skill, operator choice or objective location. We have used information gathered by a publicly released data dump of thousands of matches by the game's developer, Ubisoft Montreal.

After initial testing and converting the data dump into a usable format, the main difficulty that arose was the fact that there was two opposing teams; popular maching learning models work well and straight forward when it comes to a single entity, such as a single player game, but complications arise when attempting to apply these techniques to Siege, with the two opposing teams.

First, we will talk about the game, and how the game works. Afterwards, background research material will be discussed, followed by the preliminary analysis of the data and the prediction algorithms. Then, insight gained into the game and its system through this project will be discussed.

The main objective was to investigate methods of predictive algorithms for mutliplayer games, and study how ranking two or more teams against each other was done. By learning from previous studies, we planned to gain inspiration for our own implementation, possibly integrating multiple ideas to form a more accurate model for the two teams in Siege. And finally, we will discuss the possible future to the project, and what avenues it could head down for improved success.

1.1 Objectives

There are two main objectives of this project; to create a maching learning framework that will accurately predict which team will win in any given round of Siege, and through this testing, to gain insight into the game and its inner workings. It will consider factors such as player skill as well as the teams' composition in its algorithm, and should be able to consider what teams work best on different maps, or in different skill brackets.

2 Rainbow Six: Siege

This section of will cover Siege's gameplay aspects, and the core concepts of the game, that will be used for the machine learning framework. It will also cover the data dump used for the analysis, as well as the prediction task

2.1 The Game Structure

Figure 2.1 showing a sample screenshot of a player's perspective in-game

Siege is a First Person Shooter (FPS) style game. It focuses heavily on tactical gameplay and planning, with two teams of five players fighting it out to decide the victor. At the start of each round, one team is assigned as the defenders, the other as the attackers.

Each team member will choose a unique operator for that round, which is a character within the game with a certain set of skills and equipment. For example, a sample attacking operator can place breaching charges on reinforced walls to open up the enemy's defences, while a possible defending operator could place a signal jammer near a reinforced wall, stopping the attackers' breaching charges from activating.

There are specific operators that can only be chosen for attack or defence, and each player must choose a different operator. At the end of each round, the team swaps roles, and continues to play until one team has won four rounds in total over the match. Each round has a 45 second "preparation phase", where the defenders set up their defences and the attackers scout the area, followed by a 3 minute "action phase", during which the attackers are deployed, and must try and complete the objective.

Within the game, there are three game modes that you can play; bomb disposal, hostage rescue, or "secure area", which is where attackers must stay within a certain distance of a biohazard container uncontested for a period of time to win. A team can also win if they eliminate the other team within the time limit.

The game has several different maps to choose to play on, with a variety of different locations that defenders can select to defend for a current round. A map is a fictional location set somewhere on the globe, featuring building complexes and outside areas which form a virtual playground for players to attack and defend. Objective locations can be such things as different floors or rooms within the buildings. Our dataset features a total of 16 maps.

Figure 2.2 Left shows some sample maps, right shows the floor plan of the 1st floor on the Club House map

Within the game, there is a competitive Elo ranking system[1], seen in figure 2.3, similar to the system used in chess. An unranked player must play ten rounds within the competitive ranked queue before they are given a number of Elo points, are assigned an Elo rank group, which are shown in FIGURE NUMBER HERE. A player can then play after this, gaining points from the enemy team if they win, and forfeiting points if they lose. Gaining more points can allow a player to move to a higher Elo group.

Figure 2.3 shows the different categories of Elo rankings present in the game

The main goal of the ranking system is to give an indication of the individual player skill, so that the system can match two teams of relatively similar skill for a fair and, hopefully, enjoyable match. However, as the game relies heavily on team coordination and cooperation for success, it is possible that a player's Elo score could be decreased by a "worse" team, potentially introducing inaccuracy into the system.

Players also have a "clearance level" attached to their account. Clearance level of a player is the current level of experience points a player has gained over their time playing. The more a player has played, and positively contributed towards the team, the more experience points they gain, but they cannot lose experience points from playing poorly. Therefore, it is an indication of how long a player has been playing the game.

The game is available on PC, Playstation 4 (PS4) and Xbox One (XONE), and all of these elements come together to form a tactical, round-based FPS.

2.2 The Data Dump

	1	2	3	4	5	6	7	8	9	10	11	12
1	dateid	platform	gamemode	mapname	matchid	roundnumber	objectivelocation	winrole	endroundreason	roundduration	clearancelevel	skillrank
72	20170212	PS4	PvP – BOMB	PLANE	1522488121	1	MEETING ROOM / EXECUTIVE OFFICE	Defender	AttackersEliminated	192	131	Unranked
73	20170212	PS4	PvP – BOMB	PLANE	1522488121	1	MEETING ROOM / EXECUTIVE OFFICE	Defender	AttackersEliminated	192	94	Gold
74	20170212	PS4	PvP – BOMB	PLANE	1522488121	1	MEETING ROOM / EXECUTIVE OFFICE	Defender	AttackersEliminated	192	137	Gold
75	20170212	PS4	PvP – BOMB	PLANE	1522488121	3	STAFF SECTION / EXECUTIVE BEDROOM	Defender	AttackersEliminated	183	137	Gold
76	20170212	PS4	PvP – BOMB	PLANE	1522488121	2	CARGO HOLD / LUGGAGE HOLD	Attacker	DefendersEliminated	113	137	Gold
77	20170212	PS4	PvP – BOMB	PLANE	1522488121	1	MEETING ROOM / EXECUTIVE OFFICE	Defender	AttackersEliminated	192	85	Platinum
78	20170212	PS4	PvP – BOMB	PLANE	1522488121	3	STAFF SECTION / EXECUTIVE BEDROOM	Defender	AttackersEliminated	183	120	Gold
79	20170212	PS4	PvP – BOMB	PLANE	1522488121	4	CARGO HOLD / LUGGAGE HOLD	Attacker	DefendersEliminated	120	176	Gold
80	20170212	PS4	PvP – BOMB	PLANE	1522488121	4	CARGO HOLD / LUGGAGE HOLD	Attacker	DefendersEliminated	120	132	Platinum
81	20170212	PS4	PvP – BOMB	PLANE	1522488121	1	MEETING ROOM / EXECUTIVE OFFICE	Defender	AttackersEliminated	192	120	Gold
82	20170212	PS4	PvP – BOMB	PLANE	1522488121	4	CARGO HOLD / LUGGAGE HOLD	Attacker	DefendersEliminated	120	85	Platinum
83	20170212	PS4	PvP – BOMB	PLANE	1522488121	4	CARGO HOLD / LUGGAGE HOLD	Attacker	DefendersEliminated	120	120	Gold
84	20170212	PS4	PvP – BOMB	PLANE	1522488121	2	CARGO HOLD / LUGGAGE HOLD	Attacker	DefendersEliminated	113	132	Platinum
85	20170212	PS4	PvP – BOMB	PLANE	1522488121	4	CARGO HOLD / LUGGAGE HOLD	Attacker	DefendersEliminated	120	94	Gold
86	20170212	PS4	PvP – BOMB	PLANE	1522488121	2	CARGO HOLD / LUGGAGE HOLD	Attacker	DefendersEliminated	113	176	Gold
87	20170212	PS4	PvP – BOMB	PLANE	1522488121	3	STAFF SECTION / EXECUTIVE BEDROOM	Defender	AttackersEliminated	183	94	Gold
88	20170212	PS4	PvP – BOMB	PLANE	1522488121	3	STAFF SECTION / EXECUTIVE BEDROOM	Defender	AttackersEliminated	183	85	Platinum
89	20170212	PS4	PvP – BOMB	PLANE	1522488121	1	MEETING ROOM / EXECUTIVE OFFICE	Defender	AttackersEliminated	192	176	Gold
90	20170212	PS4	PvP – BOMB	PLANE	1522488121	2	CARGO HOLD / LUGGAGE HOLD	Attacker	DefendersEliminated	113	151	Gold
91	20170212	PS4	PvP – BOMB	PLANE	1522488121	1	MEETING ROOM / EXECUTIVE OFFICE	Defender	AttackersEliminated	192	153	Gold
92	20170212	PS4	PvP – BOMB	PLANE	1522488121	2	CARGO HOLD / LUGGAGE HOLD	Attacker	DefendersEliminated	113	163	Gold
93	20170212	PS4	PvP – BOMB	PLANE	1522488121	4	CARGO HOLD / LUGGAGE HOLD	Attacker	DefendersEliminated	120	151	Gold
94	20170212	PS4	PvP – BOMB	PLANE	1522488121	3	STAFF SECTION / EXECUTIVE BEDROOM	Defender	AttackersEliminated	183	151	Gold
95	20170212	PS4	PvP – BOMB	PLANE	1522488121	4	CARGO HOLD / LUGGAGE HOLD	Attacker	DefendersEliminated	120	153	Gold
96	20170212	PS4	PvP – BOMB	KANAL	1522514641	3	SERVER ROOM / CONTROL ROOM	Attacker	DefendersEliminated	231	81	Gold

Figure 2.1: the data format inside of the data dump

On the 27th of June 2017, Siege's developer and publisher Ubisoft released their "Data Peek" initiative[2] . In this, they released a data dump to be used by the public, featuring a snippet of their own data that they gather on a daily basis. This data dump is the data that we have used for this project.

This 20.8 GB csv file, holds data from approximately 500,000 rounds over multiple days within the months of February, March and April of 2017. A sample snippet from this file is shown in figure 2.1 above. The data file holds data for each round of multiple matches, with each row representing one player. They are organised by a match ID, and a round number.

For each round, the data holds such things the current objective, operator choice for each player, their equipment, what map and objective location the round took place on, the players' Elo skill ranking, as well as the date the round took place on, and what gaming platform the players were using. It also denotes which team each player was on, and which team won that specific round.

2.3 The Prediction Task

Using the data released by Ubisoft, we planned to create a supervised machine learning framework, with the goal of predicting which team would win a particular round. We planned to examine and analyse the data to investigate possible signals that could help predict if a specific team composition would be more likely to win.

We would examine the different factors within the data, such as map, operator choice or Elo rankings as examples, to see if these features would lead to a strong signal for a team's likeliness to win. We would start off with filtering and formatting the data for feature engineering purposes, and then lead onto the machine learning prediction, experimenting with different machine learning models to see the performance of them. As the data dump is quite large, we were able to use it for both training and testing purposes without overlap.

To make the task simpler, rather than seeing which team would win, it was performed from the view of the defending team, whether they would lose or win. Overall, the main task and goal of the project was to make a relatively accurate model for round prediction for Siege. Features would be added to try and improve the model while not oversaturating it with relatively useless features.

3 Background Research

This section of the report will outline background research completed for the purposes of this project. The main aim of the research was to investigate past machine learning experiments which involved game playing for multiple teams. Viewing previous techniques of ranking and evaluating multiple teams for games, no matter the type, was essential for the continuation of this project.

3.1 Greyhound Racing

A paper from 1994[3] conducted an experiment on Greyhound racing at a track in Arizona. It's main objective was to see if machine learning could be used to more accurately predict the winner of a greyhound race than three human track experts. They planned to predict the winners by using this past information in two frameworks; a decision tree building algorithm (ID3) and a neural network learning algorithm (backpropogation).

3.1.1 Initial Planning

For the purposes of this study, they had details of each of the dogs' past races and their performance throughout these races. They also considered the dogs' performance in its last seven races, and what position the dogs were in on each lap of the races.

The initial stage of the study was to reduce the problem's complexity by pruning the problem space; they did this by selecting a set of performance attributes based on the opinions of frequent bettors. They selected features such as win percentage, finish place average, and fastest time. Once that was completed, the next step was data collection and entry based on these performance features.

3.1.2 ID3 Decision Tree

In their testing they adopted a divide-and-conquer strategy for clasification. Each dog could be classified as a winner or a loser in their module, allowing for a binary tree, and they developed the tree such that it would use the entropy of each feature to decide which greyhound it would predict to win.

Due to the greyhounds' features, they would be classified into groups, and the first five attributes in the tree were useful for deciding first place winners. They also developed a ternary tree with continuous entropy values, which led to a simplified tree, but the same efficiency in runtime.

Out of 100 games, the tree did not select a winner in 26 games, correctly selected the winner 34 times and incorrectly selected the winner 50 times. Therefore it had a 40% success rate on picking only the first place winner. This may seem low, but compared to the three human experts, it performed better, with the three experts having a success rate of 19%, 17% and 18% respectively.

The tree performed better than the humans, and their calculations that betting $2.00 on the tree's predicted winners lead to a profit of $69.20.

3.1.3 Backpropogation Neural Network

The second strategy for predicting the first place winner was a neural network using backpropogation. The input layer of the neural network was determined by the number of independent variables, and consisted of the various dog racing attributes that could affect if the dog would win or lose. The neural network correctly selected the winner in 20% of the races, with a payoff of $124.80.

The idea of backpropogation, to calculate the error contribution of each node, along with the idea of the performance attributes may lead to a good strategy for this project.

3.2 Matchmaking System in Ghost Recon Online

In a paper published in 2012[4], three researchers investigated a better way to rank players in the online multiplayer game Ghost Recon Online. The objective of the study is different to this project in terms of its goal, but the work done in it may be applicable to this project.

The paper focuses on improving player statisfaction in online video games; It can be easy to tailor an experience in a single player game, but once other players are added into the equation, it can become difficult. This is because player behaviour and interactions are hard to predict. They argue that the current ranking system in games, reducing a player's skill down to a number, can lead to player dissatisfaction in an online match where they feel they were unfairly matched against the enemy team.

This study aims to rectify that by using machine learning, specifically neural networks, to more accurately evaluate the players and teams, so that a more balanced (ie, a 50% chance of either team to win) match can be made. Although they do make the point that a balanced game does not necessarily equate to a fun or entertaining game, it is more likely that the players will enjoy themselves when they are not hopelessly out-skilled.

3.2.1 The Neural Network Approach

In their system, they have access to historical data of the players, to try and assess a player's skill off of this information. As well as this, they also take the current map that the match will be played on, and the game mode.

Labelling the teams as team A and B, the neural network was trained whose output is the probability that team A wins over team B, with their desired output to be 50%. The networks input are the player profiles, which are then transformed using a vector of player features, which are a summary of the profile, holding a skill estimate of the player's skill in various areas of the game.

While their ultimate goal is different, following their framework to predict the probability that one team will win over the other would fit into this project.

3.3 Adapting Above Frameworks For the Project

The aforementioned machine learning techniques can be modified to fit the needs of the current project. The main problem is the past information on the players; the data dump we currently use is anonymous, therefore we cannot base our algorithms off of past data of users.

We plan to experiment with different ways of representing the data so that it can fit these frameworks. While we cannot get historical data for a specific player, we can try to emulate it by considering that player's skill ranking and current account experience level. If we represent the information gathered by a player placeholder that has the same basic statistics as any possible player, we may be able to develop a model which performs well.

4 Preliminary Analysis

This section of the report will cover the preliminary tests and analysis that were performed prior to the machine learning prediction. It will discuss the feature engineering performed as well as the tests to investigate possible signals in the data.

4.1 Initial Processing

Before beginning testing, the data had to put into a useful format. The data was unorganised and messy, but due to the existence of a matchID and round number, a simple script reorganising the file fixed the problem. It was then easier to write scripts to analyse the data from this point as it was in a uniform format for each round.

From here, we noticed that there was a large amount of rounds that were not complete, ie they did not have a full ten player lobby. This can happen with online multiplayer games, with people leaving due to frustration over losing rounds, or just having an unstable internet connection, and subsequently getting "kicked" from the lobby.

We decided that these rounds were not useful for our prediction task, and would in fact hinder the machine learning model; a round which is not full would naturally lean more towards the team with more players winning, and would not offer much to overall model. Therefore, the incomplete rounds were removed, and the data in the below table (Figure 4.1) shows the results of the removal.

Platform	**Percentage of Rounds Stripped (Approximate)**
Total	45%
PC	30%
Consoles (PS4 and Xbox One)	50%

Figure 4.1 Shows the overall percentage of rounds removed throughout the entire data set, and for the two main platform skews

It was surprising that so many rounds had to be removed, but it left enough data to work with that it was better to remove these incomplete rounds to avoid unnecessary complication in the framework.

4.2 Preliminary Testing

For our initial analysis period, we performed various tests, including "straw man" testing on the data to investigate possible signals. A straw man test is a very simple test which involves a basic scenario of checking a feature's value on one team to another, and predicting the team with the larger value would win. The objective of the straw men test is to see if there is any accuracy in predicting solely from that feature alone.

4.2.1 Feature Engineering

Before performing straw man tests, we had to extract features from the data to perform the tests on. To this end, a multitude of simple scripts were written, to extract data in a useable format, but first, we had to consider how much data was truly needed for a good representation of all the data. If too much data was used, it would add little benefit to the tests while costing processing time.

To cut down on the data processed, we decided to choose data from one day only, and one map only. Later testing showed that the results from different days and maps for the features were incredibly similar, so adding them to the data processed would not yield much gain.

Firstly, we aimed for features in the data such as the win rates of the operators (ie, what percentage of winning teams featured that operator), the win rates of operator tuples, and the total combined clearance level of team. The scripts would compile a round as a numeric value for each of these features, such as the combined total win rates for each team's operator composition, and the combined clearance level.

A few problems arose with the operator tuples; within the data set, some operator tuples appeared very little in comparison to others, so much so that their win rate was heavily skewed in one direction or the other. To counteract this problem, we set a cut off point that, for a touple to be considered, it had to have appeared at least 50 times. This cut down on potential skews in the data, hopefully improving overall accuracy at the end.

Another problem with the tuples was that the most successful tuples contained the operators that were also the most successful in isolation, or at least near the top of the list. Our initial hypothesis was that some operators that preformed relatively worse than the best, would perform much better in combination with another operator, due to the team-oriented style of the game.

However, it seems the best operators still perform the best within a tuple, suggesting that the analysis of operators in isolation and in tuples simultaneously would have strong overlap in the data, and would not be of benefit to the system. Therefore we left out tuples in our analysis.

4.2.2 Testing Results

This subsection will discuss the results of the preliminary testing performed, and the implications of the data and possible signals found in the data.

4.2.2.1 Operator Win Rates

Firstly, we performed a straw man test on the combined operator win rates on each team. The win rate of an operator is a decimal number between 0 and 1, representing the percentage of victorious rounds where the operator was chosen by a player. Each team's total win rate is then the total of all the operator win rates, and this test predicted that the team with the larger win rate would be more likely to win.

Over a large amount of rounds, the prediction accuracy of the straw man is relatively poor. With two teams, you can expect that random guessing would give 50% accuracy, and over the whole dataset, there was an accuracy of 53%.

We needed to investigate further to see if the accuracy gain was not just random. To do this, the data was ordered as to rank the rounds by disparity between the two team's win rates. The larger the difference between the teams, the higher it was placed. The same test was performed, taking certain sets of the top rounds to investigate preformance. This data is shown in the figure 4.2.

Top Number of Rounds – Operator Win Rate	Performance of the Straw Man (%)
100	61.0%
500	58.4%
1000	58.04%
5,000	58.04%
10,000	57.19%

Figure 4.2 shows the performance of the straw man for operator win rates

When the difference between teams is larger, there is some indication as to which team would be more likely to win, therefore showing operator choice offers some signal, despite it being weak.

4.2.2.2 Clearance Level Tests

The next test to perform was on player's clearance level, or their experience level. The hypothesis was that a player with a higher clearance level, which has played the game for longer, would be more skillfull, and a team with a higher combined clearance level would be more likely to win.

The test was performed in a similar manner to the operator test; each team's total combined clearance level was compared, and the test predicted that the team with the larger overall value would be more likely to win. Similarly ranking by difference between teams as in the last test, the following results were gathered (Figure 4.3).

Top Number of Rounds – Clearance Level	Performance of the Straw Man (%)
100	55.0%
500	52.8%
1000	50.6%
5,000	51.16%
10,000	52.56%

Figure 4.3 shows the performance of the straw man test for team's clearance level

The performance of the test on this feature alone is poor. It is a very weak signal and would not be much better than random guessing. This can suggest that a player who has played for a long time does not necessarily perform better in the game.

4.2.2.3 Elo Ranking Testing

For testing Elo, we decided to categorise and rank the different elo segments for the players. We labelled each group with an integer value, ranging from Diamond on the top with the value 7, and unranked and the bottom with 0. With these values, we wanted to test two things; total overall Elo, as well as the highest Elo value on each team.

The first test was combining each player's categorical Elo ranking together and comparing teams, as in the previous two tests. Another hypothesis arose, however, that the best player on each team would actually have a larger impact on success than the whole team combined. This arose from the possibility that if the top player is much more skilled than the enemy team, they can dominate on their own, without teammates help. From testing these ideas we got the following results (Figure 4.4).

Top Number of Rounds – Categorical Elo	**Performance of the Straw Man (%)**	**Top Number of Rounds – Highest Elo**	**Performance of the Straw Man (%)**
100	60.0%	100	64.0%
500	56.99%	500	63.80%
1000	55.60%	1000	61.40%
5,000	56.80%	5,000	61.00%
10,000	60.14%	10,000	Not enough data

Figure 4.4 shows the performance of the categorical Elo straw man (left) and the highest categorical Elo straw man (right)

To remove noise in the data and improve accuracy, rounds with unranked players who had not completed their placement matches were ignored, and only rounds with a full set of ranked players on both teams were considered. Rounds which led to a tie in the two tests were also ignored. This led to a smaller data set but one that could be predicted much more accurately and consistently.

This outcome suggests that Elo could be a large signal for indicating which team could win. This may be self explanatory as the Elo is an indication of player skill, but the idea of one member dominating the rest may yield more promising results.

4.2.2.4 Objective Location Analysis

The final preliminary test performed involved the various objective locations that defenders could choose for a specific round. Within a map, there are multiple locations, some specific to a game mode, and this test aimed to analyse the most successful objective locations for the defending team.

As we have decided to analyse data from one map only, the data discussed will cover the locations in the map, "Club House".

Objective Location	**Defender Win Rate (%)**
Bedroom	44.3
Church/ Arsenal Room	53.2
Bar Backstore	44.7
Church	62.4
Club House	49.2
Cash Room	42.0
Strip Club	46.0
Arsenal Room	54.4
Gym/ Bedroom	43.5
CCTV Room/ Cash Room	34.6
Bar/ Stock Room	45.2
Garage	47.1

Figure 4.5 shows the defending team's win rates on all objectives across all game modes on the map Club House.

The win rates generally float around 50%, suggesting that the objective location has no major impact, but some locations are largely skewed in one direction or the other. In these skewed locations, we could gain a valuable signal for predicting the outcome of a round.

The above tests have offered possible signals that could be used in our machine learning model to produce and accurate system for predicting the winner of a round.

5 Evaluation

This section of the report will be separated into three sections; prediction and insight, with a preface about the implementation of the model. In a machine learning model, there are two main objectives for the project; to create an accurate model, and to gain insight into the system that you are developing the model upon. These two sections will discuss the success of these objectives in this project.

Prediction will cover the development and results of the machine learning model created using the data set and features created in the preliminary evaluation. After the prediction, the insights into the data, the game and its systems gained through the analysis tasks will be discussed.

5.1 Implementation

Before discussing the results of the machine learning model, we will briefly discuss the implementation of the scripts created to gather and format the data. Ignoring the scripts that create the straw man tests, scripts created in java were used to format the data for the program Weka.

Weka is an open source machine learning software which can perform machine learning analysis using a variety of models on data formatted in .arff format. We decided that creating the models for the use of our data was outside the scope of this project, and decided that formatting our data for use in Weka was more appropriate.

Due to complications of streaming the very large data file into our java project, a C++ script was written to grab specific segments of the data, such as data for one map only, etc. This data was then passed to the java project, and using the classes, formatted this data into .arff format for Weka to use. Once the data could be used by Weka, more analysis could be performed using Weka's tools to gain further insight into the features.

5.2 Prediction

To aim to create an accurate model for the system, the features chosen were the aspects of the game that indicated the largest signals for a winning team. To make the prediction outcome easier to analyse and understand, the prediction is done from the defender's point of view. Therefore the outcome is labelled as a defending win or loss, rather than which team won.

These features were chosen during the preliminary analysis, and were such things as operator win rate delta, Elo delta and objective location. Elo delta is simply the difference between the categorical value of the defender team and the attacker team. If the value is negative, the attacking team has a larger Elo value than the defenders. Win rate delta works similar, except using operator win rates rather than elo. In the data, we used three Elo features, e1delta, e2delta, and e5delta. To explain these, refer to the following example; (FIGURE)

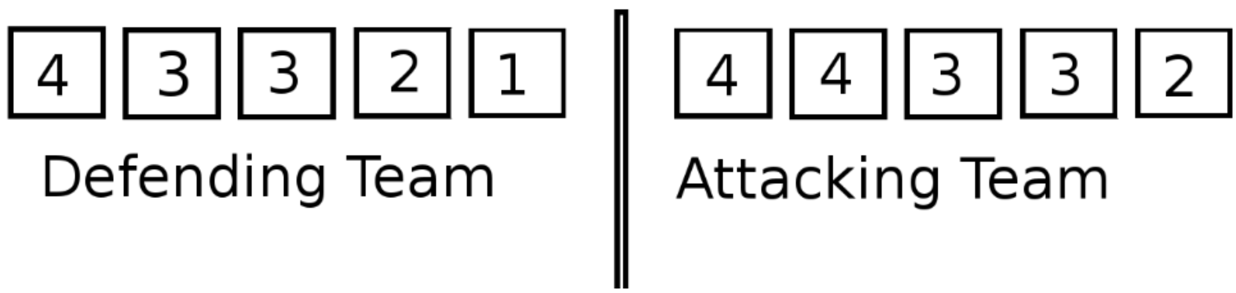

Figure 5.1 shows a sample team represented by their categorical Elo values

The example shown in FIGURE shows two team represented by their categorical Elo value. The larger the value, the higher they are on the Elo ranking scale. From this represenation, we have created three features to examine:

- E1delta represents the difference between the defender's strongest player and the attackers strongest player. In this example, this works out as 4 – 4, which is 0, depicting no difference in Elo category.
- E2delta represents the difference between the top two players on either team. The example labels the top two defenders with 4 and 3, and the attackers both with 4. E2delta, in this example, is then given the value of (4+3) – (4+4) = -1, indicating the attackers are one Elo ranking above the defenders.
- E5delta encompasses the whole team on either side. It takes into account all members in the calculation, so for the example, e5delta = (4+3+3+2+1) – (4+4+3+3+2) = -3, showing that the attacking team is stronger than the defending team in terms of Elo ranking.

Before prediction testing, any rounds with unranked players, or with either Elo or win rate delta values that equalled zero were removed, due to the noise they present in the data. While this does decrease the data set size, the increase in accuracy is a good trade off.

5.2.1 Prediction Accuracy of Models

This section will briefly discuss the different machine learning models that were experimented with, and their accuracy with each feature, and then collectively. This task was performed to understand the impact of the features for overall predictability of the model.

Each of these tests were performed using Weka, with 10-fold cross validation, on data for the Club House map. The tests were performed on a few of the other maps, and similar results were produced, therefore we decided to just choose on map in isolation. The following table shows the results (Figure 5.2).

Model/Feature	**E1delta**	**E2delta**	**E5delta**	**WinRateDelta**	**Obj Location**	**Total**
Naïve Bayes	58.81%	58.00%	59.93%	52.68%	54.85%	59.64%
kNN (k = 5)	58.81%	59.02%	59.93%	53.00%	54.87%	56.88%
J48	58.81%	59.03%	59.93%	52.88%	54.87%	59.60%
Log Regression	58.81%	58.94%	59.93%	53.56%	53.81%	60.08%

Figure 5.2 shows the percentage of correctly classified instances of the data set for the Club House map. These results were gained from using Weka

The different models produce similar results, some being exactly the same number of correctly classified instances within Weka. This could be due to the format of the data and the short ranges that some of the features have. Overall, it seems that one model does not dominate heavily over another.

This data suggests that the Elo rankings of the players seems to be the most dominant feature for predicting if defenders will win, while operator win rate's contribution to prediction is relatively low. This is somewhat surprising, and the possible insight gained from this testing will be discussed in the next section of the report.

Before discussing insight, we explored logistic regression model more to give another indication of the impact each feature has on the overall prediction of the model, and ways to improve the accuracy of the model.

5.2.2 Logistic Regression and Improving the Model

When performing analysis with logistic regression, the model provides you with the cooefficients and the odds ratios of the features within the model created. These two can indicate the impact of a feature on the models outcome for a particular value which, in our case, is the defenders winning.

One must be careful, however, when taking these values at face value; they are not always truly correct, and can indicate things that may not be true. We decided to take them into account with our above analysis in section 5.2.1 as they could still be beneficial in this case. In combination with our other analysis, it could yield good insight.

Below, the coefficients and odds ratios taken from the test of all features from 5.2.1 are shown(Figure 5.3).

Coefficients...

Variable	Class W
e1delta	-0.0127
e2delta	0.1086
e5delta	0.1458
winratedelta	0.045
objlocation=BEDROOM	-0.1296
objlocation=CHURCH_/_ARSENAL_ROOM	0.1898
objlocation=BAR_BACKSTORE	-0.3572
objlocation=CHURCH	0.5742
objlocation=CASH_ROOM	-0.3514
objlocation=STRIP_CLUB	0.0809
objlocation=ARSENAL_ROOM	0.2345
objlocation=GYM_/_BEDROOM	-0.1452
objlocation=CCTV_ROOM_/_CASH_ROOM	-0.6066
objlocation=BAR_/_STOCK_ROOM	-0.3726
objlocation=GARAGE	-0.0674
Intercept	0.2907

Odds Ratios...

Variable	Class W
e1delta	0.9874
e2delta	1.1147
e5delta	1.1569
winratedelta	1.0461
objlocation=BEDROOM	0.8784
objlocation=CHURCH_/_ARSENAL_ROOM	1.209
objlocation=BAR_BACKSTORE	0.6996
objlocation=CHURCH	1.7757
objlocation=CASH_ROOM	0.7037
objlocation=STRIP_CLUB	1.0843
objlocation=ARSENAL_ROOM	1.2643
objlocation=GYM_/_BEDROOM	0.8648
objlocation=CCTV_ROOM_/_CASH_ROOM	0.5452
objlocation=BAR_/_STOCK_ROOM	0.6889
objlocation=GARAGE	0.9349

Figure 5.3 shows the Weka coefficients and odds ratios output for Logistic Regression

From the above odds ratios, we can denote that Elo rankings do have a larger impact on success than the operator choice. The larger the value of the odds ratio, the more impact it has on the outcome being a win. Some objective locations contribute more, these being the objectives which defenders win more frequently, but in an average sense the location does not contribute as much as Elo.

The coefficients can indicate contribution towards a defending win also. The values cannot be judged by the size as much, due to the fact that a feature which has a larger range of values would have a larger coefficient. One thing that can be deduced is the sign of the coefficient; a negative value indicates a negative contribution towards a win, while a positive value indicates the opposite.

Once e2delta and e5delta were introduced, e1delta's impact decreased. This is most likely caused by overfitting, the idea that e1delta's data was already stored in the other Elo features. This along with e5delta's better performance suggests that our hypothesis that the strongest player carries the team may be incorrect.

Due to the data suggesting e5delta to be the most important, we decided to investigate this feature to try and improve the model. We followed suit on our previous steps, which was to observe the increased performance by ignoring rounds with Elo scores that were too similar. We experimented by removing different ranges to see the increased performance and how much the data set decreased in size. The following data was produced; (Figure 5.4)

Number of Rounds	9837	4970	2294	1165	463
E5delta range removed	[0]	[-1,1]	[-2,2]	[-3,3]	[-4,4]
Prediction Accuracy (Log Regression)	60.19%	63.31%	65.99%	68.84%	73%

Figure 5.4 shows the results of different ranges removed from e5delta, and using logistic regression within Weka to see improvements in accuracy

The rounds with more extreme differences in Elo, either in favour or against the defenders, are more easier to predict, even if there is much less rounds. The model can be accurate if only predicting upon the rounds where its confidence level is higher, which stops it from being accurate over a large set of possible rounds, but shows that there are signals within the game system and its data.

5.3 Insights into Rainbow Six: Siege

The second objective to a machine learning project, is to gain insight into the system that you are performing tests upon. In our case, throughout testing we have gained insight into the game's structure, how teams succeed, and also how the developers may balance aspects of the game for player enjoyment and user retention purposes.

5.3.1 The Strongest Player vs The Team

An initial hypothesis asked if the strongest player on the team had more impact on the success of the round than the rest of the members. If one player was significantly better at the game, could they overcome the teamwork of the other team, and win?

This is very rarely the case; viewing the data, there are rounds that exist with a large gap between one player and the rest of their team, but more often than not, Ubisoft's matchmaking algorithm will balance out the disparity. To ensure player enjoyment, the algorithm seems to try and get the total Elo scores on each team to be relatively the same. Therefore if one player has a much higher Elo than their teammates, the enemy would balance out that by, more often than not, being in the middle of the gap, in terms of Elo.

The data and testing shows that taking the team as a whole to be stronger in terms of Elo is a more predictive factor. This shows that actual team work can lead to more success rather than individiual player skill. In very fringe cases a single player can dominate, but overall the team is more important.

5.3.2 Operator Popularity is not Reflected in the Win Rate

Within the list of chosen operators, there are some that are more popular than others. This could come down to a variety of factors, such as enjoyment factor, perceived success of that operator from the community, and so on. As shown in figure 5.5, some operators are much more popular, but all operator's win rates are relatively the same, around 50%.

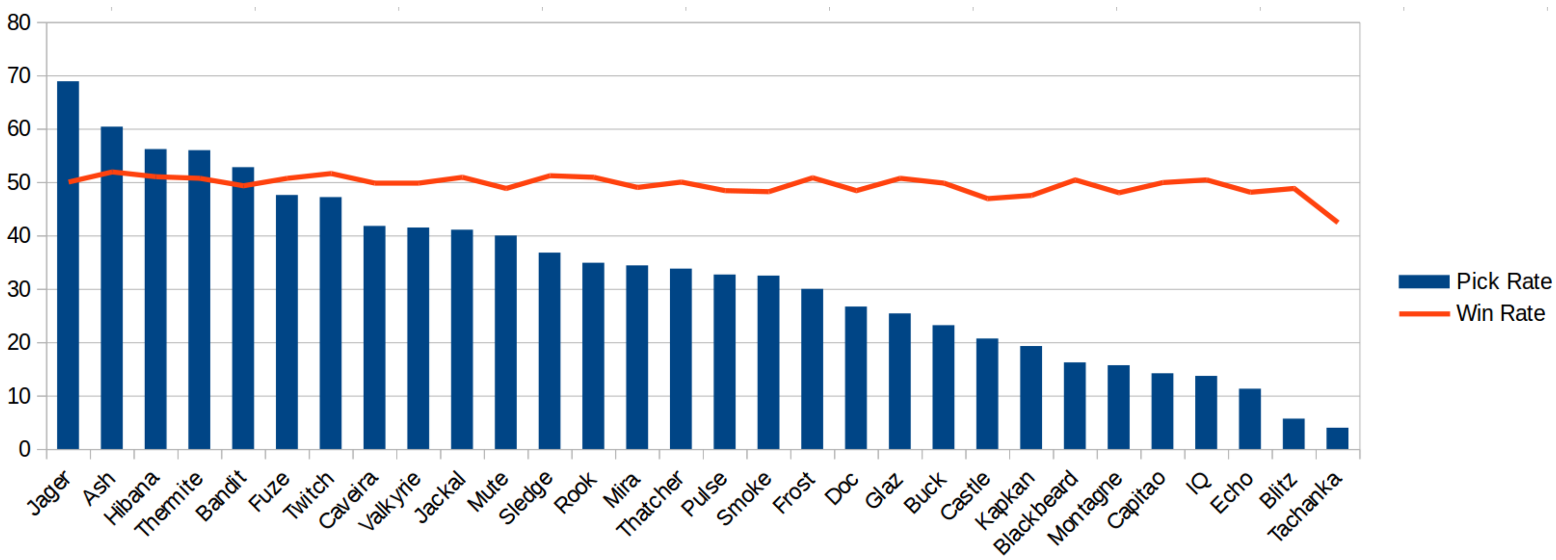

Figure 5.5 shows the operator's pick rate and win rates throughout the data set, ordered by pick rate in descending order

For example, Jager, the most picked operator showing up in 68% of all games, has an overall win rate of 50.1%. Blitz, the second least picked operator, being chosen in only 5% of rounds, has a win rate of 48.9%. While his win rate is still less than Jager's, his pick rate being so much smaller shows that there is no real correlation between pick rate and win rate.

This leads onto the next insight, and the possible implications of this will be discussed in the next section.

5.3.3 Operator Choice vs Elo Ranking

In our testing, it reveals that operator choice had little impact on a team's success. This was surprising as, within the game, choosing the right balance of operators is an important part of teamwork. The game features many synergies between operators, and facilitates certain strategies to work effectively.

The hypothesis was that a disorganised team without a planned operator selection would be more likely to lose than a planned composition. However, due to the data, and the larger impact of Elo over the operators, it suggests that a good set of players would be likely to perform well without much consideration to the operator selection, and a bad team would be likely to perform poorly regardless of a good and balanced composition of operators.

This surprise may be because of the developers focusing on the overall experience of the game and player enjoyment. In the next section, this will be discussed.

5.3.4 Game Balancing and Player Retention

Throughout testing, a few key aspects of the game seemed to give very little signal as to who the winner of a particular round could be, namely operator win rates, and objective location win rates. The majority of these features float around 50% success rate. This could give insight into the process of game balancing that Ubisoft employs to improve the experience for players and boost player retention.

Siege is primarily an online multiplayer game. For a game that focuses on playing with others, player retention can be very important; if there is no one online to play with, new players will not want to purchase a "dead" game. As well as this, keeping players playing for longer could persuade more players to purchase microtransactions or in-game purchases, increasing the revenue for Ubisoft.

With an ever evolving game that adds new content like operators and maps regularly, it can be important not to introduce something deemed "unbalanced" that may upset the player base. A map which may lean to one team winning over another (ie defender or attacker sided) or an operator which is so superior in their ability that a player with less skill could consistently defeat a player of higher skill, could lead to player frustration and, in turn, could lead to a decrease in regular players.

In this light, the developers may work hard and rigorously test their new additions to avoid this. This would then lead to a 50% win rate on maps and operators overall, with player skill being the dominating factor for a win. The results of the data suggests that they have somewhat succeeded at this, with no operators seeming to be unbalanced, or one team having a major advantage over the other on any map.

However, this has lead to less strong signals in the data. Ensuring no objective location or operator have no inherit advantages decreases the possibility to predict more accurately off of potential unbalanced features of the game.

6 Conclusion

In conclusion, the data presented was more beneficial from a data analytics standpoint and less from a machine learning prediction standpoint. While signals within the data exist, and in a smaller subset of rounds, prediction can be relatively successful.

Ubisoft have been relatively successful in balancing the game for operators and maps, possibly for the purposes of player enjoyment and player retention. They have managed to have a relatively consistent win rate of approximately 50% for all operators.

The data used is quite coarse, and covers a variety of different teams and skill levels. It is possible with more fine data or data unique to certain players, prediction accuracy, or insight into the player's strategies could be gained.

Possible future work for this project could allow creation ofT a sample round to be inputted to predict the winner. Another addition would be the ability to enter previous statistics on a player, to create a profile for them; with this profile, a possibly more accurate prediction could be created unique for that person. Then, a player who has created their profile could create a fictional round, or possibly recreate a round they are about to compete in, to see their chance of winning according to the model.

Another possible improvement would be a possibility of scraping data off of player profiles online. Within a player's profile, some statistics are public, and could be scraped to mass produce profiles for players. With this personal data along with the data results from this project, it could possibly lead to a more accurate model due to the personal nature of the data. If profiles had been created of the enemy team, you could get a strong indication of that team's strength also.

7 References

[1]. What is an Elo Rating? https://www.thechesspiece.com/what_is_an_elo_rating.asp

[2]. *Introduction to the Data Peek*, Ubisoft Blog, the data dump file is hosted on this page https://rainbow6.ubisoft.com/siege/en-us/news/152-293696-16/introduction-to-the-data-peek-velvet-shell-statistics

[3]. Hsinchun Chen, P. Buntin Rinde, Linlin She, S. Sutjahjo, C. Sommer, D. Neely, *Expert prediction, symbolic learning, and neural networks. An experiment on greyhound racing*, Arizona Univ., Tucson, AZ, USA, December 1994

[4]. Olivier Delalleau, Emile Contal, Eric Thibodeau-Laufer, *Beyond Skill Rating: Advanced Matchmaking in Ghost Recon Online*¸ University of Montreal, QC, Canada, February 2012