Pre-match Prediction In League Of Legends Using Machine Learning With Deep Analysis Of Gameplay-styles And Player Profiles

by

H00359327 - Nicolas Albiges

Submitted for the degree of

Msc Data Science

COMPUTER SCIENCE

SCHOOL OF MATHEMATICAL AND COMPUTER SCIENCES

HERIOT-WATT UNIVERSITY

August 2021

Declaration of Authorship

I, Nicolas Albiges, confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed : Nicolas Albiges

Date : 10 August 2021

Abstract

Electronic sports, E-sports are computer games played in a competitive environment between human competitors. This new kind of sport is growing. There exist multiple E-sports, they can be played between teams or individually and these confrontations take place in video games competitions in physical events or online on the Internet.

In that regard, gathering data is easier because they are computers video games, a lot of computer games provide game player statistics and match histories freely available through a web-based application programming interface (API). Indeed, gathering data for traditional sport is harder because each player is not related to a computer directly while playing, most of the time an individual who is watching a game has to count and manually write statistics down.

This research will take data from one of the most famous online video games called 'League of Legends'. League of Legends is a multi-player online battle arena game (MOBA). In the game, two teams of five players battle in a player versus player combat, each team is defending their side of the map and each player is controlling a character also called "champion" which will become stronger the longer the party lasts. The winning team is defined by the team who achieve to destroy the enemy base. In this research we predict, thanks to in game data and machine learning algorithm, the winner of a game. We train our machine learning algorithm with thousands of games played available from players who are very skilled playing at a high level in the leaderboard of League of Legends.

After the demonstration of our prediction and the right knowledge acquisition, we made a list of different gameplay styles and profiles of players. We assigned each player their playstyle and created a dataset composed of in game player statistics and player playstyles.

Finally, once all the datasets were created we conducted experiments to test whether combining data about playstyles with an in-game dataset gives better predictions than prediction from the in-game dataset alone. We obtained the same best results with 85% of accuracy with each dataset but across the other models with different parameters we

obtained a small improvement thanks to the playstyle data. However, the in-game dataset still is the one who has the biggest impact on the prediction and adding the playstyle has only a slightly impact. Nevertheless, the playstyle dataset could be improved and this is why we conducted a second experiment. We also considered whether a continuous representation of playstyle gave better results than a discrete two-valued representation. The results from the predictions of game with only playstyle data showed that continuous data was an improvement over the discrete data with an increase of 9%. Between the two types, we achieved 63% for the discrete data and 72% for the continuous one.

Acknowledgements

I would like to thank my supervisor Dr. Diana Bental who guided me through this journey and gave me precious advice. I would also like to give a special thank to Mr. Hichem Ammar-Boudjelal who is working at Dpliance and helped me to give his knowledge, perspective from the game and, his passion for this topic. Finally, my friends who play the game at a high level and helped me to analyze my results and gave me valuable input and feedback.

TABLE OF CONTENTS

LIST OF TABLES

FIGURE TABLE

GLOSSARY

AI Artificial Intelligence.

API Application Program Interface.

Elo Elo rating system.

KDA Kills Deaths Assists.

KPI Key Performance Indicator.

LoL League of Legends.

MOBA Multi-player online battle arena game.

Xp Experience Points.

Chapter 1

Introduction

E-sport also known as, electronic sport, is a form of sport in which professional players are competing in video games. This ecosystem is fairly new, as the first tournament was organized in the 1990s and with the growth of Internet, it has never really stopped growing.

Thanks to the research of "The influencer marketing hub", since 2016, there has been a significant increase in E-sports viewers, both occasional and enthusiasts viewers. Between 2018 and 2019, there was a 12.3% increase year over years. In 2019, there were 245 million casual viewers and 198 million enthusiasts, making the total audience 443 million [5, 16, 27].

In addition to the COVID-19 crisis which forced people to stay at home, many of them took the opportunity to discover new ways of entertainment. They came across new platforms and one of them was Twitch. "Twitch" is a live streaming service owned by Amazon that is well-known for hosting most E-sport tournaments. A report made by Streamlabs showed a significant growth, with around 20% increase in hours watched [27].

Thanks to Streamlabs's report of hours watched, across all the E-sport competitions during the third quarter of 2020, the biggest competition is the LCK (League of Legends Champions Korea) and has been watched for 33,310,312 hours. This was closely followed by the PUBG Mobile World League 2020, with 33,179,000 hours watched, and League of Legends European Championship, with 28,957,234 hours watched [16].

On this project we will work on League of Legends. League of Legends is an online computer game created in 2009 by Riot Games and has one of the biggest E-sport competitive scenes. From the famous business magazine called Forbes, in 2014 Riot games registered up to 27 million players playing daily and 67 million monthly [44]. In 2021 a fan base website called Leaguefeed made an investigation and found out 115 millions

players monthly have been playing the game [41].

Basically, a game is played by two teams composed of five players each, one on the "blue" side and the other on the "red" side. Each player has to choose a role before the game starts. There are five roles :

- Top: playing in the top lane
- Jungle: playing on all the map
- Mid: playing in the middle lane
- ADC: playing in the bot lane
- Support: playing in the bot lane

We can clearly see each role thanks to figure 1.1.

Figure 1.1: League of Legends map

At the start of a game, each team has to pick a champion which they will play. Champions are the player-controlled characters in League of Legends, each champion possesses unique abilities and attributes. Today, there are 156 released champions. At the same time they pick their champion, each player can remove one champion from the list of available champions. Once a champion is picked or removed, it will be unavailable for the other players of the game. When each player has their champion, the game starts. The goal is to destroy the base of the enemy team. During the game each player will gain

experience points which allow the champion to level up, so that he can learn new spells and increase these spells levels. Gold allow each player to purchase items from the shop and therefore more power.

This ecosystem is dynamic and interesting to work with. We can retrieve a huge amount of data through the Riot's games API which is available for the public [34]. Thanks to the API we can get precious information from the game. Indeed, we can get in-game statistics of each player after a game is over. Once we get these data, we can pick the best ones which we call key performance indicators, KPIs. These indicators, but also called features, are given to a machine learning and will predict an output, in our case the outcome of a game of League of Legends. Machine learning is a subdomain of Artificial Intelligence, also called or AI, it's an algorithm that improves itself through experience using data.

Furthermore, we want to go beyond the previous papers using prediction. We want to add a new perspective using player playstyles. Playstyle can also be called gameplay style, it defines the way a player is playing the game. Each player has their own way of playing the game but they can be classified, some players are more aggressive than others or some of them are more passive. These new features will improve the accuracy and the ecosystem. The goal is to create a player team which has good synergy between players. In fact, this is one of the mandatory things to do for a team, several papers underline the importance of the team composition to make a team win [19, 42].

However, sometimes the best players with the best statistics will not perform better than a team with bad statistics but good synergies. This dimension needs to be exploited by the team owners when they are selecting players to build a team. Building a team in E-sport to perform in a video game can be a very tedious, and yet vital process. The manager in charge of this task will have to be very careful because it will dictate the results of the future team and of course how much the E-sport structure will have to spend in salaries for each player. Often, the most expensive are those with the best statistics, however these players will not be a good fit in certain teams because they will not have the best synergies with their future teammates. This is why we want to study, and question ourselves whether the player playstyle actively affects the results of a team in LoL or not.

1.1 Aim

The aim of this research is to find an array of key performances indicators which could determine which team will win during a match. These indicators will come from in-game data collected from an open API. In addition we want to introduce the concept of a playstyle, and new key indicators related to a player's playstyle, and determine if these new features will help us predict the outcomes of a game of League of Legends. In the future, this could very well help E-sport teams to perfect their management design inside their structure, especially when they want to acquire new players and make sure the newer recruits fit well with the other members.

1.2 Objectives

The main objectives of this project that need to be completed should be the following:

- Understand the data and factors which have the most impact on a game of League of Legends
- Find the right KPIs to predict the outcomes in a game of League of Legends games
- Thanks to the KPIs, find the player playstyle that performs the best for each League of Legends role
- Build and evaluate a machine learning model that predicts the outcomes in a game of League of Legends games with different datasets:
    - In-game dataset
    - Player playstyle datasets
    - Combination of in-game and player playstyle
- The difference between discrete and continuous data for the player playstyle dataset

The optional objectives that may be completed depending on the amount of time left are:

- Retrieving pro players games from a private server in Riot's API
- Analyzing the differences between professional games and amateurs ones
- Building and evaluating a machine learning model that predicts the outcomes in a game of pro players in League of Legends games from in-game data and player playstyle

1.3 Motivations

E-sport is growing each year and will be one of the biggest industries in the world with the growth of Internet. Like sports it is a new way to entertain people through high level performance which is taking place in a video game.

In addition, this project is filling a gap in the research topics in League of Legends. Indeed, there is only two papers which dealt with the importance of using player playstyle to predict the outcome of a game of League of Legends [8, 30]. If this project and research proves to be successful, it could be a good opportunity for E-sport teams which are looking to improve their recruitment.

Chapter 2

Literature Review

2.1 State Of The Art: E-sport

"E-sport" started around 1980 with the birth of the first multi players games. In the years 2000/2010 Internet became more and more mainstream, and a lot of new people were drawn to use the Internet.

Thereby, a lot of people are contesting the fact that E-sport is an actual sport. Several papers have tried to make a definition of E-sport and the difference between the traditional ones. The general conclusion is that E-sport and sport has a lot of similarities. Both are requiring a huge amount of skills, hard training sessions, a good competitor mindset and surrounded by big competitions with significative prize money and high stakes. The main difference is, E-sport lacks physical attitude, however it requires other cognitive sense like the eyes and hands ordination and insane reflexes [18, 28].

Nevertheless, other papers made an analysis on the future of E-sport and if this new discipline is still growing it will be comparable and defined as a new sport and therefore become mainstream [23, 31, 36].

Throughout the years, more and more people began to play video games as well, although there is no denying that this phenomenon mainly concerned a younger population. As a consequence, some companies and brands start to regard E-sport as a lucrative media to invest in. The E-sport community, nowadays, represents a part of a population which is massively composed of younger players that are, for the most part, between 12 years old and 40 years old. The companies that want to see their brands name on the different broadcast of E-sport tournaments, have grown, as a result, more and more numerous. An interest that also allowed E-Sport to become financially viable, as more sponsors meant more money, and more money meant bigger prize pools for the winners to take home, and potentially afford a living with [18].

Like more traditional sports such as football, everyone can play the game but many people would rather watch it because they may not have the time or talent to directly participate in the most intense games and thus simply enjoy spectating the highest level of competition. One study also tried to compare the game of Chess with E-sport, called "Esports: The Chess of the 21st Century". Chess is very similar to many E-sport games indeed. An E-sport player is forced to make the best out of his perceptual-cognitive abilities, such as anticipation, decision-making or visual research but also has to be in fairly good physical shape, to actually allow his keyboards and mouses to follow the lighting-fast decisions taken by the brain, and maintain a stable coordination between the two [32]. A twofold necessity that can be found in both Chess and video-gaming.

With the affluence of many skillful players, the ecosystem has become more professional over the years with a better viewer experience, better user interface, better image quality and content overall. As stated in a research there are different ways to consume E-sport, based on different motivations for doing it. The study used different approaches and techniques to measure it, and achieved to find some very enlightening conclusions. A consequent part of the viewers is watching E-sport for the different shows on stage and the fact the games are sometimes truly beautiful to watch in term of aesthetics. Other viewers are watching it to think about something else, they find some escapism in E-sport. Furthermore, viewers can acquire knowledge for themselves out of the games as a large proportion of E-sports spectators play the game themselves. Finally, of course they do appreciate the skills of the professional players they are watching [18].

For a long time, video-gamers who played locked inside their rooms were widely considered as anti-social, and yet the success of E-Sport tends to prove that social emulation to support their favorite E-sport team are comparable in terms of amplitude to the most active soccer supporters teams. Furthermore, E-sports players amateurs or professionals can also attend Local Area Network (LAN) and play physically, not online, shattering the distance, and bringing a population with common tastes together.

Plus, as a direct consequence of the COVID-Crisis, E-sport was one of the few ecosystem that was able to gain affluence. Because most of the people were stuck in their homes, they discovered new ways to entertain themselves. Some by playing video games they wanted to improve in, some others by sheer curiosity [17]. During the pandemic some traditional sports got shut down and were not able to continue. However, in motorsports like F1, they have found new way to train on tracks and battle with each other. E-sport has made this possible thanks to home-based simulated racing. Simulated racing or racing simulation, is a racing game software that attempts to accurately simulate racing, with real-world variables such as fuel usage, tire wear and grip, damage etc. Hence, the traditional races drivers have been racing in real conditions against traditional race drivers from various competition like F1 or Indy car but also with E-sport professional players in big sim racing tournaments [45].

Nowadays, most of the E-sports teams, on an amateur level, are formed via social networks and friends playing together. However the professional teams use more elaborate tools and take very complicated choices regarding decisive factors such as the salary and external expenses when building a team. Actually, several papers have pointed up the importance of the team composition [19, 42].

We found, in that regard, an interesting paper related to this topic named "Understanding eSports Team Formation and Coordination" [11]. It brought how teams are created, how they are interacting within themselves and several other subjects. As stated in that paper the most common sites used to recruit team members were Reddit, Facebook pages, leaderboards, and online game forums. [....] P19 (male, 19, amateur) said: 'In LoL specifically, I used to browse either the public forums or forums on a third party site like Reddit to find teammates from around the continent.'

Overall, the leaderboard, when the game provides one, is undoubtedly useful for people who want to find teammates that are good and reliable even if they are complete strangers. As a manager from a team stated: 'We will find the ones on top of the leaderboards, and trial them for hours. It's always about reputation. OMG the application process was nuts.' [11].

Nevertheless, as stated in the paper, statistics are not always the best indicators to build teams, the players playstyle and synergies are also very important: “Gaming skills were fundamental for an eSports team, but meshing personalities and gameplay styles was critical for making this team sustainable. [...] For example, a few participants explained why being a pleasant peer was the key to bond the team, as P1 (male, 23, amateur) described, ‘You can always develop skills or teach mechanics over time. But you cannot teach passion or attitude. A selfish player is a bad fit no matter how good.’” [11]. In fact, this is the reason why this project could be a good help for those who would like to put a team together.

Finally, the strong aspect of E-sport is the fact these games are all computer-engineered. Thus, data can be retrieved in real-time if the game permits it through an API. In our case, this project will be conducted on League of Legends, and we have the chance to have access to data after every game [34]. On top of that, nowadays, we could say that data are one of the most valuable assets a human can have.

2.2 State Of The Art: Machine Learning In E-sport

Some papers are using another game for their research, called “Dota 2” [5, 8, 25, 39, 49]. Dota 2 is the direct competitor of League of Legends. It was created by Valve which is also a company that owns “Counter-strike global offensive”, a well-known FPS (First Personal Shooter) game. Valve is also a hosting platform for video games. The rivalry between Dota 2 and League of Legends has been running for years as the two belong to same type of games, the MOBA. Both are composed of two teams of five players and their goal is to destroy the enemy base with their champions. In League of Legends the players are controlling their champions and in Dota the players are controlling their heroes. In both games each team gets to enter in a “pick and ban” session, where each player picks a champion from the champion pool and bans one. There are admittedly a lot of similarities between both games, but there is also some key differences between them that makes each game feel unique in a sense.

In this research report we will try to predict the outcome of a game in League of Legends. Several other researches have tried to predict them not only in LoL but also in Dota 2. Moreover other types of predictions do exist, not only for the outcomes of a game.

These types of prediction can be called “micro-predictions”, they are predictions of different events during a game, for example the outcome of a team fights, which team will win the team fight but also who will die and survive. But, also for example, which player will most likely be the best player out of the game by several metrics, kills, assists, gold etc. All these predictions are made with data the game offers thanks to their API. In fact, these predictions are using machine learning algorithms. Through the presentation of the different papers, we will try to see a potential, common trend in terms of how machine learning is used, which will compel us, later on this research report, to dive more in depth about these machine learning models.

The first paper “How Does He Saw Me? A Recommendation Engine for Picking Heroes in Dota 2” [5] tries to predict the outcomes of the “pick and ban” session from a team. They have tried two different machine learning models: the first models is called Logistic Regression, to achieve their goal they have used the pick and bans of a game as a feature for the model and tried to predict the outcome of a game. Their results were not exactly promising so they tried with a different approach this time using a different model and different features. They used the K-Nearest Neighbors model and achieved good results. However, in their conclusion they stated they wanted to try different models because the K-Nearest Neighbors was too slow.

The prediction of the pick and ban session is a common topic, two other papers have tried to use machine learning to predict a “pick and ban” session. Both of these papers have used specific machine learning algorithm. The Long Short-Term Memory, also called LSTM, belong to the family of neural networks and more specifically in Recurrent Neural Network (RNN). This type of model enable the algorithm to process not only single data point but also entire sequences of data. All the papers who handled the picks and bans prediction have used previous pick and ban session from the hundreds of thousand game played [43, 51].

In the “E-Sports Ban/Pick Prediction Based on Bi-LSTM Meta Learning Network” paper, they have recorded a good accuracy across their different models, from 60% to 80% of accuracy and the LSTM gave the best ones [51]. This paper “Draft-Analysis of the Ancients:Predicting Draft Picks in DotA 2 Using Machine Learning” also used Bayes

nets as machine learning model but they did not showed their accuracy neither their results, their research was purely experimental [43].

Other papers tried to predict different things in Dota 2 and Lol, some of them tried to perform real-time match predictions. For example, the paper "Real-time eSports Match Result Prediction" [49], performed prediction on the outcomes of a game while the match is playing in real time. To do this performance they used a combination of real time in-game statistics and also player statistics from their previous matches. This gave plenty of features for the machine learning models to perform its predictions. To complete their prediction, they used different models. The first one is, once again, the Logistic regression. They achieved a good accuracy of almost 72%, and also tried it with a different machine learning model which is Neural Network. They also had a good result with the Neural network, 70%, however they could improve their results by adding new features to train their models.

On the other hand, some papers are also trying new techniques such as deep learning algorithms. As part of game prediction, "Time to Die: Death Prediction in Dota 2 using Deep Learning" [24] tried to predict death in a match of Dota 2, this type of prediction is called micro-prediction. Micro-predictions are granular predictions about what might happen in the near future in the game, as opposed to predicting the outcome of the match itself. They complete their predictions using Deep learning models which is an improved version of neural networks. They again reached a good accuracy, 70% on average.

Predicting the outcomes of a game is a focal point of such analyses. For example, this paper "DOTA 2 Win Prediction" explains how to predict the outcomes of a game [25]. For that purpose, they use the Logistic regression and a Random forest. After some experimentation the Logistic model was the most stable and efficient with 74%. However the random forest had some trouble with the dataset, the model suffered from over-fitting, about which we will discuss more in details later. And their best accuracy was 67%. In addition another paper who tried to complete the same goal to define a winning team of a game, "Predicting the winning side of Dota 2" [39]. Nonetheless they only used Logistic regression and trained their models with a small dataset which contained only some heroes of the said teams. This is why they received nothing but bad results with this

model. However, this paper "Win Prediction in Esports: Mixed-Rank Match Prediction in Multi-player Online Battle Arena Games" also tried to demonstrate how to predict the outcomes of a game by using two datasets, one with an amateur playing the game and another one with pros. For both datasets they have used Logistic regression and Random forest and each one had great results. The logistic regression had 75% on amateurs games, 71% on pros and Random forest had 76% on amateurs and 68% on pros [20].

An interesting paper called "Classification of Player Roles in the Team-based Multi-player Game Dota 2" is doing prediction on game styles of a player. They used a variety of machine learning models. Once again Logistics regression and random forest gave the best results between 76.27% and 75.85% and three others which also gave good results overall. The sequential minimal optimization (SMO) 75%, Bayesian networks 72% and finally Naive Bayes 70% [8].

We can pinpoint a clear trend by going through the different machine learning models used. Logistic regression is clearly the most widely used one and also the one wielding the best results. Two others stick out, the first one is the neural networks and also the random forest. Nevertheless, the results are bound to remain unique as much as each paper had a specific set of data and different features. This shows clearly that data can have important impact on research.

2.3 Data Integration And Connection To An API

Data is essential for predictions and thanks to video games, data can be retrieved relatively easy if the game creator decides to open an API. Especially with League of Legends we appreciate the fact that Riot games, the game creator, has opened an API for everyone [34].

If a user wants an API key he has to agree with Riot's rules and policies. A user can request an API key daily or ask for a permanent one. For the permanent key Riot will have to approve the request. The request has to be valid with a good explanation of the goal and description of the project.

Then finally, he will be able to send request to Riot's API which will send back json files. Json files are very common in the Data Science world because they are easy to understand and work with. Although the json files have to be adapted to the machine learning algorithm. You have to clean the file and apply a pre-processing on the data but we will discuss more about it later in the sections 2.6.

Every dataset has its particularities, and the data scientist must check them before making predictions with a machine learning model. In this research paper we will create three different datasets with specific features each time. These features will be created thanks to Riot's API.

2.4 Key Performance Indicators

To predict games' outcomes, you must have good features for your machine learning models. Some features appear to be some key, almost mandatory factors to take into account, and of course very useful regarding our topic.

Every research paper that is using machine learning models is getting the data trough different APIs or datasets available on the web. These data contain the games of hundreds of thousands players and each player can have also hundreds of games.

First, the most obvious factor to study had to be the level of each player. In LoL there is a ranked system where each player possesses an Elo rating system, ELO. The elo rating system is a method for calculating the relative skill levels of players, it is often representing their ranks, higher the elo, higher the player's rank is [3]. The best players are indeed in the top of the ranking system and the leaderboards. The number of games can be a feature, and someone who has played a huge number of games is more likely to have more experience. However, someone who played a lot is not always highly so the ratio between the wins and losses must be taken into account. A player who has a high number of wins and low number of losses is likely to be a strong player.

A lot of the papers collected games from higher ranked players [5, 24, 25, 37] because most of them wanted to predict outcomes of professional games for various reasons. However, some papers also take amateurs ones or both [20, 39, 49]. In the end, it really depends on what the research paper goal is.

Before each game, each player has to pick a champion which they will play with during the whole duration of the game. Each player has their favorite champions and some they has never played with. In addition, specific champions are better against specific champions and each player has their own personal stats going along with this match-up knowledge. If a player plays for the first time a champion, chances are the performance will have less impact during the game than with a champion they have played for several hours. One paper in particular tried to predict this feature [5, 25] other ones used it as a feature to train their machine learning models.

Few papers used vectors to hold the possible composition of each team's champions. In both games there are a lot of champions, Dota 2 contains 119 heroes and LoL 155 champions as of today. This paper, for example is using this method to define the hero selection with, at that time 113 heroes available, 113 heroes $\times$ 2 teams = 226-dimensional vector [49]. Thus, other papers also used these methods [24, 39].

During the game each player embodies the Champions they have picked. There are plenty of features available which we can analyze during a game. However some are more important and powerful than the others. The champions can reach the maximum level of 18 and for that the champion needs to earn experience points. For each level gained, it upgrades its statistics, such as speed, strength, or resilience. You can also gain new skills for your champions by buying items from the shop during the game, each items costs golds which you can earn via multiple means. Three, actually: the first consists in destroying the objectives, the second one in killing some enemies and finally, farming minions. Minions are spawning periodically from their home base in each team and advance along a lane towards the enemy base, automatically engaging any enemy unit or structure they encounter. They are controlled by artificial intelligence and only use basic attacks [3]. Every paper uses these features which give a lot of information about the game.

In addition, the API track each action a player is doing with the time code associated. Hence, we can know at what time the champion passed a certain level or bought a certain item. We can also track when the objective was destroyed, by whom. We can also track the position of each member when an event occurred, when someone was killed, by who and its position for example.

Finally, post-match features, which is a sum-up of the game, the kills, number of deaths and assists of each player. All the previous events and features summed-up. And of course, the outcomes of the game, which team has won. In this research we want to predict which team will win a game of LoL.

2.5 Gameplay Styles

Gameplay style, or also called player playstyle, are the ways a player is playing the game. Although all the papers didn't use gameplay styles as KPI to improve their predictions, two found it interesting and tried to predict them [8, 30]. In a game of League of Legends, each role requires certain skills and certain gameplay styles that are known to be working better in certain roles. For example the role "adc" is, most of the time, taken while playing characters with lower than average health. So an adc has to adopt a playstyle which will require more self control and rigor, an adc will have to play safe and always play with his support teammates, never alone.

On the other hand, other roles like top or mid can afford more aggressive behaviors, the champions better suited for that task have a good amount of health and don't truly rely on other members to do well. As a consequence, they can get early kills, and enjoy more brutal approaches at the beginning of the game, on their own.

The support and jungle are more in the spirit of playing with the team. They have to help their other teammates during the game, that is what they were designed for. Sometimes the jungler will help his teammates in the toplane by killing the enemy toplane players. This gameplay style can be defined as a "Team player".

These assumptions are, for some of them, confirmed thanks to this paper which also analyzed the different gameplay styles possible for the Leagues of Legends players. They determined five different types of gameplays thanks to the in-game statistics of each player across 10 000 games played [30].

The first one called “Ranged physical attacker” is a player who maintains distance from fights while dealing high damage with long-range attacks. The second ones “Ambushers” are players who move stealthily around the battlefield and engage in quick. Then we have players called “Team support”, which are players who assist ranged physical attackers by healing, using cooperative attacks etc. The fourth gameplay style is “Magic attacker” it’s a player who relies on magic-based attacks as opposed to physical damage in the gameplay styles. Finally, we have the last gameplay style “Miscellaneous” this is when a player has no clear style defined previously and hard to define [30].

In addition to these playstyles, they stated within gameplay style they had different clusters for each player. For example if we take the “Ranged physical attacker” gameplay style, a player can be in this class but can also be aggressive or passive, this is why they used clusters within their class. Players in each cluster differ in risk attitudes, such as whether they attack deeper in enemy territory. In other words, a player can be a “Ranged physical attacker - aggressive” because he goes deep in enemy territory or on the other side someone who plays safely will be classed as a “Ranged physical attacker - passive”. Each gameplay style has its own clusters.

To define this gameplay style they used the in-game statistics of each player and used clusters to fit every player in their gameplay style. When all the players have their gameplays style associated they predicted the outcome of game. This paper achieved up to 72.3% of accuracy [30].

Another paper analyzed the gameplay style. However, they tried to predict purely the different gameplay style for each players. Moreover, this paper went in depth with in-game statistics, also called indicators, used for the predictions.

The first feature of this dataset is the position and the time where the player is in the map. It gives information on the player if he stays alone on his lane or maybe he is helping his teammates. Then it analyzes the statistics of a player during the game, how many times he killed someone, made an assist to kill someone or even how many times he died. In addition, it also tracks the type of the items bought during the game. If someone is buying a support item it's more likely this person who will play with his teammates. Also the damage dealt during the game, it's good to know if someone has an impact during the game [8].

This paper set multiples gameplay styles, here is the list: Ganker, Support/Babbysitter, Support/Roaming, Support/Farming, Carry/Active, Carry/Farming, Pusher, Feeder, Inactive.

There are three main gameplay styles related to the role. The first ones are the Gankers which try to attack enemy heroes with surprise attacks, sometimes very early in the game. Support players in different ways try to help other players, often even sacrificing themselves in the process. They define three kinds of support players to cover different strategies. Babysitter support players, protect teammates, usually a carry. In contrast, roaming supports are active around the map and help their teammates. Farming supports also take their share of experience and gold, sometimes even steal to their carries. The "Carry" who is usually weak and needs protection early on, but is very strong in later stages, often defining the conclusion of some games. Carries typically end up with a high gold per minute and overall kills. Active carries engage enemy players and participate in team fights to gain experience and gold, while farming carries focus on using enemy or neutral minions for character development.

Pushers always try to destroy enemy towers, thereby pushing their lane. Feeders are players that show very bad performance during the whole game. Finally, inactive players are a small percentage that have technical difficulties and do not actively participate [8].

They made good results to predict the gameplay style of players with 75% for the logistic regression which was the best score of all the machine learning models they used. However classifying players with gameplay styles is not easy. For example, the farming

carry wants as much gold and experience as possible and the pusher on the other hand wants to destroy towers as fast as possible but these goals in many cases can be achieved by the same actions. It can be confusing for the machine learning to predict. Furthermore, the paper has also asked experts in E-sport to define the gameplay style of players and even them seemed to struggle.

Moreover, there are different ways to play the game, and each player has their own one. In League of Legends it unquestionably depends on the champion you are playing, the champion you are playing against, and also on your choice of playstyle. Some players like to adopt an aggressive style, some don't. This is why you have to make up a team with good synergies and sometimes this is no easy feat.

2.6 Cleaning The Dataset And Pre-processing

Usually, data are formated in a way to give every features possible. However sometimes data are incomplete or missing and you have to handle every case possible. This part is often underestimated, and you have to take it seriously because it can give you a lot of wrong if data are incorrect or incomplete, especially in the machine learning world.

In our case we will have to handle json files, this is a good starting point because it is most of the time well organized and clear. Nevertheless, you have to be careful and check all data an API sends you, even more when the data can contain hundreds of thousands of lines.

A lot of papers which are doing prediction for Dota 2 games are getting their data on datasets available on the web. For example this paper got its data through an API called "OpenDota" where they collected 5000 Professional (Major tournaments) and 5000 Semi-Professional (Minor tournaments and leagues) games [24].

After getting the data you have to do a very important method for your machine learning model which is called pre-processing. Pre-processing is a process of preparing the raw data and making it suitable for a machine learning model. Some machine learning does not handle categorical data which are string of characters, they only handle numeric

data, this is why we have to change the categorical data into numeric data [9, 15].

Especially in LoL or Dota 2, datasets can contain some features as categorical data. Several papers have handled the champions chosen by a team which are a list of strings. There are some solutions, the first one is Integer encoding. This method is simple, it will class every string in order, like this example: "car : 1, bus : 2, plane: 3, train: 4". However, it's not the best method, because the machine learning model can understand that maybe this is an order of strength. Maybe the last one (4) is better than the first one (1), which is not what we are looking for. This is why the papers found a better solution which is an elaborated version of the Integer Encoding, called One-Hot encoding [49]. Basically, the integer encoded variables are removed and a new binary variable is added for each unique integer value. In our case here, in Table 1, there is an example using the previous one.

car	bus	Plane	Train
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

Table 2.1: One-Hot Encoding Example

2.7 Technologies

Nowadays, there are multiples ways to work with machine learning algorithms. However, there are some that tend to come out a bit more than the others. There is one particular language in specific which is truly active and well documented called Python. Another tool which is interesting when you start to learn machine learning is Weka which is a software tool specialized in machine learning. And of course you have some libraries available in a lot of other languages such as C++ or Java for example.

Python is an object oriented and functional programming language. Within Python you have some interesting "libraries" which are commonly used in the machine learning community, named Scikit-learn and Tensorflow. A library is a collection of functions or objects which can be added to an application using the same programming language. Scikit-learn and Tenserflow duties are to help people who want to learn and use machine learning algorithms and are both open source. The fact both are open-source allowed the community in contributing and thanks to that, new machine learning techniques are discovered almost everyday [29]. As a new machine learning developer you have a huge amount of data available on how to make things work and understand it. These papers [5, 39] used python and more particularly the Scikit-learn library to implement their algorithms.

There is also Jupyter notebook, which is based on the Python language. Jupyter Notebook is an open-source web application that contains live code, visualizations and narrative text. This is actually used to give visualization of the code you produce in a small to medium project. If the project becomes bigger Jupyter will stop being a good fit.

Weka is an open-source software which provides tools for data pre-processing, implementation of several Machine Learning algorithms and visualization tools. This software is made for beginners in Data Science and not a good pick when you are handling a huge amount of data because it will be restricting [29]. However, this tool has been used in a lot of research papers, [8, 20, 37] they used it to perform their pre-processing data and machine learning models.

Other tools and languages exist to use machine learning. However, some of them are made for this, but most of these at this stage are not. For example if we pick C++ which is popular and widely used in the programming world, we find out that it isn't really the case here. There are not a lot of libraries and documentation about that topic. You often have to do it from scratch by yourself, and the language is sometimes harder than the other in comparison to Python which is very friendly. In opposite, C++ offers the advantage to hand yourself the memory and python doesn't so if you are looking for good performances you can use this type of language [47].

2.8 Machine Learning Algorithms

As we have seen previously we can see a trend of machine learning used in E-sport. Particularly three of them stand out. The first one is the logistic regression.

The logistic regression is the most used machine learning model, it is also the one which gives the best results in most cases [5, 18, 25, 30, 39, 49], especially because most of the papers are trying to classify something and Logistic regression is working well in this domain. Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The output of these models is binary, either 1 (success) or 0 (failure). This is one of the most used machine learning algorithms thanks to its simplicity to use and learn.The logistic function, also called the sigmoid function is following this formula and represented by this graph [10, 40, 46].

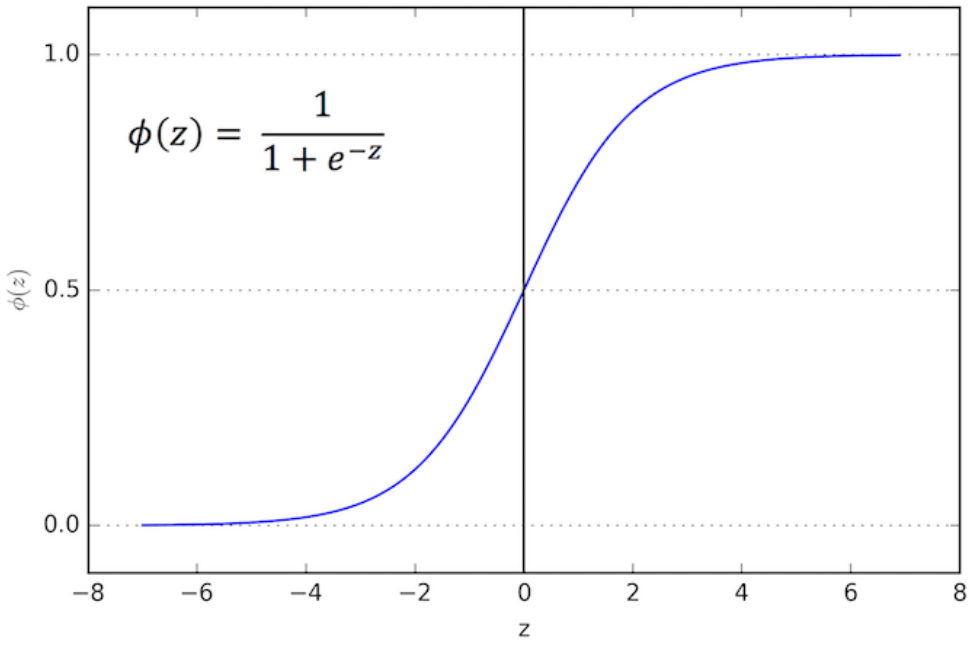

Figure 2.1: Logistic regression function representation

The second model is the Random forest, this model is very common in machine learning, it is known to be very fast and easy to industrialize. Various papers used it and got respectable results [8, 20, 25, 39]. The Random Forest algorithm is a classification algorithm that reduces the variance of a single decision tree's predictions, thereby improving their performance. To do this, it combines many decision trees in a bagging-type approach. In its most classic formula, it performs parallel learning on multiple decision trees built randomly. This machine learning is also easy to use and to understand. It is widely famous as it is one of the most efficient thanks to its speed [26, 38, 46].

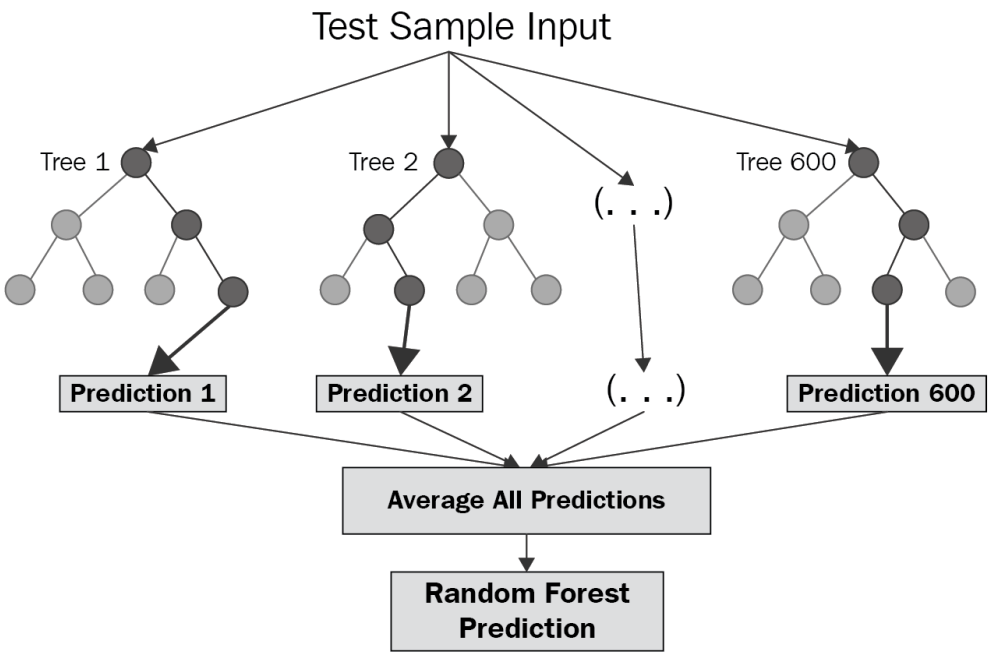

Figure 2.2: Random Forest representation

The third one is the neural network. This one may be the most famous because its original goal was to copy the human brain. This model is designed to recognize patterns out of the data it has received. A variety of different architectures of neural networks already exist and can predict different types of data. This one is one of the most flexible machine learning algorithms but it requires a lot of knowledge to use it efficiently. It's also often related to deep learning which is simply layers of multiple neural networks paired together [7, 48, 50]. This algorithm is genuinely hard to work with, you must have a strong knowledge in machine learning and maths to make a good neural networking working, this is one of the reasons only two papers used them [24, 49].

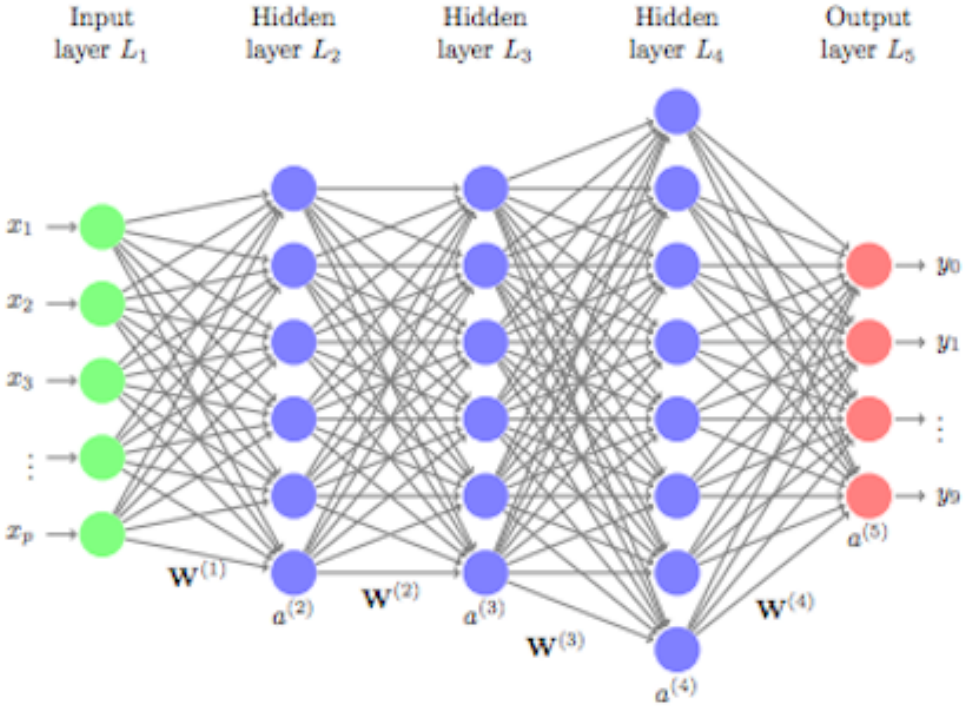

Figure 2.3: Neural network representation

Nevertheless, we can find other interesting machine learning algorithms. The first one is the K-nearest neighbors algorithm, it will classify an object by a plurality vote of its neighbors, with the object being assigned to the class most common among its K-nearest neighbors. It all depends on the number of objects named K we want to look for [12, 14, 33]. As shown in this image we can assign an object to a specific class. Only one paper used this algorithm [5].

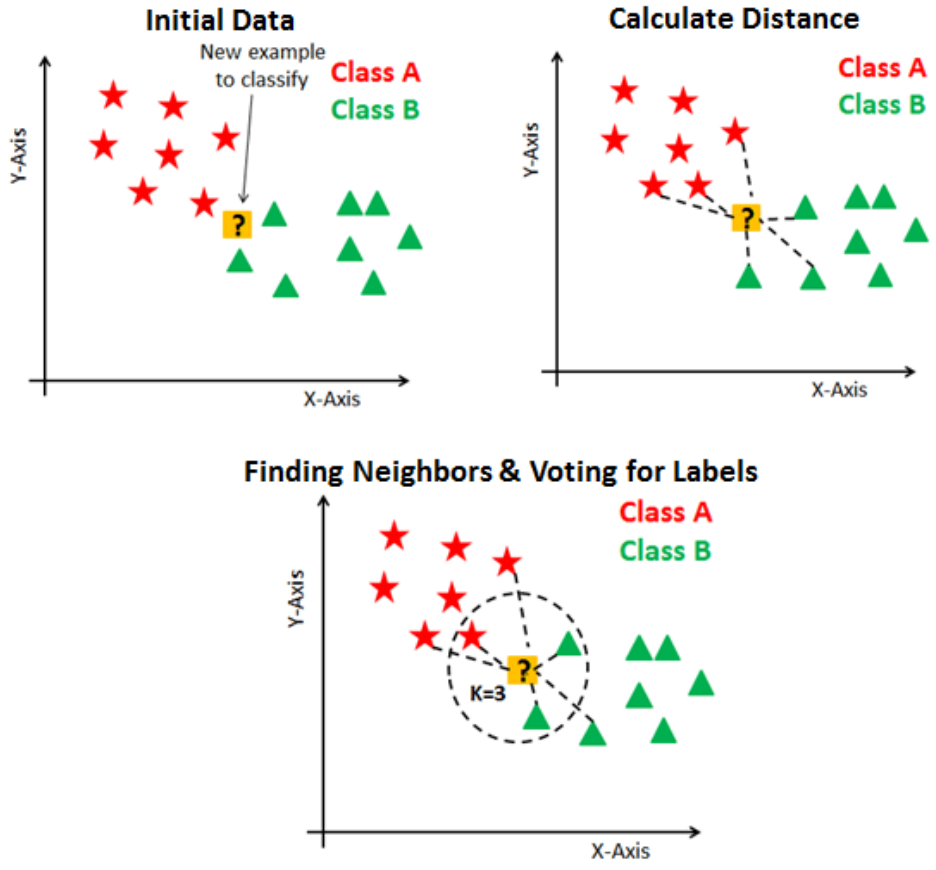

Figure 2.4: K-nearest neighbors representation

Finally, the last machine learning model is the Naive Bayes classifier. This is a probabilistic machine learning model which is used for classification tasks. The Naive Bayes are following this formula.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Figure 2.5: Naive Bayes formula

The probability of A happening, given that B has occurred. Here, B is the evidence and A is the hypothesis. The assumption made here is that the predictors/features are independent. The presence of one particular feature does not affect the other. Hence, it is called naive [35, 46, 52]. Once again only one paper used this model to do its prediction [8].

These machine learning processes can be all evaluated and with the results we can actually begin to analyze them.

2.9 Evaluation

This topic in the machine learning fields is often left aside. Obviously most of the papers showed their accuracy represented by a percentage [5, 8, 18, 20, 24, 25, 37, 39, 49] but as we have learned in lectures this is sometimes not the best metrics to take into consideration. For that you have to do a complete analysis of your result using other metric tools.

Nevertheless, accuracy is still an important indicator. However, you can be victim of over-fitting which is machine learning models that corresponds too closely or exactly to a particular set of data [7, 46]. One of the papers suffered that with the Random forest [25]. The opposite, under-fitting, is to avoid obviously so that you have to take very seriously the dataset and the machine learning model you are using.

$$\text{accuracy} = \frac{\text{correct predictions}}{\text{all predictions}}$$

Figure 2.6: Accuracy calculus

One of the best representations of metrics is the confusion matrix [1]. If you run a classification prediction, there are four types of outcomes that can occur:

- True positives happen when you predict an observation belonging to a class, and it actually does belong to that class.
- True negatives happen when you predict an observation not belonging to a class, and it actually does not belong to that class.
- False positives occur when you predict an observation belongs to a class when in reality it does not.

- False negatives occur when you predict an observation does not belong to a class when in fact it does.

This is often displayed like in the following:

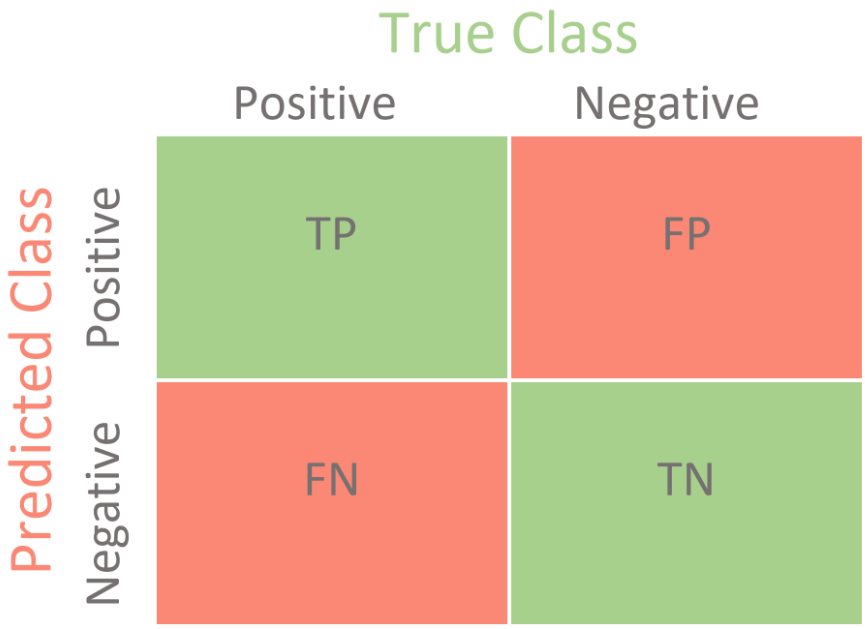

Figure 2.7: Example of Matrix confusion

Two other metrics that are related are Precision and recall. Precision is defined as the fraction of relevant examples (true positives) among all of the examples which were predicted to belong in a certain class [1].

$$\text{precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

Figure 2.8: Precision calculus

Recall is defined as the fraction of examples which were predicted to belong to a class with respect to all of the examples that truly belong in the class.

$$\text{recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

Figure 2.9: Recall calculus

The last metric is the F1-score, this metric is a combination of the precision and Recall. It gives a good balance between the recall and precision. This metric follows this formula [1].

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

Figure 2.10: F1-score calculus

These indicators are undoubtedly helpful and can indicate if there is a problem with your machine learning model.

Chapter 3

Requirement Analysis

In this section we will cover the different stages of the research paper, the project and which methodology we will use. We will explain the mandatory tasks and then the optional one finishing by the methodology.

3.1 Mandatory

In the first mandatory tasks, we will have to analyze the different features of Riot's API and create a good dataset out of it. Once the dataset is created, we will have to find the best machine learning model to fit our prediction which is the outcomes of a game of League of Legends. Then to understand the results we will have to conduct an evaluation. The evaluation will allow us to dive into the metrics as much as we can and explain the results, because we will have a lot of features and indicators we expect to have good results between 70% and 90% of accuracy.

After that we will try to define the different player playstyles possible thanks to the features we have from Riot's API. Once we have a complete dataset, we will run the same procedure that the one with the previous dataset, however we expect the results not to be better than the in-game data because these data will be created by our own hand and will not be directly related to the in-game data. So we cannot predict the accuracy we will get but there is a lot of chance getting less than the in-game data so, around 50% of accuracy.

Afterwards, we will create a last dataset composed of both datasets created previously. Once again we will predict the outcomes of a game and analyze the results. These results will be very interesting to investigate because there are two options. On one hand, having close results from the in-game data or less good result (60%-80%) because of the player playstyles data or the opposite, the player playstyle will make the model sharper and have the best results (80%-98%).

On the other hand, these information will be very useful for the final part which counts as our optional, and yet challenging, objective.

3.2 Optional

First we will have to analyze the games of the professional players and see the difference between them and the amateurs ones. We will also have to get data from professional games in the API. With the previous analyze of player playstyles we will create a new dataset for the machine learning model. The machine learning models will predict the outcome of a game once again but this time we will do a simulation of the E-sport teams. For example we will have an E-sport team which has five players but they want to change the fifth one. First we will try to find the best player playstyle for the team and then thanks to the machine learning algorithms try to predict with this new player what are the chances this new team will perform in the future. This could help the E-sport team to recruit members that will be really good in terms of performance and synergy with the initial, existing team.

3.3 Methodology

For the project management we used the agile methodology to organize the project. An agile methodology promotes continuous iteration of development. Every week you set tasks to perform through the week and the next week you see if the task assigned went well or not and if not can change the incoming ones [4, 6].

Figure 3.1: Agile methodology representation

Chapter 4

Implementation

In this chapter we will be showing how the project was developed and evaluated. The project runs under Python 3 and some libraries such as Scikit-learn for the machine learning models, Pandas for the data handling and finally Plotly and Yellowbrick for the data representation.

We are going to split this section into three parts. In each part we will create a dataset to perform our predictions on games outcomes of League of Legends. The first section will gather in-game data and create a clear dataset with key performance indicators. Then in the second section we will define the multiple player playstyles and assign for each player his/her playstyle. Finally, we will create the combination of the two previous datasets with in-game data and player playstyle data.

4.1 Prediction Of Games Outcomes In League Of Legends From In-game Data

4.1.1 Gathering the data

League of Legends has a ranking system for its player. The ranking system in League of Legends includes nine tiers, each split into four divisions, with four being the lowest in that tier and one being the highest. Players progress through each division and tier by earning League Points (LP) for each game that the player's team wins. In our case we will try to analyze data from the best players because in the future we will try to analyze the game of E-sport players in their professional leagues. This is why we will analyze data from the top players from the higher league possible, this league is called "Challenger". This league is composed of 300 players maximum [3].

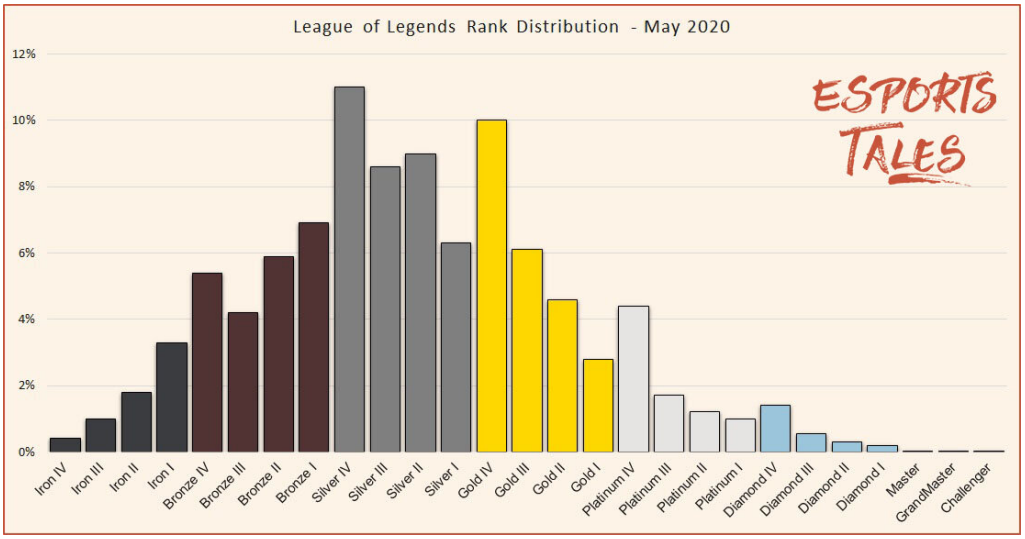

Figure 4.1: Player distribution between each rank

In order to create the dataset, we had to retrieve the games of this top 300 players. A game of League of Legends is composed of 2 teams of 5 players, each player can play with each other but sometimes, mostly late at night, League of Legends tries to make games even if there are not enough people from their tiers and make a game with lower tiers. So sometimes there are games with Challenger and top Grand master. We end up with 2400 players and around 600 unique games.

4.1.2 Formatting the data

From Riot's API we can discover two interesting API calls [34]. The first one gives all the statistics from each player and the second one is the timeline of all player's movements, actions and positions on the stage. There is a different event, especially the destruction of a tower, the kill of a champion, an item purchased etc. We have 600 games and now we have to take the best features possible.

For each player we will take into account: their statistics across all the games he has played. To that purpose, we took the average of each indicator. Here is the list of all the key indicators:

'goldEarned', 'kills', 'deaths', 'assists', 'trueDamageDealtToChampions', 'visionScore', 'totalHeal', 'champLevel', 'goldSpent', 'damageDealtToTurrets', 'magicDamageDealtToChampions', 'wardsPlaced'.

4.1.3 Creating a heuristic algorithm for players roles

The game is composed of 5 players in each team, and each player is playing a role. There are 5 roles, the top laner, the mid laner, the jungle, the carry also called ADC and the support which assists the ADC, as explained earlier.

Figure 4.2: Player roles repartition on the map

From Riot's API we can get the role of each player but it turned out that this information was sometimes missing or wrong. So we took the decision to manually create an algorithm which will give each player their roles by following a set of rules defined beforehand.

In order to create a set of specific rules, we had to retrieve all the games' timeline from each player. Thanks to those timelimes we have the position of each player each minute. In addition we have access to two important features called "jglMinionsKills" and "minionsKilled". These features are truly important to assign the good role. The

first role we assigned is the jungle, thanks to the indicator “jglMinionsKills”, we take the person who has the biggest number at the end of the game. Indeed, a jungler is the only player who will kill the jungle minions, this is why the player who has the most “jglMinionsKills” is the jungle of the team. For the other roles we have to be more methodical. During the first 15 minutes of a game, we track the position of each player, in the early game most of the time each player will play on their main lane and farm the minions to gain xp and gold. Indeed, at the start of a game they go as fast as they can on their main lanes to grow their ability and power and as the time goes by they start to play together for the objectives and to be more powerful. So for each player we take their position in the first 15 minutes and take the average of their x and y, if some players are in a certain portion of the map, we will assign this player his assignated role. Hence, for the top lane if the position is in the top left of the map the player is a top laner, for the middle laner if the position is in the middle of the map he is assignated to the mid lane and finally for the last two players who are located in the bottom right of the map we have to do more than that. Because the ADC and support are playing in the same area we have to differentiate them, this is where the indicator “minionsKilled” takes place. In fact, the ADC is the one killing the minions so we assigned this role to who is in the good and area and has the most “minionsKilled”, the other one in this area is assigned to support.

4.1.4 Pre-processing data

The pre-process is absolutely important because some of the machine learning models are not working with strings for example, so sometimes you have to transform those into numeric values. In our case, each game is composed of 10 different players which are playing for a team. So for each player from each team, we create a set of features. Besides, we have to make sure each column only contains a feature for a specific role. We don’t want to have at the 1st row and columns 1 to 14 a player who is playing in the top lane and then at the 2nd row a player who plays support. We have to be careful, so each player has 14 columns, each columns represents an indicator and is also the mean of that indicator across all the games that player has played and in total we end up with 140 different features. On top of that we have the final feature called “win” which stands for the team which has won or lost.

Once our dataset is created, we split it into two parts, the training one and the testing one. The training is composed of 70% of the main dataset and the 30% remaining for the test. This ratio is working quite well and even advised by the machine learning model specialist. In order to have the same distribution of games for every dataset we will use a seed. The "seed" is a starting point for the sequence and the guarantee is that if you start from the same seed you will get the same sequence [2]. In our case we have used seed equal 912.

4.2 Prediction Of Games Outcomes In League Of Legends From Only Player Playstyle Data

4.2.1 Define multiples different player play styles

Every player has their own ways to play. However some behaviors and playstyles will be more victory inducing than others, in addition, a composition of different types of players will also alter the percentage of victory. This is why, we have created 4 different families of playstyles.

The first one is **AGGRESSIVE**, the player will risk his life to gain the most out of it, also called high risk - high reward. The opposite of this is **PASSIVE**, it's someone who will play safe and avoid taking any risk that could lead to death.

The second one is called **TEAMPLAYER**. A teamplayer will play a lot with his teammates. On the other side the **LONELY PLAYER** will spend most of the time alone which will not prevent him from actively trying to win.

The third one, **LEADER** is related to the **TEAMPLAYER**'s style, but it's someone who will guide their teammates to the victory and making the team more powerful. If the player is not a leader, he will be a **FOLLOWER**.

Finally, a player can be **RESILIENT**. Even if he loses a game he will try to bounce back and try to win. On the other hand, someone who is **SURRENDERING** is someone

who is highly mentally affected by its deaths, making him want to stop playing, in addition they are unlikely to try to play a new game if they have already lost one or more in a row.

4.2.2 Creating a heuristic algorithm for players play styles

In the spirit of the algorithm assigning players into game roles, we have created an algorithm that could give the best traits to one player. For that we analyze each of their games after adding a set of rules defining their traits. In total there are 16 different playstyles for a player to be assigned to.

First, let us study the aggressive playstyle. We define aggressive someone who is taking a lot of damage. In the Riot APIs there is indicator called “totalDamageTaken”, it’s the number of damage taken from a player during a game. So first, we take the “totalDamageTaken”, we store it in a list and we take the median. Then to define if a player is aggressive we take all the “totalDamageTaken” of each game he has played and make the mean. In addition, we watch this feature “longestTimeSpentLiving”, as the feature is named, it’s the longest time living during the game. If the means are both over their respective median found previously the player is defined as aggressive player, otherwise he is defined as passive.

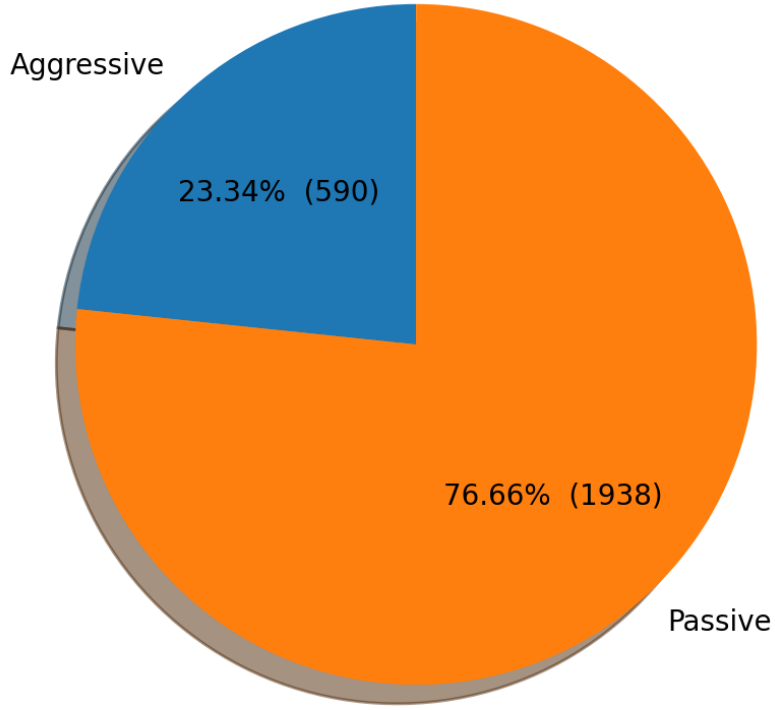

Figure 4.3: Pie chart representation of aggressive and passive players playstyle

We define a teamplayer someone who has more "visionScore". This indicator follows this formula Vision Score = (1 point per minute of ward lifetime provided) + (1 point per minute of ward lifetime denied). A ward in League of Legends is a deployable unit that removes the fog of war in a certain area of the map. The fog of war is the area of the map in which a team does not have sight over. So it helps the team to move around the map more easily and even set traps for the enemies [3]. So, if the player has higher "visionScore" he is placed into the teamplayer category.

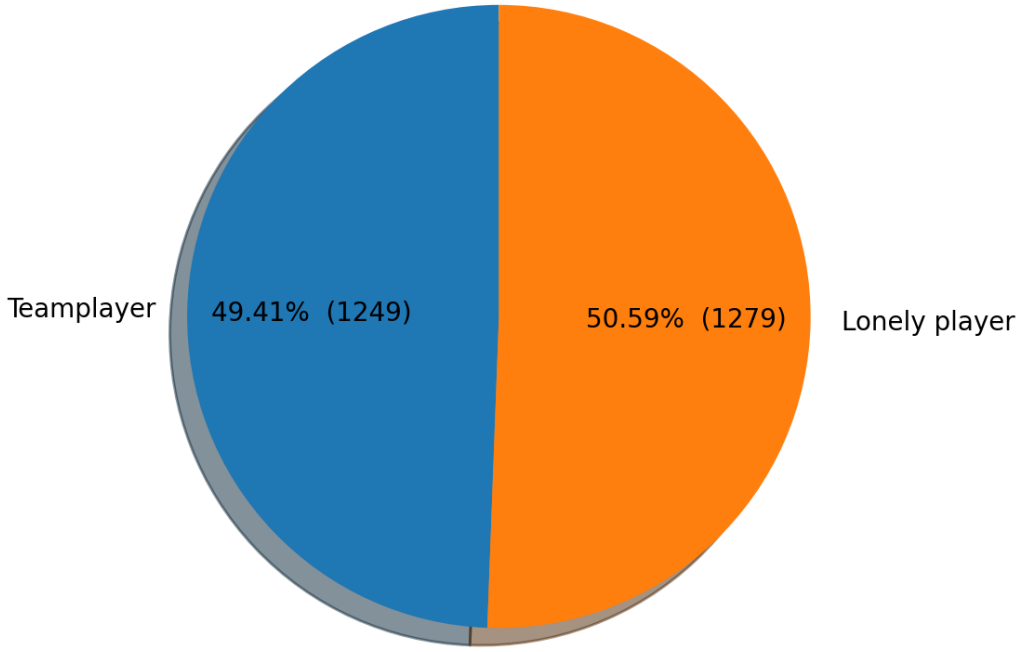

Figure 4.4: Pie chart representation of teamplayer and lonely players playstyle

A leader is the player who happens, most of the time, to be the decisive key to victory or the one who will highly contribute to the win, to the very least. To define this role, we have created our own feature called “kpa” which stands for “Kill participation”. This feature is calculated thanks to the events from a game, if a player has participated in a champions kill or a building kill he gains a point and at the end of the game we make a fraction kpa = playerKillParticipation / allKillParticipations. Moreover we also take into account the feature we used to define a trampler called “visionScore”. If both of these features are once again higher than the median we define this player as Leader and Follower on the contrary.

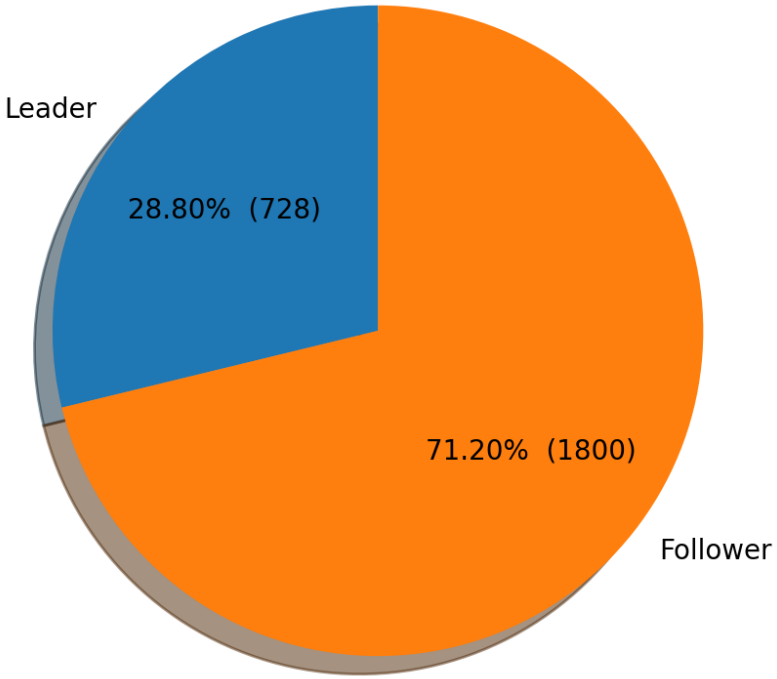

Figure 4.5: Pie chart representation of leader and follower players playstyle

Finally, we defined a playstyle which is harder to tell and define, “Resilient”. For this we took a feature called “gameDuration”, relative to the time spent in-game by one player. So, someone who has a bigger “gameDuration” is someone who wants to try everything to win.

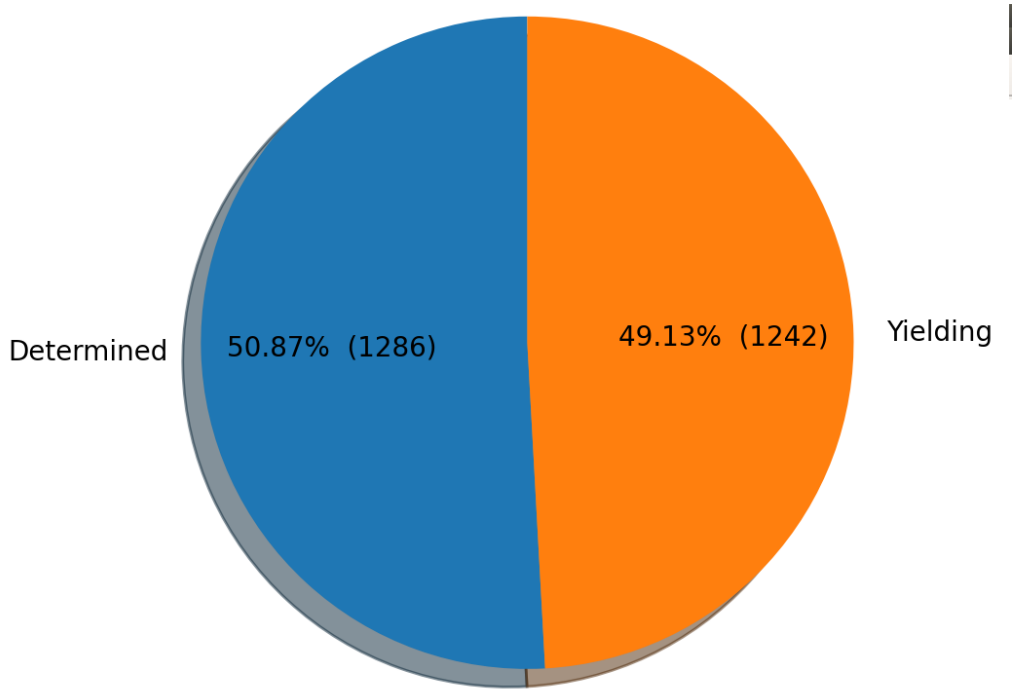

Figure 4.6: Pie chart representation of determined and yielding players playstyle

However, each situation is neither black or white, sometimes it's grey and a player is not always passive or aggressive, some players play a different way depending on each game and adapt their playstyles. Before we were counting the number of aggressive or passive players, this type of data is called discrete. Discrete data involve counting rather than measuring. In our case, measuring may be the best fit to make our predictions, this type of data is called continuous.

In our case, we will use continuous data and we will define the playstyle of each player using normal distribution from probability distribution. A probability distribution is a statistical function that describes the likelihood of obtaining the possible values that a random variable can take. The normal distribution density function accepts a data point along with a mean value and a standard deviation and throws a value which we call probability density.

First of all, we have to check if the data is following a normal distribution which is a shape of a bell curve. For each indicator we have to analyze the data. Each indicator is following a normal distribution like the "kpa" indicator. We can show that by using a histogram for the data, it will show the distribution of a variable.

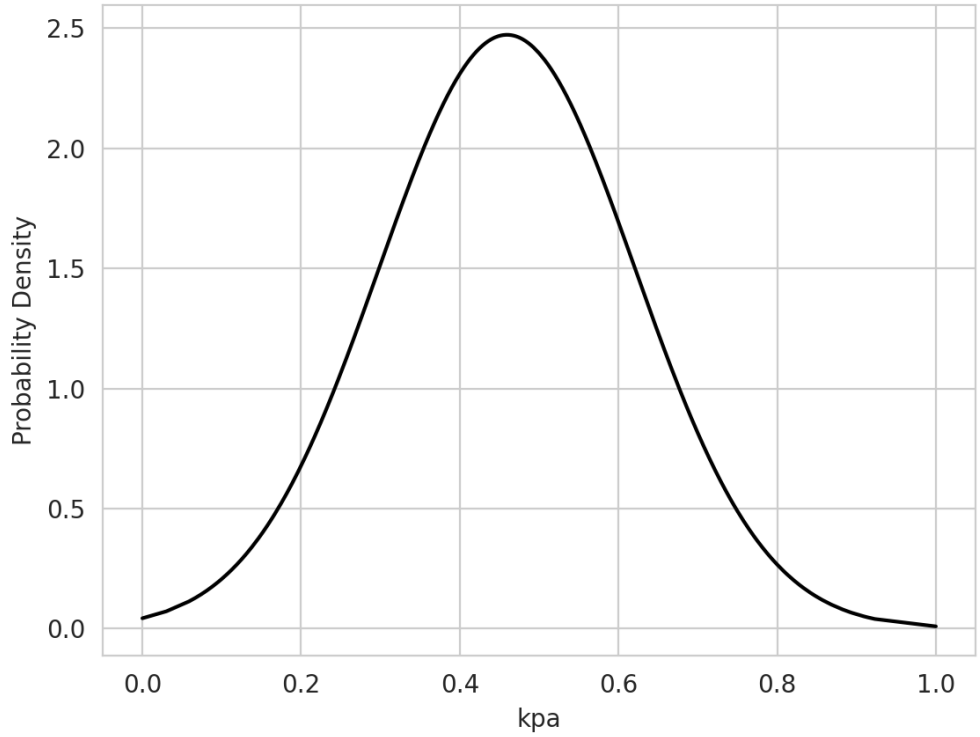

Figure 4.7: Histogram of the “kpa” indicator

Then as stated we have to compute the mean and the standard deviation. For the “kpa” indicator it shows this normal distribution density function.

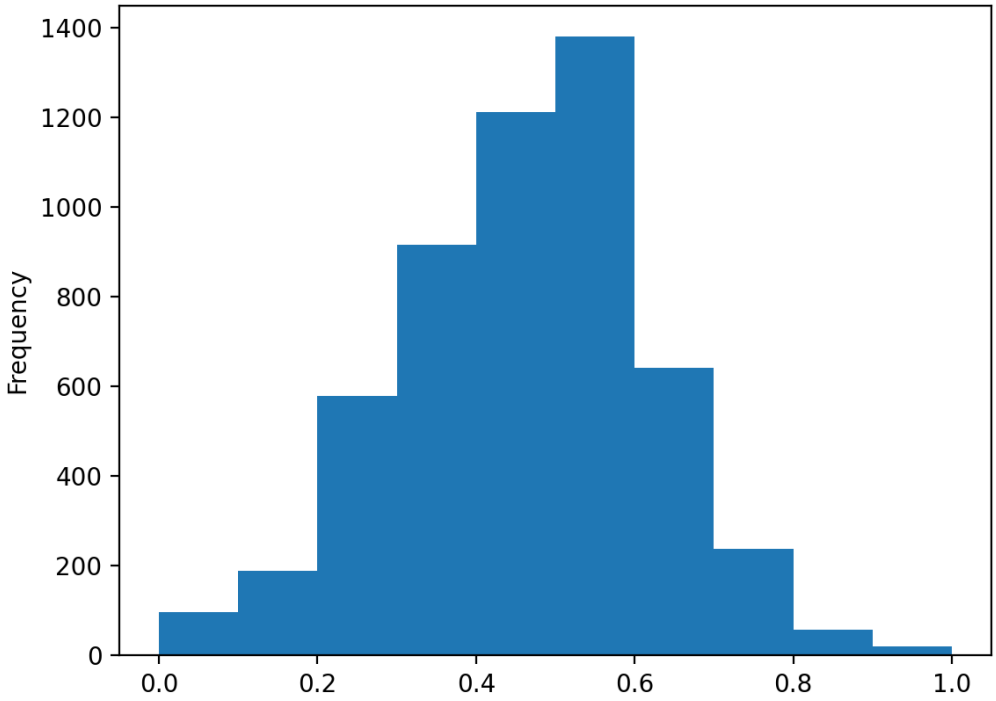

Figure 4.8: Normal distribution representation of the “kpa” indicator

Finally, we can find the probability by using the area under the curve from the density function. For each playstyle we do this on each indicator which is related to the playstyle, we can define the mean probability of this playstyle. For example, if we get a probability of 1 it means the player is very aggressive, 0.75 slightly aggressive, 0.5 normal, 0.25 slightly passive and finally 0 means passive. We use this process for each playstyle to find each probability for each player.

4.2.3 Formatting the data

We will use the continuous data because it makes more sense. So each player has 4 features and will be placed in the good role. So we end up with 41 features.

Once again, when our dataset is created we split in two parts, the training dataset with 70% of the main dataset and the 30% remaining for the test dataset. This ratio is working quite well and even advised by the machine learning model specialists. Indeed, each time we use a seed to have always the same dataset for the trainset and testset (seed=912).

4.3 Prediction Of Games Outcomes In League Of Legends From In-game And Player Playstyle Data

4.3.1 Formatting the data

For this part is we will take the two previous datasets, the in-game dataset and the player playstyle dataset. It will give more features to the machine learning models and it will maybe help it to define a certain pattern for the predictions. The dataset will be composed of 18 features per players and 181 features in total. Once more we will split our dataset in two splits with the seed=912.

Chapter 5

Experiments

We have run two experiments. The first one is to show which indicators are working the best for predicting the outcomes of a game of League of Legends. The second experiment is to define for the player playstyle what is the best between discrete and continuous data. For each experiment we will use the same environment, machine learning models and parameters for the classifiers.

Indeed, in our case we want to predict the outcome of a game of League of Legends. The two teams have their side, one called blue side and second one red side. In our case we will define the win or the loss of the blue side. Therefore, we will use machine learning techniques for the classification with only two outcomes 0 for win and lose for 1.

We have decided to test different machine learning models. As stated in the part 2.8, we will use the Logistic regression, Random Forest, K-nearest neighbors, Naive bayses. To train this model we have split out dataset to 70% to train the model, which is up to 370 games and the 30% remaining to test the model with 160 games. For each classifier we will use different parameters and see which one provides the best results.

5.1 Experiment One: Predictions Of Games Outcomes With Different Datasets

Through this section we will run the different machine learning algorithms from the different datasets we have created previously in the section 4. Our hypothesis was to find out if the addition of player playstyle with in-game dataset will improve the results from in-game data only. Furthermore, we will analyze the results we obtained with the one from the different papers we have analyzed in the literature review.

5.1.1 Results for prediction of games outcomes in league of legends from in-game data

Name	Parameters	Precision	Recall	F1-score	Accuracy
Logistic Regression	**max-iter=100 class-weight="balanced" solver="liblinear"**	**0.85**	**0.85**	**0.85**	**0.85**
Logistic Regression	**max-iter=10000 class-weight="balanced" solver="liblinear"**	**0.85**	**0.85**	**0.85**	**0.85**
Logistic Regression	max-iter=10000 class-weight="balanced" solver="sag"	0.83	0.83	0.83	0.83
Random Forest	max-iter=100 class-weight="balanced" random-state=42	0.79	0.79	0.79	0.79
Random Forest	n-estimators=100 class-weight="balanced" random-state=42	0.82	0.82	0.82	0.82
K Nearest Neighbors	n-neighbors=2 algorithm="ball-tree" weights="uniform"	0.70	0.70	0.70	0.70
K Nearest Neighbors	n-neighbors=2 algorithm="brute" weights="uniform"	0.58	0.55	0.53	0.58
K Nearest Neighbors	n-neighbors=2 algorithm="kd-tree" weights="uniform"	0.58	0.55	0.53	0.58
Naive bayes	Default	0.74	0.74	0.74	0.74

Table 5.1: Results from in-game data

With the in-game dataset, the machine learning algorithm which performs the best result is the Logistic regression with cross-validation (k=10). We have 85% of accuracy which is a honestly satisfactory when it comes to accuracy, which is not over-fitting. The other models are less effective, the random forest and logistic regression are close from 85% to 70% for the best, but the K Nearest Neighbors and the naive Bayes have more difficulties, from 74% to 58% of accuracy.

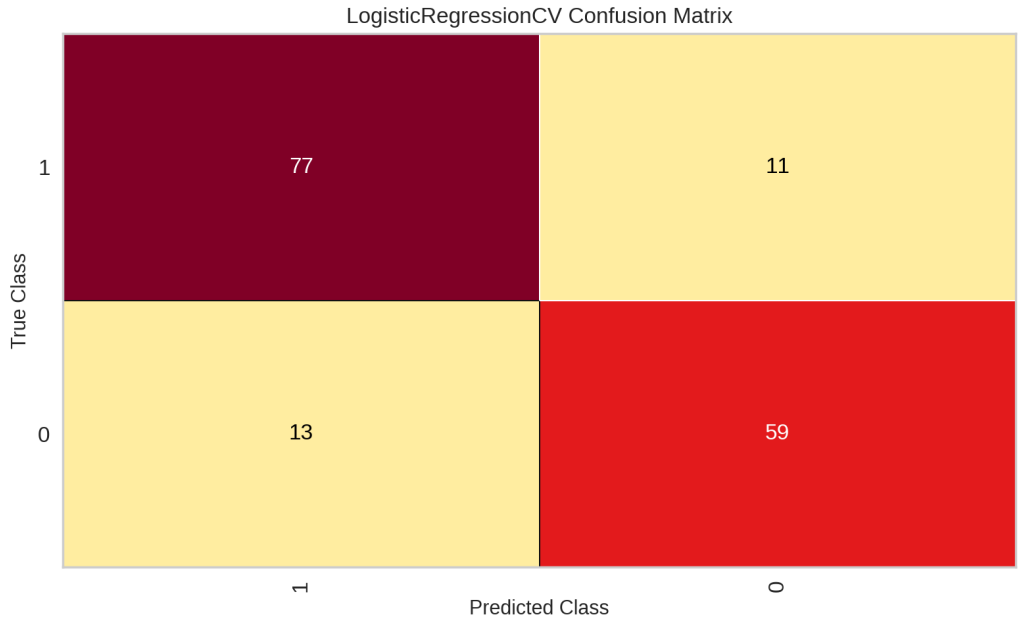

Figure 5.1: Matrix confusion for in-game data from Logistic regression

The confusion matrix demonstrates in total 11 games that were predicted 0 but ended been 1 and on the opposite, 13 which were predicted 1 but were 0. The fact that these wrongly classified games turned out to be actually close to each other 11 and 13, is also a good sign because we would have gotten a model which could be very powerful for predicting the win or loss but not the other one. We can then conclude that this model is both useful and trustful.

5.1.2 Results for prediction of games outcomes in league of legends from only player playstyle data

Name	Parameters	Precision	Recall	F1-score	Accuracy
Logistic Regression	**max-iter=100 class-weight="balanced" solver="liblinear"**	**0.73**	**0.73**	**0.72**	**0.72**
Logistic Regression	**max-iter=10000 class-weight="balanced" solver="liblinear"**	**0.73**	**0.73**	**0.72**	**0.72**
Logistic Regression	**max-iter=10000 class-weight="balanced" solver="sag"**	**0.74**	**0.74**	**0.73**	**0.72**
Random Forest	max-iter=100 class-weight="balanced" random-state=42	0.67	0.66	0.66	0.67
Random Forest	n-estimators=100 class-weight="balanced" random-state=42	0.68	0.68	0.68	0.69
K Nearest Neighbors	n-neighbors=2 algorithm="ball-tree" weights="uniform"	0.50	0.50	0.50	0.51
K Nearest Neighbors	n-neighbors=2 algorithm="brute" weights="uniform"	0.50	0.50	0.47	0.53
K Nearest Neighbors	n-neighbors=2 algorithm="kd-tree" weights="uniform"	0.50	0.50	0.47	0.53
Naive bayes	Default	0.68	0.68	0.68	0.68

Table 5.2: Results from player playstyle data

With the player playstyle dataset, the machine learning algorithm has some difficulties to predict the outcomes of the game compared to the in-game data predictions results. Nevertheless the results are not bad at all, there are only four features per player and it is therefore harder for the model to define a pattern to predict the winning team. The best prediction is given by the logistic regression with 72% accuracy.

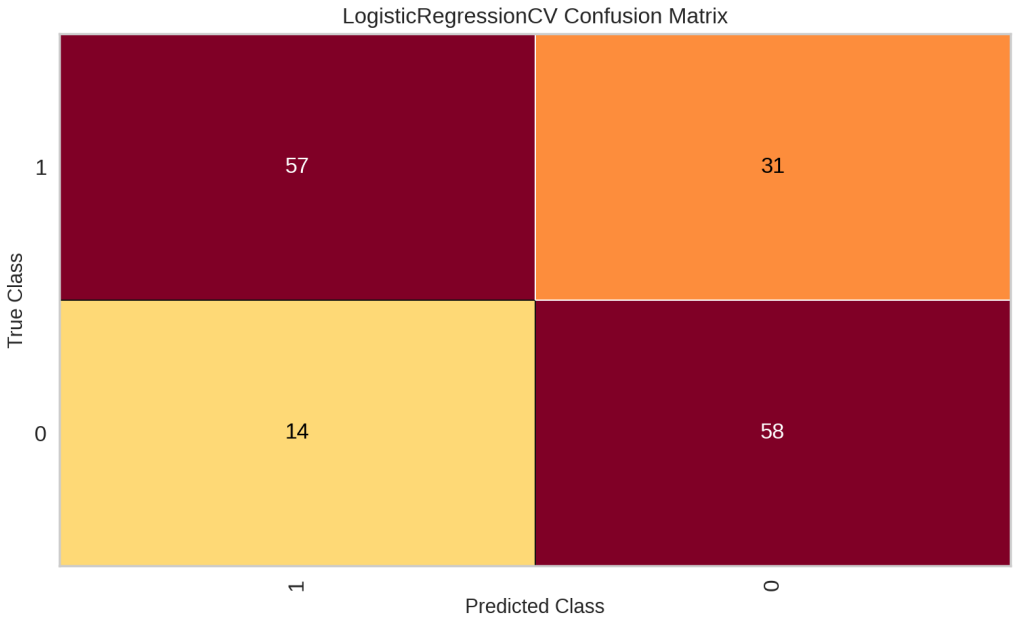

Figure 5.2: Matrix confusion for player playstyle data from Logistic regression

The confusion matrix demonstrates that there are in total 31 games that were predicted 0 but ended been 1 and on the opposite, 14 which were predicted 1 but were 0. We can see a small trend, and the model tends to predict more class 1. Nevertheless, it's still a good looking confusion matrix which is balanced with 57 true positives and 58 true negatives.

5.1.3 Results for prediction of games outcomes in league of legends from in-game data and player playstyle data

Name	Parameters	Precision	Recall	F1-score	Accuracy
Logistic Regression	**max-iter=100 class-weight="balanced" solver="liblinear"**	**0.85**	**0.85**	**0.85**	**0.85**
Logistic Regression	**max-iter=10000 class-weight="balanced" solver="liblinear"**	**0.85**	**0.85**	**0.85**	**0.85**
Logistic Regression	max-iter=10000 class-weight="balanced" solver="sag"	0.83	0.83	0.83	0.83
Random Forest	max-iter=100 class-weight="balanced" random-state=42	0.81	0.81	0.81	0.81
Random Forest	n-estimators=100 class-weight="balanced" random-state=42	0.82	0.82	0.82	0.82
K Nearest Neighbors	n-neighbors=2 algorithm="ball-tree" weights="uniform"	0.70	0.70	0.70	0.70
K Nearest Neighbors	n-neighbors=2 algorithm="brute" weights="uniform"	0.58	0.55	0.53	0.58
K Nearest Neighbors	n-neighbors=2 algorithm="kd-tree" weights="uniform"	0.58	0.55	0.53	0.58
Naive bayes	Default	0.73	0.74	0.73	0.73

Table 5.3: Results from in-game and player playstyle data

With the hybrid dataset, the machine learning algorithm which performs the best is the Logistic regression with cross-validation (k=10). We have 85% of accuracy, this is a undoubtedly good accuracy which is not over-fitting. The other models are also working well except for the K Nearest Neighbors with specific parameters. The accuracies are close to the in-game ones and in some cases slightly better.

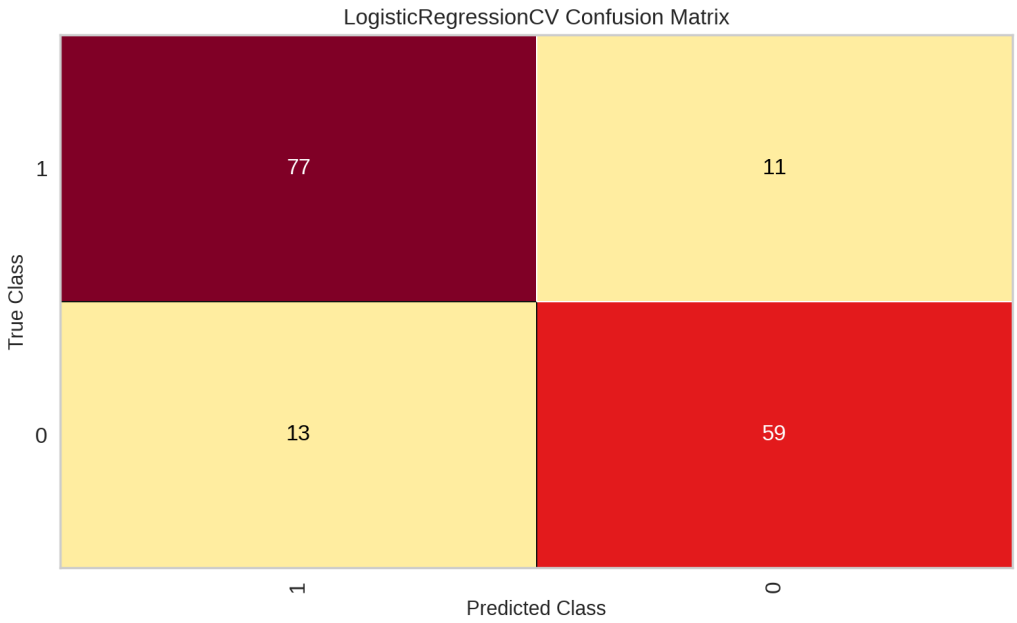

Figure 5.3: Matrix confusion for player playstyle data from Logistic Regression

The confusion matrix demonstrates that there is in total 13 games that were predicted 0 but ended been 1 and on the opposite, 11 which were predicted 1 but were 0. As the results from the models, this confusion matrix is the same as the in-game results. These results are good looking.

5.1.4 Comparison of results between experiment one and literature review

In this chapter we will compare the result from our experiment with the literature review and draw a conclusion from our experiment.

In this research we wanted to do prediction on one aspect of this game, the outcome of a game of League of Legends. Other papers made the same predictions but with different environments by using different machine learning models, different games or different indicators to perform their predictions. In addition, some of the papers tried different types of prediction, they tried two types of predictions, micro predictions and live predictions. Micro predictions are the ones which try to predict a particular event which is occurring during a game. The second one is live predictions, they try to predict an outcome of a game but while the game is played, this means the model is using pre-match data from players but also live data from the game they want to predict.

In this comparison we will try to focus on the papers which tried to predict the outcome of a game. Among them, they all achieved good results with their models from 60% to 80% of accuracy [20, 25, 39, 49]. Compared to the results we got, we have obtained similar results but we have slightly better results from 60% accuracy to 85%.

However, we have analyzed a trend among the papers. Most of them used the Logistic regression and obtained the best results at the same time. In our case it is the same, Logistic regression gives us the best results with up to 85% of accuracy.

The difference between the papers we have analyzed and our results can be caused by various factors. The main one may because we have all different datasets. The second factor can be caused by the machine learning models themselves, we have used different tools or technologies but also different parameters.

Nevertheless, in our case the results we have obtained from in-game data are only slightly better results from in-game data and player playstyle. But in-game has the biggest impact on the predictions outcome. Nevertheless, these show that player playstyle can be one of the elements that could improve the prediction of game outcome in the future if the playstyle data can be improved and this is what we tried to do in the second experiment.

5.2 Experiment Two: Player Playstyle, Discrete Data Versus Continuous Data

The second experiment was related to a hypothesis we have made previously in section 4.2. The hypothesis was, continuous data will be better than discrete data for the player playstyle dataset to predict the outcome of a game of League of Legends. We will compare the results between the ones found in section 5.1.2 - 5.1.3, and what we have found.

5.2.1 Results for prediction of games outcomes in league of legends from only player playstyle as discrete data

Name	Parameters	Precision	Recall	F1-score	Accuracy
Logistic Regression	max-iter=100 class-weight="balanced" solver="liblinear"	0.62	0.62	0.61	0.61
Logistic Regression	max-iter=10000 class-weight="balanced" solver="liblinear"	0.62	0.62	0.62	0.61
Logistic Regression	max-iter=10000 class-weight="balanced" solver="sag"	0.83	0.83	0.83	0.83
Random Forest	max-iter=100 class-weight="balanced" random-state=42	0.59	0.59	0.59	0.59
Random Forest	n-estimators=100 class-weight="balanced" random-state=42	0.60	0.60	0.60	0.60
K Nearest Neighbors	n-neighbors=2 algorithm="ball-tree" weights="uniform"	0.50	0.50	0.50	0.51
K Nearest Neighbors	n-neighbors=2 algorithm="brute" weights="uniform"	0.49	0.49	0.46	0.53
K Nearest Neighbors	n-neighbors=2 algorithm="kd-tree" weights="uniform"	0.49	0.49	0.46	0.53
Naive bayes	**Default**	**0.63**	**0.63**	**0.63**	**0.63**

Table 5.4: Results from player playstyle as discrete data

With the player playstyle dataset, the machine learning algorithm has some difficulties to predict the outcomes of the game. Nevertheless the results are not bad at all, there are only four features per player and it's harder for the model to define a pattern to predict the winning team. A lot of the models have close results from 58% to 63%, only the K Nearest Neighbors has poorer result. Surprisingly, the Naives Bayes gives us the best results with 63% of accuracy.

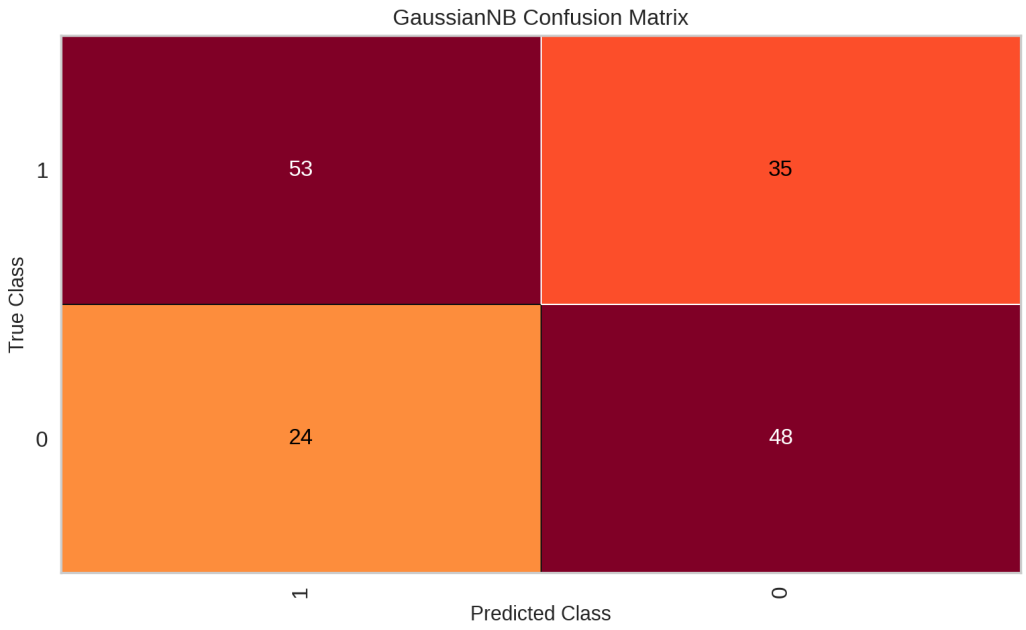

Figure 5.4: Matrix confusion for player playstyle as discrete data from naives Bayes

The confusion matrix demonstrates that there is in total 35 games that were predicted 0 but ended been 1 and on the opposite, 24 which were predicted 1 but were 0. These results are not the best.

Once again these results are not the best but this shows us the model is well balanced and not one sided to a class.

5.2.2 Results for prediction of games outcomes in league of legends from in-game data and player playstyle as discrete data

Name	Parameters	Precision	Recall	F1-score	Accuracy
Logistic Regression	**max-iter=100 class-weight="balanced" solver="liblinear"**	**0.84**	**0.85**	**0.84**	**0.84**
Logistic Regression	**max-iter=10000 class-weight="balanced" solver="liblinear"**	**0.84**	**0.84**	**0.84**	**0.84**
Logistic Regression	max-iter=10000 class-weight="balanced" solver="sag"	0.83	0.83	0.83	0.83
Random Forest	max-iter=100 class-weight="balanced" random-state=42	0.80	0.80	0.80	0.80
Random Forest	n-estimators=100 class-weight="balanced" random-state=42	0.80	0.80	0.80	0.80
K Nearest Neighbors	n-neighbors=2 algorithm="ball-tree" weights="uniform"	0.70	0.70	0.70	0.70
K Nearest Neighbors	n-neighbors=2 algorithm="brute" weights="uniform"	0.58	0.55	0.53	0.58
K Nearest Neighbors	n-neighbors=2 algorithm="kd-tree" weights="uniform"	0.58	0.55	0.53	0.58
Naive bayes	Default	0.75	0.75	0.74	0.74

Table 5.5: Results from in-game and player playstyle as discrete data

With the hybrid dataset, the machine learning algorithm which performs the best is the Logistic regression with cross-validation (k=10). We have 84% of accuracy this is a really good accuracy which is not over-fitting. The other models are also working well except for the K Nearest Neighbors with specific parameters.

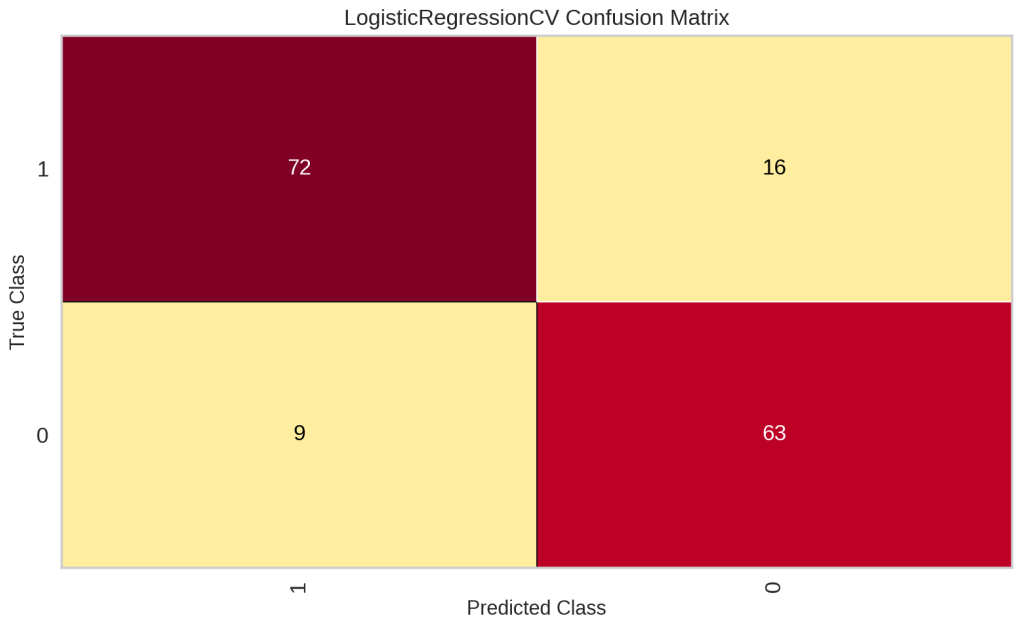

Figure 5.5: Matrix confusion for in-game data and player playstyle as discrete data from Logistic Regression

The confusion matrix demonstrates that there is it total 16 games that were predicted 0 but ended been 1 and on the opposite, 9 which were predicted 1 but were 0. We can see a small trend, and the model tends to predict more class 1. Nevertheless, it's still a good looking confusion matrix which is balanced.

5.2.3 Conclusion

The results confirmed our hypothesis. In both datasets using discrete data we did not improve the results we had using continuous data. This is very clear when we compare the results with the player playstyle only, we obtained 63% at best with the discrete data and 72% accuracy with continuous data. However it's closer when it comes to combine the data between the in-game data and player playstyle. With the continuous data we got 85% at best and 84% for the discrete one, it's a small improvement but thanks to the confusion matrix we can see that the repartitions between losses and wins is better distributed. In addition, when we confront the combination of in-game data and player playstyle (continuous) results and the in-game results we can see amongst most of the models a small improvement thanks to the addition of the player playstyle data (continuous).

Chapter 6

Conclusions Further Work

In this Msc project we presented and discussed an approach to apply machine learning techniques to predict the outcome of a game of League of Legends from different datasets. The first one is more conventional, it takes in-game data and tries to predict a winning team from the player statistics. Although we encountered some difficulties and we had to adapt by successfully creating our own heuristics algorithms to define the players roles. Thanks to those algorithms we sorted the players statistics according to the players roles. After that, we wanted to try a different approach of the problem and believed that taking into consideration the player playtsyle could help the machine learning model to predict a winning team. In the first place we had to create an algorithm that defines the playstyle for each player, then we created a dataset with only the playstyle of each player in each team and predicted the outcomes of the game. We indeed taught we would not have better results than the in-game data but we had satisfying results with 70%. Finally, we tried to combine the two datasets, and we predicted to have better results or at least close to the in-game data, as predicted we were successful and got really close to the results from the in-game data.

However, we opened a door and could even go further on the player playstyle analysis. Surely this side of the dissertation could be improved. Indeed, we were limited with the Riot's APIs. When a game is played each player has the ability to use a feature which is called "Ping". This "Ping" feature, are player-relayed alerts that provide gameplay information to the entire team. These can be used to communicate with your team faster than taking the time to type and breaking the flow of the game. This type of feature cannot be retrieved via Riot's API, that's why we could imagine using video analysis to analyze the behavior of each player thanks to the ping feature for example.

Figure 6.1: Ping feature in League of Legends

So we could use Riot's API to take the numeric data with the statistics of the players and combine that with the video analysis from the replay of the game. Alongside of this possibility we could have also used the player history for the Resilient playstyle, certainly if a player persists to play and is not afraid to lose a game it will be someone more resilient.

In this research we stuck to only 4 different family traits and having in total 16 distinct playstyles, but we could have created other ones. This game is deep, and players are all unique in a sense. From their experience, their feeling and entourage, all these factors can make a player change their playstyle.

In addition, we were limited to players playing the game in a conventional competitive game mode. We were successful to perform our mandatory tasks but we were unable to accomplish the optional tasks. Indeed, the professional games are happening in a private server called "Tournament server". The public API does not allow access to this data from the tournament server. Riot's API made this side of this API not free for the public, you need to ask special access and pay an expensive subscription. The games from this server vary from what you can find on the traditional server everyone has access to. The pro players during their pro matches are playing more with their teams and are taking

less risk in general. Nonetheless, every player has their own playstyle and every team has their playstyle too. But thanks to our research we were able to define, in the future, a new way to create teams. In fact, we could analyze what are the best combinations of playstyle and player style for a pro team and if a team is looking for a new player, analyze the games of potential players and say if this player which has a certain playstyle will fit in this team. On top of that, this project could be used in different area of E-sport, for example we could do the same on traditional sport like Football. In reality, analyzing the behavior of certain people and their way to do things can be represented by traits and then, we can predict interesting things.

References

[1] Shervine Amidi Afshine Amidi (2020). *CS 229 : Machine Learning tips and tricks cheatsheet*. URL: https://stanford.edu/~shervine/teaching/cs-229/cheatsheet-machine-learning-tips-and-tricks.

[2] Jai Bansal (2020). *How to Use Random Seeds Effectively*. URL: https://towardsdatascience.com/how-to-use-random-seeds-effectively-54a4cd855a79.

[3] Fandom company (2015). *League of Legends Wiki*. URL: https://leagueoflegends.fandom.com/wiki/League_of_Legends_Wiki.

[4] Edivandro Carlos Conforto and Daniel Capaldo Amaral (2010). "Evaluating an agile method for planning and controlling innovative projects". In: *Project Management Journal* 41.2, pp. 73–80.

[5] Kevin Conley and Daniel Perry (2013). "How does he saw me? a recommendation engine for picking heroes in dota 2". In: *Np, nd Web* 7.

[6] Michael Coram and Shawn Bohner (2005). "The impact of agile methods on software project management". In: *12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'05)*. IEEE, pp. 363–370.

[7] David Corne (2015). *Data Mining and Machine Learning*. URL: https://www.macs.hw.ac.uk/~dwcorne/Teaching/dmml.htm.

[8] Christoph Eggert et al. (2015). "Classification of player roles in the team-based multi-player game dota 2". In: *International Conference on Entertainment Computing*. Springer, pp. 112–125.

[9] A Famili et al. (1997). "Data preprocessing and intelligent data analysis". In: *Intelligent data analysis* 1.1, pp. 3–23.

[10] Antônio Alves Tôrres Fernandes et al. (2021). "Read this paper if you want to learn logistic regression". In: *Revista de Sociologia e Politica* 28.

[11] Guo Freeman and Donghee Yvette Wohn (2019). "Understanding eSports team formation and coordination". In: *Computer supported cooperative work (CSCW)* 28.1, pp. 95–126.

[12] Nando de Freitas (2014). *Machine Learning: 2014-2015 Course materials*. URL: https://www.cs.ox.ac.uk/people/nando.defreitas/machinelearning/.

[13] Johannes Furnkranz and Peter A Flach (2003). "An analysis of rule evaluation metrics". In: *Proceedings of the 20th international conference on machine learning (ICML-03)*, pp. 202–209.

[14] Weihao Gao, Sewoong Oh, and Pramod Viswanath (2018). "Demystifying fixed k-nearest neighbor information estimators". In: *IEEE Transactions on Information Theory* 64.8, pp. 5629–5661.

[15] García et al. (2016). "Big data preprocessing: methods and prospects". In: *Big Data Analytics* 1.1, pp. 1–22.

[16] Werner Geyser (2021). *The Incredible Growth of eSports [+ eSports Statistics]*. URL: https://influencermarketinghub.com/esports-stats/.

[17] Michael M Goldman and David P Hedlund (2020). "Rebooting content: Broadcasting sport and esports to homes during COVID-19". In: *International Journal of Sport Communication* 13.3, pp. 370–380.

[18] Max Sjöblom Hamari Juho (2017). "What is eSports and why do people watch it?" In: *Internet research*.

[19] KARIN HELLERSTEDT and HOWARD E ALDRICH (2008). "THE IMPACT OF INITIAL TEAM COMPOSITION AND PERFORMANCE ON TEAM DYNAMICS AND SURVIVAL." In: *Academy of Management proceedings*. Vol. 2008. 1. Academy of Management Briarcliff Manor, NY 10510, pp. 1–6.

[20] Victoria Hodge et al. (2017). "Win prediction in esports: Mixed-rank match prediction in multi-player online battle arena games". In: *arXiv preprint arXiv:1711.06498*.

[21] Mohammad Hossin and Md Nasir Sulaiman (2015). "A review on evaluation metrics for data classification evaluations". In: *International journal of data mining & knowledge management process* 5.2, p. 1.

[22] Seth E Jenny et al. (2017). "Virtual(ly) athletes: where eSports fit within the definition of "Sport"". In: *Quest* 69.1, pp. 1–18.

[23] Kalle Jonasson and Jesper Thiborg (2010). "Electronic sport and its impact on future sport". In: *Sport in society* 13.2, pp. 287–299.

[24] Adam Katona et al. (2019). "Time to die: Death prediction in dota 2 using deep learning". In: *2019 IEEE Conference on Games (CoG)*. IEEE, pp. 1–8.

[25] Nicholas Kinkade, L Jolla, and K Lim (2015). "Dota 2 win prediction". In: *Univ Calif* 1, pp. 1–13.

[26] Gilles Louppe (2014). "Understanding random forests: From theory to practice". In: *arXiv preprint arXiv:1407.7502*.

[27] Ethan May (2020). *Streamlabs & Stream Hatchet Q1 2020 Live Streaming Industry Report*. URL: https://blog.streamlabs.com/streamlabs-stream-hatchet-q1-2020-live-streaming-industry-report-9630bc3e0e1e.

[28] Christopher McCutcheon, Michael Hitchens, and Anders Drachen (2017). "eSport vs irlSport". In: *International Conference on Advances in Computer Entertainment*. Springer, pp. 531–542.

[29] Jarernsri Mitrpanont et al. (2017). "A study on using Python vs Weka on dialysis data analysis". In: *2017 2nd International Conference on Information Technology (INCIT)*. IEEE, pp. 1–6.

[30] Hao Yi Ong, Sunil Deolalikar, and Mark Peng (2015). "Player behavior and optimal team composition for online multiplayer games". In: *arXiv preprint arXiv:1503.02230*.

[31] Anthony D Pizzo et al. (2018). "eSport vs. sport: A comparison of spectator motives." In: *Sport Marketing Quarterly* 27.2.

[32] Matthew A Pluss et al. (2019). "Esports: the chess of the 21st century". In: *Frontiers in psychology* 10, p. 156.

[33] Nazneen Fatema Rajani et al. (2020). "Explaining and improving model behavior with k nearest neighbor representations". In: *arXiv preprint arXiv:2010.09030*.

[34] Inc. Riot Games (2021). *Riot Games API*. URL: https://developer.riotgames.com/.

[35] Irina Rish et al. (2001). "An empirical study of the naive Bayes classifier". In: *IJCAI 2001 workshop on empirical methods in artificial intelligence*. Vol. 3. 22, pp. 41–46.

[36] Mariona Rosell Llorens (2017). "eSport gaming: the rise of a new sports practice". In: *Sport, Ethics and Philosophy* 11.4, pp. 464–476.

[37] Matthias Schubert, Anders Drachen, and Tobias Mahlmann (2016). "Esports analytics through encounter detection". In: *MIT Sloan Sports Analytics Conference*. MIT Sloan.

[38] Erwan Scornet, Gerard Biau, and Jean-Philippe Vert (2015). "Consistency of random forests". In: *The Annals of Statistics* 43.4, pp. 1716–1741.

[39] Kuangyan Song, Tianyi Zhang, and Chao Ma (2015). "Predicting the winning side of DotA2". In: *Sl: sn*.

[40] Sandro Sperandei (2014). "Understanding logistic regression analysis". In: *Biochemia medica* 24.1, pp. 12–18.

[41] Spezzy (2021). *How many people play League of Legends? – UPDATED 2021*. URL: https://leaguefeed.net/did-you-know-total-league-of-legends-player-count-updated/.

[42] Harlan E Spotts and Anthony F Chelte (2005). "Evaluating the effects of team composition and performance environment on team performance". In: *Journal of Behavioral and Applied Management* 6.2, pp. 127–140.

[43] Adam Summerville, Michael Cook, and Ben Steenhuisen (2016). "Draft-analysis of the ancients: predicting draft picks in dota 2 using machine learning". In: *Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference*.

[44] Paul Tassi (2014). "Riot's' League of Legends' reveals astonishing 27 million daily players, 67 million monthly". In: *Retrieved October* 30, p. 2014.

[45] Elizabeth SV Tudor (2019). "The Emergence of eSport During Covid-19: How Sim Racing Replaced Live Motorsport in 2020". In: *Journal of Motorsport Culture & History* 1.1, p. 8.

[46] Stanford University (2020). *CS 229: Machine learning course notes*. URL: http://cs229.stanford.edu/syllabus.html.

[47] Christina Voskoglou (2017). "What is the best programming language for Machine Learning". In: *Towards Data Science* 5.

[48] Brad Warner and Manavendra Misra (1996). "Understanding neural networks as statistical tools". In: *The american statistician* 50.4, pp. 284–293.

[49] Yifan Yang, Tian Qin, and Yu-Heng Lei (2016). "Real-time esports match result prediction". In: *arXiv preprint arXiv:1701.03162*.

[50] Jason Yosinski et al. (2015). "Understanding neural networks through deep visualization". In: *arXiv preprint arXiv:1506.06579*.

[51] Cheng Yu, Wan-ning Zhu, and Yu-meng Sun (2019). "E-sports ban/pick prediction based on bi-lstm meta learning network". In: *International Conference on Artificial Intelligence and Security*. Springer, pp. 97–105.

[52] Harry Zhang (2004). "The optimality of naive Bayes". In: *AA* 1.2, p. 3.

Appendix A

Professional, Legal, Ethical And Social Issues

A.1 Professional Issues

The research project is well documented and respect code standards. Indeed, we respect policies and cite any third party sources. We test and are critic about the results to prove the serious of the research and scientific paper.

A.2 Legal Issues

The project is respecting the legal policies from Riot's API. When players sign up for the game players give consent for anonymous use of their gameplay data through the API. The only potentially identifiable links are the player nicknames which are anonymised. In addition, the data we are working with shall not be redistributed to anyone.

A.3 Ethical Issues

We will make sure that there are no privacy breaches from the data. In addition, we will respect the Heriot-watt policies and of course the GDPR rules. The dataset doesn't harm anyone and is public if you have access to an API Riot's key.

A.4 Social Issues

We respect the policies from Riot's API and any personal data is anonymized, as per the agreement of the player, when they have signed up to the game.