Master Thesis - Design and Implementation of a football data analysis application

Thesis · August 2021

CITATIONS
0

READS
2,395

1 author:

Omar El Yousfi
Université Mohammed Premier
1 PUBLICATION **0** CITATIONS

SEE PROFILE

University Mohammed First
Multidisciplinary Faculty of Nador
Computer Science department
Nador - Morocco

Master Thesis

Submitted in Partial Fulfillment of the Requirements for the Degree of
Master in Data Science and Intelligent Systems

By

Omar El Yousfi

Design and Implementation of a Football Data Analysis Application

---

Sustained publicly on July 26, 2021 in front of the jury:

Pr. Redouane ESBAI	**ENCG, University Mohammed First, Oujda**	**Chairman, Supervisor**
Pr. Mouncef FILALI BOUAMI	**FPD, University Mohammed First, Nador**	**Examiner**
Pr. Khalid EL MAKKAOUI	**FPD, University Mohammed First, Nador**	**Examiner**
Pr. Mostapha BADRI	**FPD, University Mohammed First, Nador**	**Examiner**

Academic year: 2020−2021

Acknowledgement

The Prophet ﷺ said: ❝He who does not thank the people is not thankful to Allah. ❞ [Sunan Abi Dawud, graded Sahih by Shaikh al-Albani.]

First of all, I want to thank Allah for the strength, good health and the ability to surpass the difficulties I faced during my studies journey.

I would like to express my deep gratitude to my research supervisor, professor Redouane Esbai, who despite of his busy schedule, took the time to supervise and guide me throughout my research stages. I was honored to have had the chance to work and study under his guidance. I would also like to thank professor Mohamed Bellouki for his efforts and dedication in the last two years to provide us with a great master program.

I am thankful and grateful to my family for their love and support during all these years, from the moment I was born to the time I started working on my masters' graduation research thesis and for the years to come. Special thanks to every professor that ever taught me, from my first day at school till my last.

I cannot let this opportunity pass without expressing my deepest thanks to those who had helped and provided me with assistance when I needed it the most.

Last but not least, I would love to thank the members of the jury, professor Mouncef Filali Bouami, professor Khalid El Makkaoui and professor Mostapha Badri for agreeing to evaluate my work; that is an honor.

Abstract

Football is the most popular sport in the world, watched by kids, teenagers and adults. As Football becomes more popular by the minute, teams are constantly looking for new creative ways to improve their performance and compete on a higher level.

This research aims to create an application that provides analysis of football based on event data collected through matches. To achieve our goal, we would start by defining the research thesis, extracting requirements and studying the subject. Phase of design follows us to develop a set of UML diagrams. Finally, we create the application using Python language and implement a machine learning model that would predict the probability of a shot being a goal.

Keywords: Football Analytics, Machine Learning, Data Analysis

Résumé

Le football est le sport le plus populaire au monde, observé par les enfants, les adolescents et les adultes. Comme le football devient de plus en plus populaire à la minute, les équipes sont constamment à la recherche de nouveaux moyens créatifs pour améliorer leurs performances et de rivaliser à un niveau supérieur.

L'objectif de cette recherche est de créer une application qui fournit une analyse du football basée sur les données d'événements collectées lors des matchs. Pour atteindre notre objectif, nous commencerions par définir la thèse, extraire les exigences et étudier le sujet proposé. Suivie d'une phase de conception pour développer un ensemble de diagrammes UML. Enfin, nous créerons l'application en utilisant python et implémenterons un modèle d'apprentissage automatique qui prédirait la probabilité qu'un tir soit un but.

Mots-clés : Football analytique, Apprentissage automatique, Analyse de données.

Table of Contents

List of figures:

List of Tables

General Introduction

In order to validate my studies acquired over two years in the poly-disciplinary faculty of Nador, and for the sake of obtaining my Data Science & Intelligent Systems Master Diploma, I have worked on my research thesis for a duration of 4 months (April 2021 - July 2021).

Football is the most popular sport in the world, watched by kids, teenagers and adults. As Football becomes more popular by the minute, teams are constantly looking for new creative ways to improve their performance and compete on a higher level.

After every match in every competition, media football analysts and fans start analyzing the match. We find communities that analyze football matches in every online platform, like forums, social media and Reddit. Meanwhile these matches analysis are based on individual's opinions; data allows us to analyze these matches from the numbers perspective.

Clubs at every level of the football pyramid are becoming more intelligent and more efficient thanks to data. In the early years of the last decade, football analytics was born, parallel with the rise of data science and machine learning. Over the last 10-15 years, football clubs have had to deal with a technological revolution which was followed by the start of collecting data through third party vendors. That data was primarily collected for fans and media outlets to use, until they have made their way into the clubs themselves.

In recent years, teams started hiring data scientists and building data departments dedicated to developing new ways of understanding and monitoring football matches.
Our research thesis, we will create an application that provides match analysis based on sample data from StatsBomb. With this application, it is easier for managers, players, coaching staff and even fans, to analyze their football team without the need of assistance of a specialist. To make it easier to navigate, we will create an interface that executes analysis codes and displays the results in the same place.

Our thesis is about a football data analysis application dedicated to matches analysis, teams' analysis and player analysis. We will be using events data collected during football matches. In every football match, thousands of events occur, starting from simple passes, to shots, bookings and fouls. Therefore, to create event data, trackers record every move on the pitch during the match, including the position and timestamp of every event.

Many teams are now recording data from thousands of actions during games and training sessions to help shape pre-match preparation and post-game debriefs, pinpoint transfer targets and develop young talent. Our application will be dedicated to pre-match preparation and post-game debriefs, it is important to create an application that provides good analysis that is easy to understand by coaching staff. The objective of our application is to implement these functionalities in a desktop application:

- Detailed analysis of different events of the match (Match analysis).
- Detailed analysis of each player (Player analysis)
- General Information about the match.
- Team analysis.

This report will discuss the objectives, the tools and algorithms used and the results obtained. The report is divided into five chapters in addition to a general introduction, general conclusion and fourteen appendices:

- **The first chapter** is dedicated to presenting analytics on football, and talking in detail about its history, applications and purposes.
- **The second chapter** is devoted to presenting the process of development of the application.
- **The third chapter** is dedicated to the specification and modeling of the application, using UML diagrams and presenting the overall architecture of the application alongside the data architecture.
- **The fourth chapter** presents the different tools used to implement our application, the visualizations used in the analysis and the implementation of the application.
- **The fifth chapter** is a demonstration of the results and the implementation of the application.

Chapter 1: Analytics in Professional Football

1. Introduction

Before diving into football analytics, we should first define what data analytics is? What are the types of data analytics? What is it used for? And then we move to define analytics in football, its history, its importance and its applications.

2. Data Analytics

Data analytics is a strategy used by businesses and industries to gain a competitive edge and be the first to make a profitable decision. Data analytics helps in understanding the state of the business and predicts the future outcomes.

Any data can be subjected to data analytics in order to obtain information and conclusions from that data. Data analysis is not just a step that an analyst implements for the purpose of extracting information; it is, in fact, a process of multiple steps:

- The first step is determining data requirements, such as the type of data and how the data is grouped.
- The second step is collecting data. Data collection can be done with various techniques such as surveys, online sourcing, cameras, sensors, GPS, or through personnel.
- The third step is organizing data on a spreadsheet or any other form of software that can take statistical data.
- The last step is cleaning data. In order to create dashboards or machine learning models, data should not have duplicates, incomplete data, missing data, wrong data types, etc.

2.1. Data Analytics Components

There are multiple methods that can be used to process any set of data, such as:

- Data mining can identify anomalies in different groups of data and extract information and conclusions. Data mining is used for determining patterns.

- Natural language processing is used for text analytics; one of the most famous applications of text analytics is the auto-correct we have in our phones.
- Data visualization is the graphical representation of data; it is used to understand trends, outliers and patterns in data. In this application we will use multiple visualizations to understand the players' performance and teams in general.
- Business Intelligence is a technology process for analyzing data, it involves transforming data into insights that help business owners and managers make decisions.

2.2. Types of Data Analytics

After gathering, organizing and cleaning data, then comes the step of analyzing it. There are four types of analytics, each type answers a certain type of question:

- Descriptive analytics answers the question "What happened?" or "What is happening?". As the names states, this type of analytics describes what has happened, and unlike other methods of analysis, it is not used to draw inferences or predictions from its findings.
- Diagnostic analytics answers the question "Why did it happen?". It is used to understand why a certain phenomenon has happened and search for valuable insights in the data.
- Predictive analytics answers the question "What will happen and when?". It makes predictions using historical data combined with statistical modeling. Many companies use predictive analytics to find patterns in data to identify risks and opportunities. In our application, we will use machine learning to create a model that will predict the expected goals a team would score based on shots and goals data from previous seasons.
- Prescriptive analytics is used to find the best course of action for a given situation. Prescriptive analytics can also suggest decision options for taking advantage of a future opportunity or avoiding a future risk and displaying the implications of each decision option.

3. Introduction to Analytics in Football

Football analytics became a center of interest of the biggest football clubs worldwide, the added value that analyzing the data obtained from matches gives one step ahead of the competition. Now managers can easily understand how a certain player has performed, and get insights about the overall

performance of the team or the opposition. Even the most basic stats, like the number of shots, shots on target, ball possession, fouls, can give an insight on how two teams have performed.

Wyscout: "Football analytics is attracting an increasing interest of academia and industry, thanks to the availability of sensing technologies that provide high-fidelity data streams extracted from every match". [1]

4. History of Football Analytics

We often think that football analytics has been discovered or adopted only in the last decade and started being used by football clubs, which has some truth to it, but actually the first person that collected event data from matches and used it to analyze the performance of the teams was an English accountant in the British Royal Air Force named *Charles Reep*, his discoveries influenced English football for many years from the World War Two until the eighties.

Charles collected data and analyzed it; he concluded that most goals in the English football were scored from three passes or less, so teams started adopting this theory: Long ball technique, which insists on getting the ball forward as soon as possible, which is also known as direct attacking style. *Reep* worked with many English teams such as Wolverhampton, Brentford and Sheffield Wednesday.

We can say that *Charles* was the first person to use analytics in football, even though his conclusions were proven inaccurate later by *Jonathan Wilson*, highlighting the importance of interpreting data.

We talked about how *Charles* was the first person to use data to make conclusions, but who was the first person to be known to make a huge impact using data analytics? Not in football actually, *Billy Beane*, general manager of *Oakland Athletics*, a baseball team in the United States that was suffering from debt and bad results, then *Billy* stepped up and replaced scouts by using data to recruit players that were cheap and nobody wanted, as a result *Billy Bean* created one of the strongest teams in the Baseball history as they won 20 consecutive games back in 2002. *Michael Lewis* wrote a book called "*Moneyball: the art of winning an unfair game*" in 2003 narrating the story of success of Billy Beane, followed by the *Moneyball* movie starring *Brad Pitt* and *Jonah Hill* which was critical in popularizing the concept of sports analytics.

At this level, we know that football analytics can help teams improve performance, but is it really adopted in today's football? *Jan Van Haaren*, Chief Analytics Officer at SciSports answers this question: "The clubs that have bought into match data analysis can broadly be divided into two categories. On one hand, a select number of clubs have invested in in-house data analysis or data analytics departments. Clubs such as Midtjylland and Brentford have received widespread media attention for using data in their day-to-day operations. Clubs such as Liverpool, Barcelona, Manchester City and Leicester City have hired talented data scientists to carry out longer-term research projects… this list of clubs is likely still much longer, but many clubs prefer to remain secretive about their hires in this area hoping to gain a competitive advantage." [2]

5. The importance of good analysis

Collecting data alone is not enough. Today the deal is not so much about collecting the data, but about making sense of the data. data needs to be analyzed in order to obtain insights that would help in the decision making process. Data analytics helps businesses in every industry to make sense of the tremendous amount of data they generate. Football has actually been collecting the most data for the longest time. Football is different from other sports like Baseball, Basketball or even American Footbal. Basketball is high scoring, Tennis, American football and Baseball are segmented, meanwhile Football is a continuous sport, low-scoring and depends on various aspects that affect the result, which arises the problem of analysis, even with the best data, without a good analysis, the conclusions could be meaningless. The following figure is a good representation of the importance of analytics by Imperial College London:

Figure 1 Importance of analytics

6. Applications of data analytics in football

The number of potential applications of data analytics in football grows everyday with the exponential growth of data collection technologies, each team uses what is possible, affordable and what is best for them. We can summarize the use of data analytics in three points:

6.1. Performance Analysis

Performance analysis is the main focus of our thesis. Improving performance is the goal of every football team, the right performance analysis can help build a winning strategy and makes it possible for the manager to explain to players what they did wrong in previous matches, and easily explain what is required from them to do in order to win, score goals and prevent receiving goals. Performance analysis is not limited only to analyzing the team's performance, but also can be used to analyze the opponent and discover their weaknesses.

To better understand the effect of performance analysis, let us look at this example of the most common shot locations in the NBA 2001/2002 season and NBA 2016/2017 season:

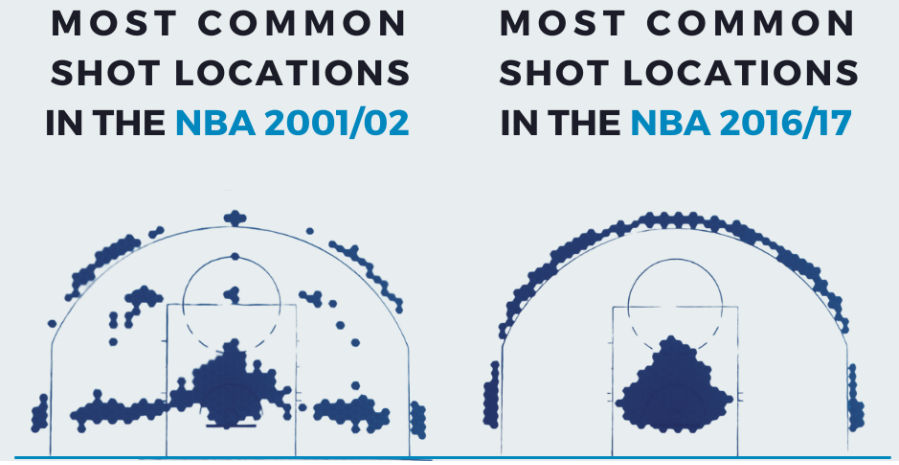


Figure 2 Shot locations in the NBA

In the help of data, we can see that how risk reward considerations have dramatically increased. It is expected that in the next 10-15 years we will see something similar in football. In fact, it started showing this effect if we analyze Barcelona shots in the last 10 years:

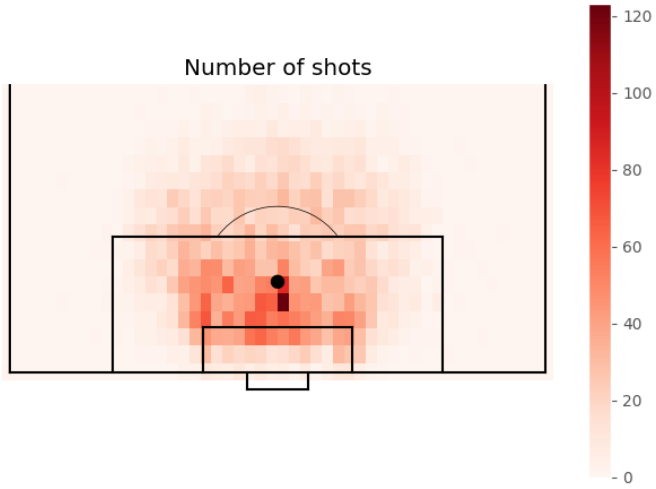


Figure 3 Barcelona shot locations

We can see that most shots were close to the penalty area and most of the the shots have wide angle. (This figure was obtained from our application using StatsBomb data).

This analysis opens a new door; calculating Expected Goals (xG).

6.1.1. Expected Goals

Soccerment definition: "An "Expected Goal" (or "xG") represents the expected value from a shot, i.e. the probability of that shot becoming a goal. Calculating such probability depends on a number of factors, including distance from goal, the angle of the shot, the body part the shot was taken with (head, strong or weak foot), the playing situation (open play, counterattack), and others (whether there were any opponents blocking the space ahead, for example)." [3]

6.1.2. Expected Assists

Beside the expected goals metric, we find expected assists, which highlights the performance of a player in terms of probability a created chance being converted into goals.

Soccerment: "Very skilled creators whose teammates have not been too good at finishing can, with the help of Expected Assists, be recognized for their efforts." [3]

6.2. Scouting

Data analytics were heavily used for scouting by *Billy Beane* in Baseball and is currently used by all teams in MLB, can it be used for football? The answer is yes!

Analyzing the performance of thousands of players by scouts is impossible. Meanwhile, the data collected can easily highlight players with good performance and discover hidden gems. Why is that? Computers remember more than we do, we humans cannot remember all the shots, passes, moves made by a player, but computers can. Computers analyze thousands matches, a human can analyze only one match per time.

One of the pioneers of scouting in football using data is *Matthew Benham*, the owner of Brentford FC and FC Midtjylland. These two teams are not the biggest teams in their leagues and definitely not the richest, but they are able to fight in divisions higher than their weight thanks to data.In the figure bellow we present a table taken from the book "*The expected Goals Philosophy*" by *James Tippet*, it presents the deals made by Brentford:

Player	Purchasing fee (£ m)	Selling fee (£ m)	Profit (£ m)
N. Maupay	1.8	20	18.2
A. Gray	0.5	12	11.5
S. Hogan	0.75	12	11.25
C. Mepham	0	11	11
E. Konsa	2.5	12	9.5
R. Woods	1	6.5	5.5
N. Yennaris	0.2	5	4.8
Jota	1.5	6	4.5
J. Tarkowski	0.3	4.5	4.2
J. Egan	0.4	4	3.6
D. Bentley	0.45	4	3.55
R. Sawyers	0.3	2.9	2.6
M. Odubajo	1	3.5	2.5
M. Colin	0.9	3	2.1
F. Jozefzoon	0.9	2.8	1.9
Total	**12.5**	**109.2**	**96.7**

Figure 4 Brentford deals

We can see that Brentford made 96.7 million pounds' profit by buying cheap players and selling them for higher fee. This strategy of buying cheap talented players and selling them for higher fees is widespread among small teams and middle table teams with limited budgets, meanwhile big teams use scouting using data to invest in players that are predicted to perform better in next years and are suited for the play style of the team. Even if those players are expensive, the best example of this is Liverpool that bought expensive players like Mohamed Salah, Van Djik, and Alisson. Besides the trophies that Liverpool has won in the past few years thanks to these singings, the t-shirts sales have increased dramatically alongside an increase in number of fans over the whole world. Also the estimated prices for these players currently are much higher than the fee that Liverpool has paid for them.

6.3. Youth player development and injury prevention

6.3.1. Youth player development

Many teams use data to develop young talents, each team aims for a different goal but they all use the same ways: Use data to better understand the weaknesses and strength of the young players, focus on developing their weaknesses and improve their strengths.

Small teams and middle table teams that develop young players aim to sell them for high fees such as Premier League team Southampton, meanwhile big teams like Real Madrid, Fc Barcelona and Manchester City tend to develop players that one day can take the lead and perform in the first team instead of buying expensive players from other teams.

6.3.2. Injury prevention

A famous example is MilanLab, that was founded in 2002 in order to reduce the risk of Ac Milan players getting injured. The aim of MilanLab is to optimize the team's results by acting as a technological support structure for the decision making process. Ac Milan defines MilanLab as the following: "With the health of individual players in mind, MilanLab studies and identifies the guidelines relating to the optimal management of athletes. Prevention and a systemic approach are the key principles on which the MilanLab structure is based. Prevention, understood as a series of actions taken to protect against a future ailment, is accompanied by a systemic approach that refers

to a widely accepted concept." [4] This side of sports analytics is out the scope of our research thesis, it is mainly focused on extracting data about players through GPS worn by players during training sessions and matches.

7. Football data

Unfortunately, these data are owned by specialized companies and are rarely publicly available for scientific research. Hence, some data providers made samples of data that can be used for research purposes, and thanks to them working on a football analytics project like ours is possible. Some data providers that made sample data available for research purposes are: StatsBomb, Wyscout and Metrica. The data used in this project contains all the spatio-temporal events (passes, shots, fouls, etc.) that occur during a match. A match event contains information about its position, time, outcome, player and characteristics.

One big problematic is that football is the most complex sport: It is low scoring, continuous, time varying, very strategic and very subjective which makes it difficult to create meaningful analysis. Two people analyzing the same game could come up with different opinions, unlike other sports.
It is very easy to do an analysis of other sports like Baseball, Tennis or Basketball unlike Football. Therefore, the big challenge is to create analysis that translates what happens in the matches facing the problem of continuous flow of data and its subjectivity.
Patrick Lucey (Director of Data Science, STATS): *"The key for football is actually to come up with the right language and ask the right questions for specific things."* [5]
Another problem is the need of data scientist to write code and analyze matches. Hence the need to develop not only code that displays plots and tables, but an application that automates the process and generates meaningful analysis for each match with no interference of a data analyst, which will allow every team coach and staff members around the world to analyze their matches.

8. Conclusion

In this chapter we dealt about the birth of analytics in football and how it affected sports, especially football, and we highlighted the importance of analytics in football and how gathering data alone is not enough. Finally, we studied the different applications of analytics in professional football and how teams use them to improve their performance.

Chapter 2: Development process

1. Introduction

In every project, the planning stage is very important, it must be carried out with such care because it will decide the course of the project as a whole afterwards.

2. Operational planning

- Tasks table: The following table presents the organization of the various tasks involved in this project:

	Task	From	To	Duration
A	Choosing the subject and studying the existing similar projects.	01 April	08 April	7 days
B	Looking for data to use in the project.	09 April	16 April	7 days
C	Study the data by creating multiple analysis using Python libraries.	17 April	30 April	13 days
D	Application specification and design.	01 May	05 May	4 days
E	Backend: Cleaning data and creating analysis code	03 May	10 June	38 days
F	Front End: Creating an interface using Tkinter.	01 June	30 June	29 days
G	Writing the research thesis	02 June	20 July	48 days

Figure 5 Tasks table

- GANTT chart: The GANTT chart is a project management tool; it is one of the most effective tools to visually represent the progress of the various tasks that constitute a project. A GANTT chart lists all tasks to be performed and indicates the date on which these tasks are to be performed.

From the table of tasks, we have deduced the diagram of GANTT of our project represented in the following figure:

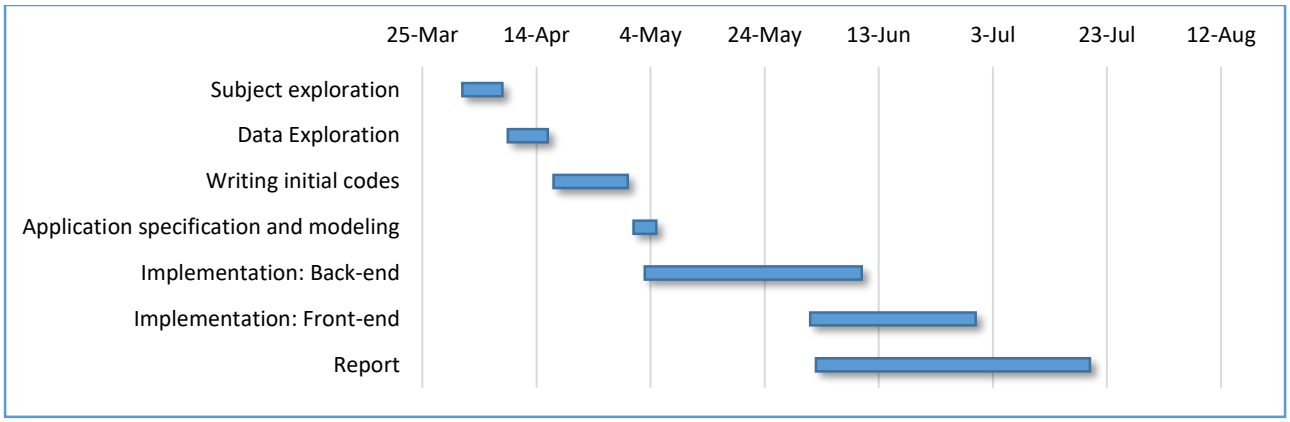


Figure 6 Gantt chart

3. Development process

- V model: The V-model is an SDLC model where execution of processes happens in a sequential manner in a V-shape. It is also known as the Verification and Validation model. [6]
- *Software Development Life Cycle (SDLC) is* a process followed for a software project. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process. [6]

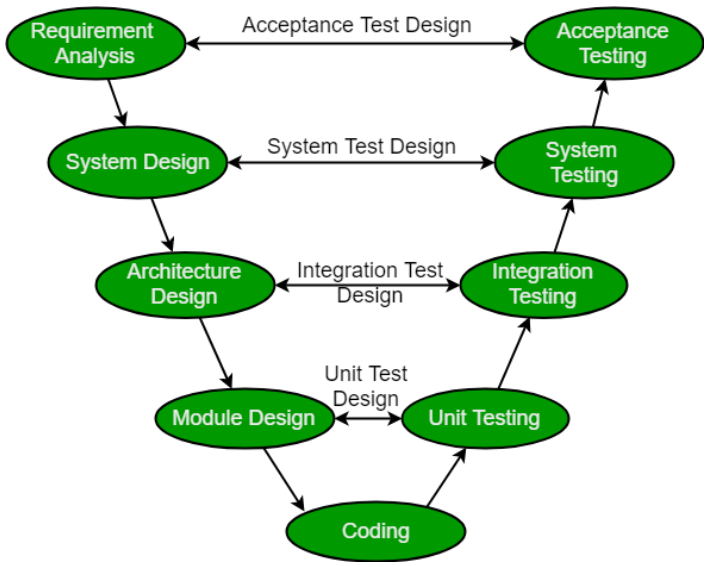


Figure 7 V model

- *Requirements Analysis:* This is the first phase in the development cycle where the product requirements are understood from the customer's perspective: Specification of user requirements and the analysis of the project.
- *System Design:* The system design will have the understanding and detailing the complete hardware and communication setup for the product under development: Specification and modeling of the application using UML diagrams.
- *Architecture Design:* Technical translation of functional specifications, and description of interfaces of the application.
- *Component/Module Design:* Consisting of precisely defining each subset of the application, and documentation that defines each functionality of the application.
- *Implementation/Coding:* The actual coding of the system modules designed in the design phase is taken up in the Coding phase
- *Component Test/Unit Testing:* Unit testing is the testing at code level and helps eliminate bugs at an early stage.
- *Integration Testing:* Integration tests are performed to test the coexistence and communication of the internal modules within the system.
- *System Testing:* System tests check the entire system functionality and the communication of the system under development with external systems.
- *Acceptance Testing:* Acceptance testing is associated with the business requirement analysis phase and involves testing the product in user environment.

4. Conclusion

This chapter was the starting point for the development of the application, the different stages of the project were presented. In the next chapter we will discuss the application specification and design.

Chapter 3: Specification and Design

1. Introduction

The purpose of specification and design is to give a clear picture of the application we are planning to create and what is required to do without getting in the details of the code which will be covered in the implementation chapter. In addition to top level details of the application's behavior.

To carry out these two phases, we use certain methods and conventions, such as UML diagrams.

2. Unified modeling language

UML, short for *Unified Modeling Language*, is a standardized modeling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. [7]

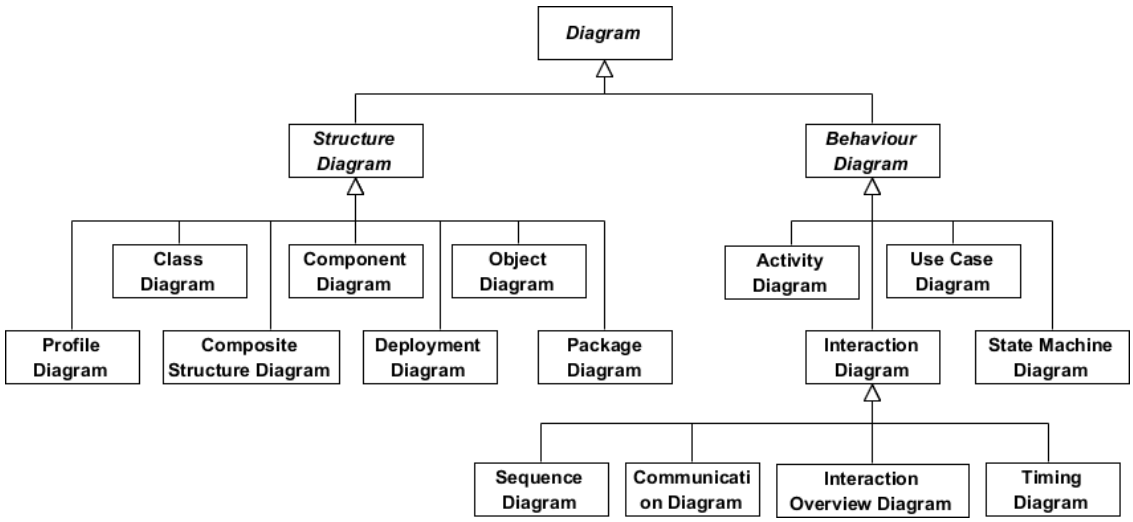


Figure 8 UML architecture

3. Specification

In order to achieve our objectives, we have to create multiple interfaces, each one presents a certain branch, but to make every feature in just one place, we will use one main interface that leads to other branches.

In a first place, let us get through the requirements of our application.

3.1. Match Analysis

Our main objective is to create analysis of events occurred in football matches, to do that we will cut this section into two parts: overall analysis and player analysis.

In the overall analysis we will have the following features:

- Plot the location of each shot of both teams on a field plot.
- Plot the location of all dribbles occurred in a match, with the names of successful dribblers.
- Plot the location of all fouls occurred.
- A list of bookings: Yellow/Red card with the name of the player and their team.
- A list with numbers of passes, missed passes and the frequency of missing passes of each player.
- Plot a possession chain analysis. It contains the events occurred before a major event like scoring a goal, missing the ball…etc.
- The lineup of each team.
- Passes ranking in the match.
- Number of fouls committed by both teams.
- Number of passes of both teams.
- Number of goal attempts by both teams.
- Shots heatmap of both teams.
- Shots analysis in depth: Expected goals analysis.
- Top 5 lists of players with most passes, most dribbles and the players who missed the ball the most.

In the player analysis we will have the following features:

- Overall detailed stats of a player, like number of passes, ball receipt ...etc.
- Plot the location of successful and unsuccessful dribbles.
- Plot the location of passes.

- Heatmap of movements of the player.
- Plot shots locations of the player.

3.2. Team Analysis

In this section we will need the overall performance analysis of a team within a certain season or in the whole dataset. One of the problems we have faced is that the event data, does not contain the season of the matches, instead the season of each match was in other repository: matches.

For example, when we want to analyze the shots of a team in the whole season, we have to take these steps:

- ✓ Loop through all the event files and keep only the events with the team name desired and type event as shots.
- ✓ Then we will have to merge the dataframe obtained with the matches files by the match id, in order to obtain the season.

With this algorithm, it will be possible to analyze the performance of each team in the whole season.

The team analysis section has multiple features, let us start with shots analysis:

- ✓ Plot a heatmap displaying the locations of the shots of the team.
- ✓ Plot a heatmap displaying the locations of scored shots.
- ✓ Plot a scatter displaying the correlation between the angle of the shot and the shot being scored or not.
- ✓ Expected goals plot: Proportion of shot resulting in a goal.
- ✓ Plot probability a shot scored by angle.
- ✓ Plot probability a shot scored by the distance of the shot from the net.

Other features of the team analysis section are:

- ✓ Displaying a list of top goal scorers of a team in a season.
- ✓ Results of the team in that season, and plot the number of wins/draws and losses.
- ✓ Display the manager of the team in that specific season.

3.3. Overall data

In order to navigate and understand the data easily, it is important to display the overall information about our data, this section will contain:

- ✓ Number of matches in the dataset of each team.
- ✓ ID of each match, the teams playing that match, and the competition.
- ✓ Competition data, containing competitions we have in our dataset, the seasons and the country of the competition.

Now that we have presented an overall description of our application, let us move to the next phase: Design.

4. Design

4.1. Use case diagrams

A use case diagram can summarize the details of the system's users (also known as actors) and their interactions with the system.

4.1.1. The overall use case diagram

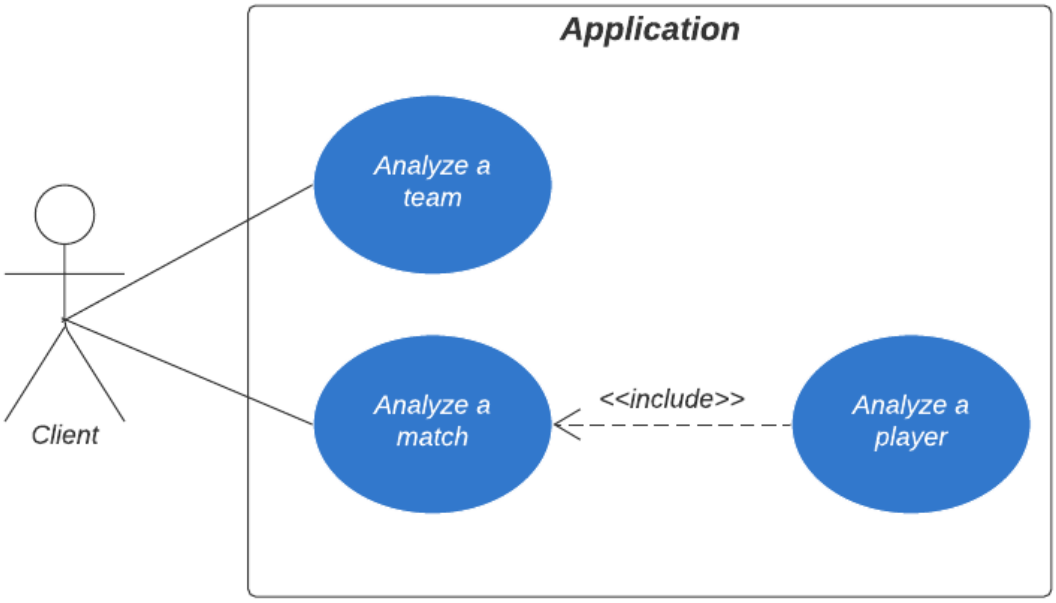


Figure 9 Overall analysis use case diagram

4.1.2. Analyze a team use case diagram:

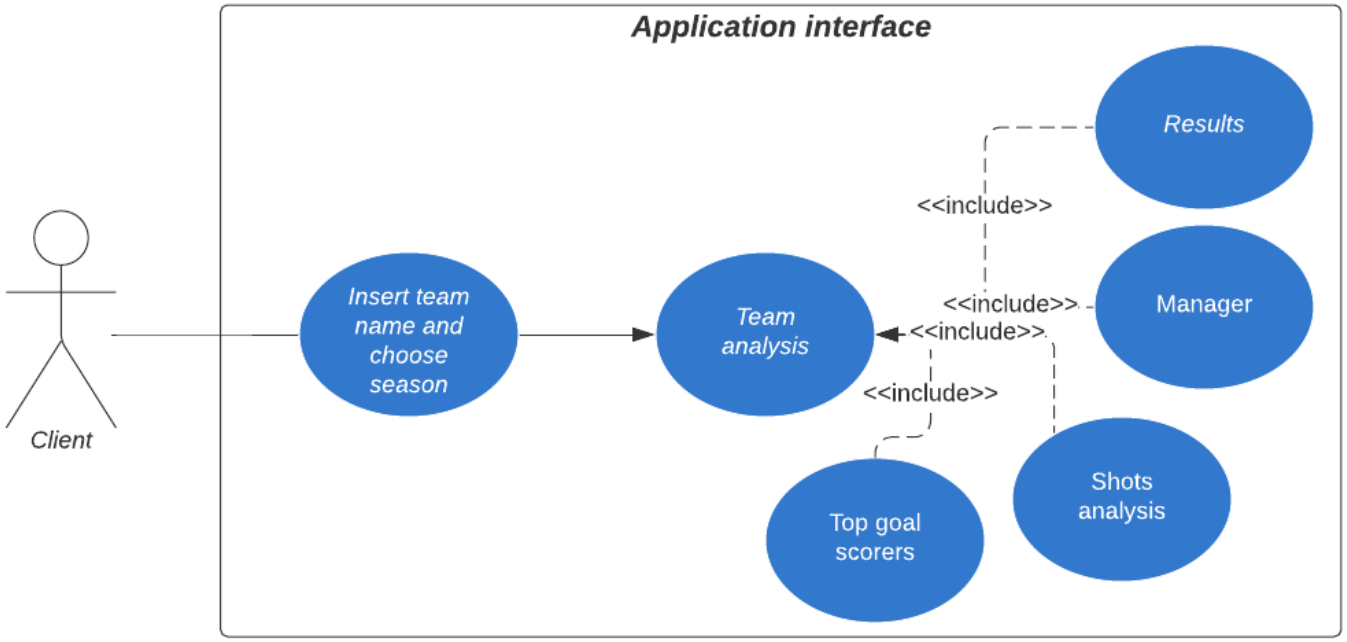


Figure 10 Team analysis use case diagram

4.1.3. Analyze a match use case diagram:

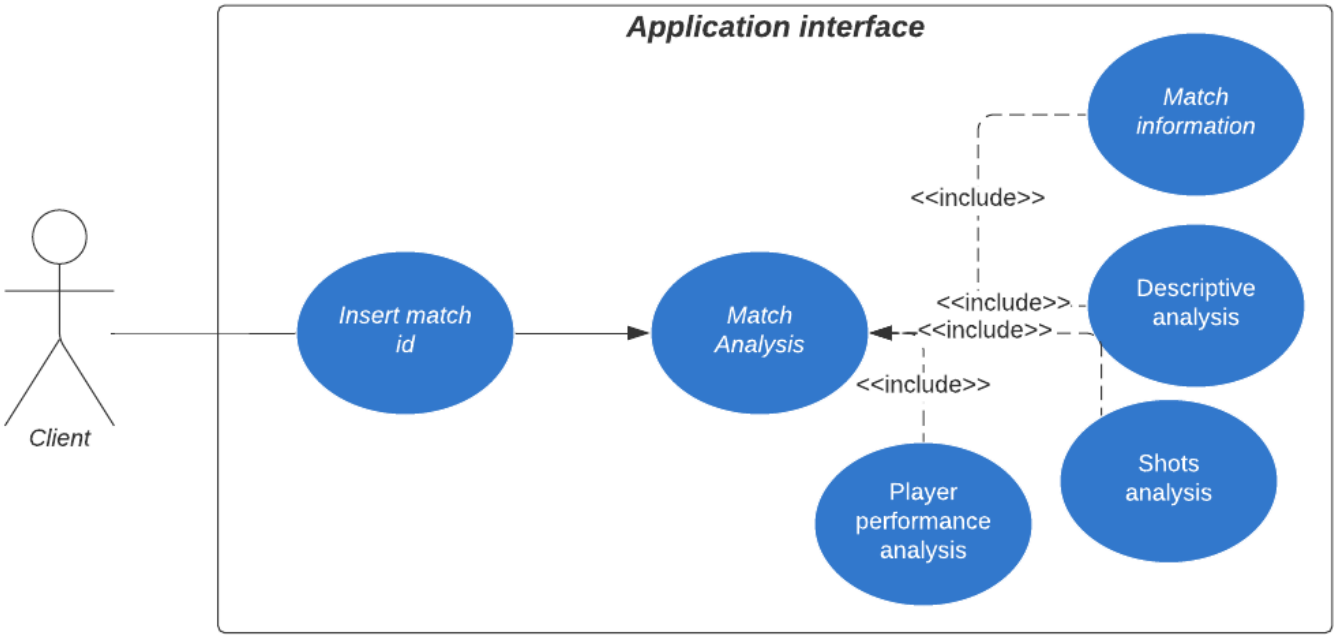
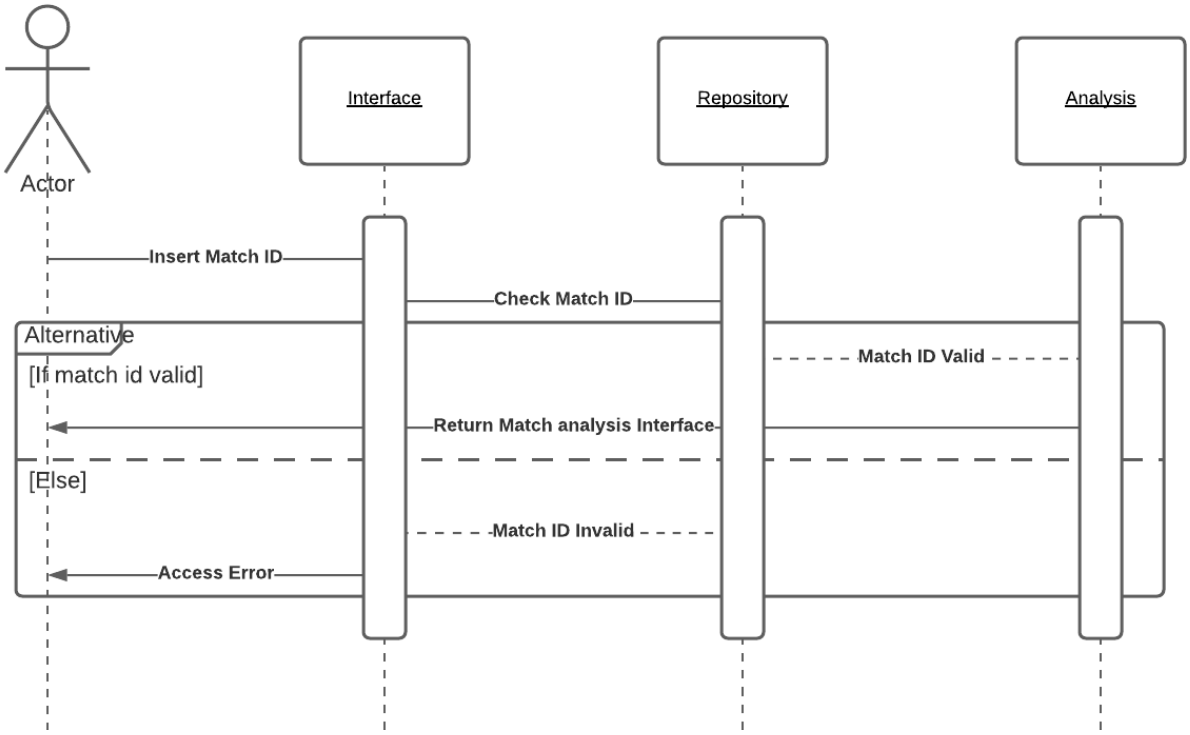


Figure 11 Match analysis use case diagram

4.2. Sequence diagrams

A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together.



5. Data providers

The data used in this project is made available for free by *StatsBomb* for research projects and genuine interest in football analytics.

Figure 12 StatsBomb logo

StatsBomb: "StatsBomb are football analytics experts. We devised and built a brand new, proprietary dataset because we knew we could do better than what was out there! Granular data with powerful analytics gives your organization a winning edge." [8] StatsBomb offers data using their API or their python package statsbombpy, but to access this data, the API needs to be registered(paid). However,

StatsBomb provide free sample data which can be obtained from their repository on GitHub: https://github.com/statsbomb/open-data

Why did I choose StatsBomb?

- Rich dataset, the data covers nearly every event in a football match, starting from basic data like passes and shots, to the specific details like timestamp and location of each event.
- More than 2x the data of any other data provider.
- Unlike many tracking datasets out there, *StatsBomb* gather real data with real players and teams, this is an advantage as it gives us an opportunity to study real football analysis problems.
- Free sample.
- *StatsBomb* work with teams in leagues across the globe: LaLiga, Premier league, Bundesliga, Ligue1, Ligue2 …etc.

StatsBomb also offer paid tracking and football analysis software.

6. Data

6.1. Data collection

There are several ways of collecting data, some of them are autonomous and others need the interference f humans, the most common methods of collecting data are:

- Data Scouts: Unlike traditional scouts that attend football matches of a certain player to sign him, data scouts record data of a match using a software, and they record the events of the match of all players and not just one player. One example of this kind of data collection is sportsdata, a brand of Sportsradar, a global leader in understanding and leveraging the power of sports data and digital content for its clients around the world.
- Global positioning system (GPS) based player movement tracking data are widely used by professional football clubs and academies to provide insight into the performance of a player during training or a match.

6.2. Data architecture

The data is provided as JSON files, in the following structure:

- Competition data stored in competitions.json. Each repository within is named for a competition ID, each file is named for a season ID within that competition.
- Events and lineups for each match, stored in events and lineups repositories respectively. Each file is named as the match ID.

In this project we will focus mainly on events that occur in matches.

6.2.1. Events repository

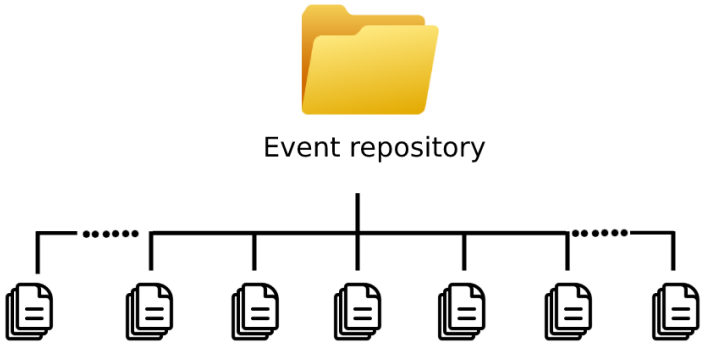


Figure 13 Event Repository

Each file in the event repository is named after the match id.

6.2.2. Matches repository

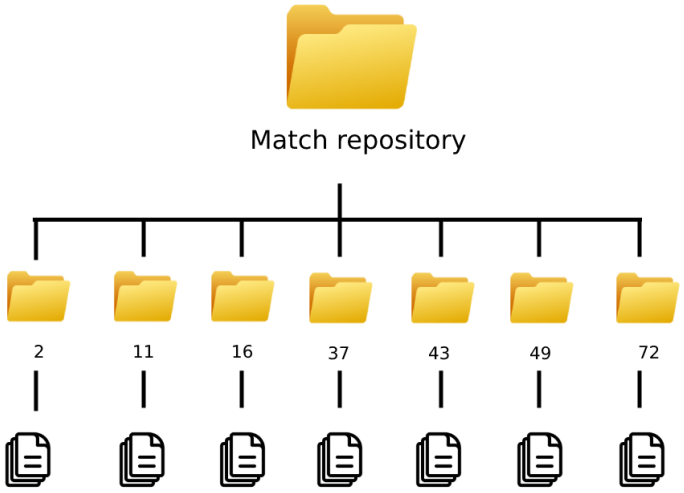


Figure 14 Match repository

Match repositories contains many other repositories, each one is named after the id of the competition.

Each file contains 104 columns:

```
1. with open(file_name, encoding='utf-8') as data_file:
2.     data = json.load(data_file)
3. df = pd.json_normalize(data, sep = "_").assign(match_id =file_name[:-5])
4. print(df.shape)
```

```
>> (3531, 104)
```

In this chapter we will present the most common columns, or the ones that we are using the most in our application, refer to Appendix 1 for the whole data architecture (104 columns). [15]

Column	Type	Child	Child Type	Description	Values	Value Description
id	uuid			The unique identifier for each event		
timestamp	timestamp			Time in the match the event takes places, recorded to the millisecond	e.g., 00:00:06.293	
type	object	id/name	integer/text	Id / name of the event type	42/Ball Receipt	The receipt or intended receipt of a pass
					2/Ball Recover	An attempt to recover a loose ball
					4/ Duel	

					8 / Offside	
					9 / Clearance	Clearing danger by a defender.
					10/Interception	
					14/ Dribble	
					16 / Shot	
					17 / Pressure	Applying pressure to an opposing player.
					22/Foul Committed	Any infringement that is penalized as foul play by a referee.
					23/ Goalkeeper	
					24/Bad Behaviour	When a player receives a card due to an infringement outside of play.
					30 / Pass	
					35/ Starting XI	
					38/ Miscontrol	Player loses ball due to bad touch
					43 / Carry	A player controls the ball.

team	object	id / name	integer	Id / Name of the team this event relates to.	e.g., 1 /Arsenal	
player	object	id / name	integer / text	Id / Name of the player this event relates to.	e.g.,5079/ Zlatan Ibrahimovic	
location	array [x,y]			Array containing x and y coordinates of the event.	e.g., the center of the field is (60,40)	See Appendix 3 below for more information.

Table 1 Data Architecture

This was an overall overview of the most used columns in our analysis, in the appendix we will discuss in detail the functioning of each column.

6.3. Data sample

6.3.1. Event data

We will take the example of a match of id 7430: Washington Spirit VS North Carolina Courage.

In a first glance, we will have a look at sample of the main columns of our data:

```
pd.set_option("display.max_columns", None)
df[["team_name", "timestamp", "location", "type_name",
"player_name"]].sample(5)
```

	team_name	timestamp	location	type_name	player_name
1303	Washington Spirit	00:32:54.213	[68.0, 71.0]	Pass	Andi Sullivan
1235	North Carolina Courage	00:31:00.850	[85.0, 68.0]	Pass	Denise O"Sullivan
2434	North Carolina Courage	00:18:14.243	[19.0, 33.0]	Pass	Abby Dahlkemper
2584	North Carolina Courage	00:23:01.363	[43.0, 73.0]	Pass	Merritt Mathias
211	North Carolina Courage	00:04:37.933	[116.0, 68.0]	Dribble	Lynn Williams

Figure 15 Event data Sample

Each event has its timestamp, location, type of the event and the player name, alongside other columns that depend on the event occurred, like when a shot happens we would like to know the outcome of the shot.

How many events we can have in a single match?

```
df.shape[0]
>>> (3531, 104)
```

In this match example, we have 3531 events described by 104 columns.

The field size in the dataset is measured in yards. The length of the field is 120 yards, the width of the field is 80 yards.

6.3.2. Match data

	match_id	match_date	home_team_home_team_name	away_team_away_team_name	competition_competition_name	season_season_name	referee_name
0	69163	2019-06-24	Sweden Women's	Canada Women's	Women's World Cup	2019	K. Jacewicz
1	68346	2019-06-20	Cameroon Women's	New Zealand Women's	Women's World Cup	2019	NaN
2	68355	2019-06-22	Germany Women's	Nigeria Women's	Women's World Cup	2019	NaN
3	69301	2019-07-06	England Women's	Sweden Women's	Women's World Cup	2019	A. Pustovoitova
4	68311	2019-06-18	Jamaica Women's	Australia Women's	Women's World Cup	2019	NaN
5	69205	2019-06-29	Italy Women's	Netherlands Women's	Women's World Cup	2019	C. Umpiérrez
6	69137	2019-06-23	France Women's	Brazil Women's	Women's World Cup	2019	NaN
7	69208	2019-06-29	Germany Women's	Sweden Women's	Women's World Cup	2019	S. Frappart

Figure 16 Match data sample

The match data files have 42 columns, each column describe the match, such as the referee, the stadium, teams playing, …etc.

7. Conclusion

This chapter was devoted to the specification and design of the project in which we drew sequence diagrams and use case diagrams. In the design phase we presented the overall architecture of the application and data.

The Specification and Design phase, with all these elements together, allowed us to delineate the project and get a global idea of its structure. This has greatly facilitated the implementation procedure of the system, which will be the subject of the next.

Chapter 4: Implementation of the Application

1. Introduction

In this chapter, we will describe the system in a finer level of detail down to the code level and describe the problems we have faced during implementation and how we dealt with them.

Besides the implementation, first we will take a look at the tools we used in this project, the tools include software, programming language and its libraries.

2. Development tools

2.1. Programming language

- Python: An interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation.

If we want to present the reasons that pushed us to choose python we would be writing a whole chapter just about that, briefly, we have used python because of its vast libraries support and improved productivity.

2.2. Essential libraries

- Pandas: A software library written for the Python programming language by *Wes McKinney* for data manipulation and analysis. Pandas provides rich data structures and functions designed to make working with structured data fast, easy, and expressive. It is one of the critical ingredients enabling Python to be a powerful and productive data analysis environment. [9]
- Numpy: Numerical Python, is the foundational package for scientific computing in Python., adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. [9]
- Matplotlib: The most popular Python library for producing plots and other 2D data visualizations. It was originally created by John D. Hunter and is now maintained by a large team of developers. It is well-suited for creating plots suitable for publication. [9]

- Tkinter: The most commonly used library for developing GUI (Graphical User Interface) in Python. It is a standard Python interface to the Tk GUI toolkit shipped with Python. As Tk and Tkinter are available on most of the Unix platforms as well as on the Windows system, developing GUI applications with Tkinter becomes the fastest and easiest. [10]
- Seaborn: Seaborn is a Python data visualization library built on top of matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. [11]
- Pandastable: Provides a table widget for Tkinter with plotting and data manipulation functionality. It uses the pandas DataFrame class to store table data. Pandas is an open source Python library providing high-performance data structures and data analysis tools. [12]
- Plotly: Open source python graphing library makes interactive, publication-quality graphs.

2.3. Software

- Sublime text: shareware cross-platform source code editor with a Python application programming interface. It natively supports many programming languages and markup languages, and functions can be added by users with plugins, typically community-built and maintained under free-software licenses. [13]

2.4. Hardware

This project was created with a computer with the following characteristics:

Name	HP Pavilion 17 Notebook
CPU	AMD A8-5550M
GPU	AMD Radeon HD 8550G
RAM	8 GB

Table 2 Hardware

3. Plotting

Generating visualizations is an essential part of our project, taking raw data and converting it to plots in order to make conclusions and help coaches rate the performance of the players.

In this project, we used a variety of plots, before discussing the implementation of the application we have first to define the plots used, and how they are going to benefits the potential users.

As defined in the previous section, we have used matplotlib and seaborn to create visualizations. With these two libraries we have created three types of plots:

3.1 The locations plot:

In this type of visualizations, we need to plot the location of the event that occurred on a field. In order to achieve this purpose, we have edited a function written by *jpj dejong* [14] that takes three arguments; the first argument is for the length of the field, second argument is for the width of the field, the third argument is used to choose the color of the lines to create the field. In our case, the default size of the field is 120yard in length and 80 yard in width.

The empty resulted plot is:

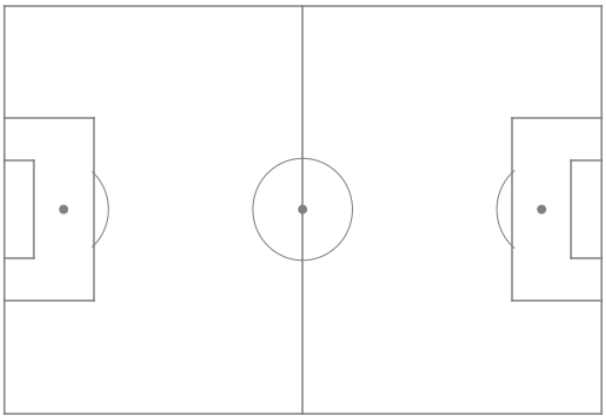


Figure 17 Field

In this plot we will display the locations of the events, such as the location of shooting the ball, the location of the dribble and the location of the pass. Refer to appendix 9 for the code to create this field.

3.2 Possession chain analysis plot

The second type of plot is quite similar to the first one, the main goal is to display events on a football field, the two main differences are:

- Multiple events: Visualize multiple events in one plot, creating a chain of events that enables us to understand the actions occurred in a ball position in terms of location and time frame.
- Dynamic plot: We can enable/disable events.

The code used to create the field is in appendix 10.

In order to create the chain, we are using the function in the appendix 12. The function takes four arguments:

- Figure: Specifying the figure where we would like to add the chain, in our case we will add the possession chain to the field above.
- Chain: Selecting the data we want to create a chain on, our main goal is to create a chain of possession of a certain team.
- Team_name: Here we select the home team.
- Opacity: by default it equals 1, it defines the opacity of the line between events that will be created.

The function iterates through the rows of the dataframe specified in the chain argument, and takes the location, timestamp and the name of every event that occurred.

Then the function assigns each event to the proper team and displays it on the figure.

3.3 Team analysis shots plot

Another special plot is used to plot the shots of a team in the whole dataset or in a specific season. Since we only need the shots of one team, we will be displaying then on one side of the filed. Let us have a look at an example of shots of Barcelona from the application:

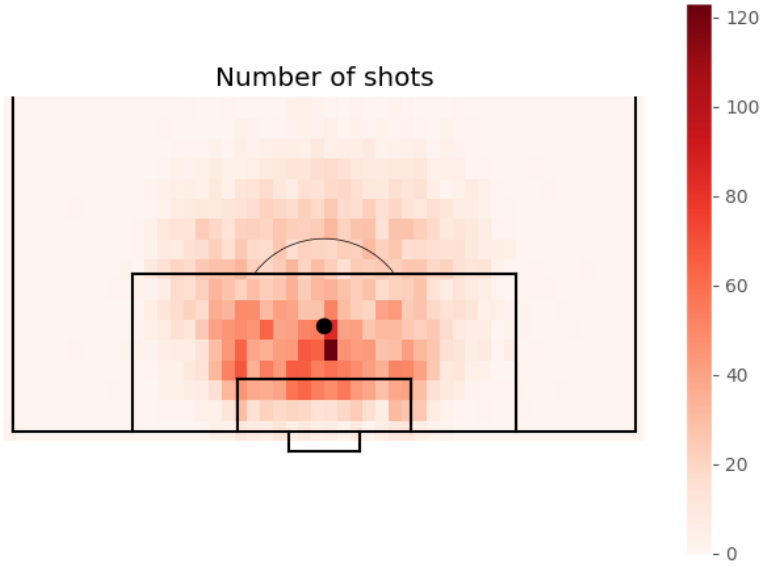


Figure 18 Shots locations

This plot takes the locations of the shots and displays them on this figure. The code of this type of visualization is in the appendix 11.

4. Expected goals

In order to calculate the expected goals metric, we are going to use a machine learning model, before deciding what algorithm we would use, we would like to try different algorithms and take the one that returned the best cv score.

The attributes that we would train the model on are the distance of the shot from the target and the angle of that shot, how we obtained these two attributes are explained in the next chapter.

4.1. Model choice

We have tested three models: Linear SVM, Logistic Regression and Gradient Boosting, we have gotten the following results:

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean
GradientBoostingClassifier	0.836470	0.844906	0.828034	0.857884	0.844906	0.842440
LogisticRegression	0.823491	0.844906	0.822842	0.858533	0.837119	0.837378
Linear SVM	0.822193	0.842310	0.822193	0.855938	0.833874	0.835302

Figure 19 Model choice

We can see that the best classifier is Gradient Boosting. The code used to test different algorithms is:

```
KFold_Score = pd.DataFrame()
classifiers = ['Linear SVM', 'LogisticRegression','GradientBoostingClassifier']
models = [svm.SVC(kernel='linear'),
          LogisticRegression(max_iter = 1000),
          GradientBoostingClassifier(random_state=0)
         ]
j = 0
for i in models:
    model = i
    cv = KFold(n_splits=5, random_state=0, shuffle=True)
    KFold_Score[classifiers[j]] = (cross_val_score(model, X, np.ravel(y),
scoring = 'accuracy', cv=cv))
    j = j+1

mean = pd.DataFrame(KFold_Score.mean(), index= classifiers)
KFold_Score = pd.concat([KFold_Score,mean.T])
KFold_Score.index=['Fold 1','Fold 2','Fold 3','Fold 4','Fold 5','Mean']
KFold_Score.T.sort_values(by=['Mean'], ascending = False)
```

4.2. Gradient Boosting

Gradient boosting is a machine learning technique for both regression and classification, it is one of the most common ensemble learning technique, it combines multiple weak machine learning models and delivers improved prediction accuracy. Before starting with creating the model first we should know what is boosting?

4.2.1. Boosting

Boosting algorithms play a crucial role in dealing with bias variance trade-off. Unlike bagging algorithms, which only control high variance in a model, boosting controls both the aspects (bias and variance).

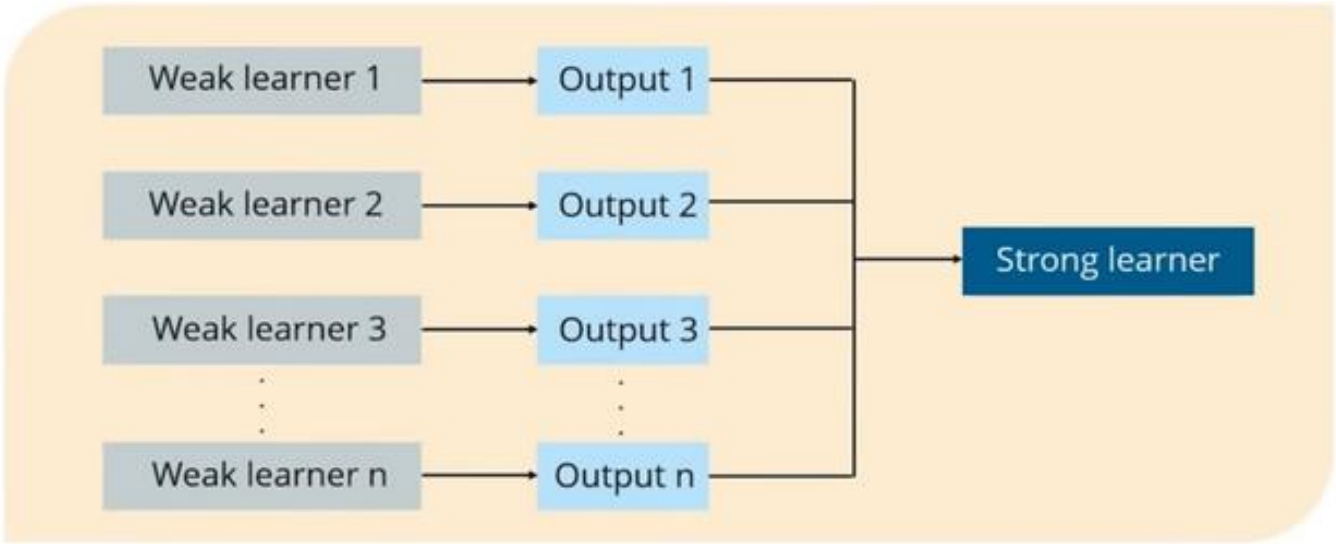


Figure 20 Boosting

Gradient in gradient boosting refers to the gradient of loss function, the function is optimized using gradient descent.

Combining multiple weak learners to obtain a strong learner usually returns good results but is it enough? Gradient boosting has many hyper parameters that we could tune; tree-specific parameters, boosting parameters and miscellaneous parameters. We are going to tune these parameters using GridSearchCV: library function that is a member of sklearn's model_selection package.

4.2.2. Hyperparameters

Parameters of gradient boosting can be divided into three categories:

- Tree-Specific parameters: Affect each individual tree in the model.
- Boosting parameters: Affect the boosting operation in the model
- Miscellaneous parameters: Other parameters of overall functioning.

In appendix 14 we presented in detail all the parameters of gradient boosting.

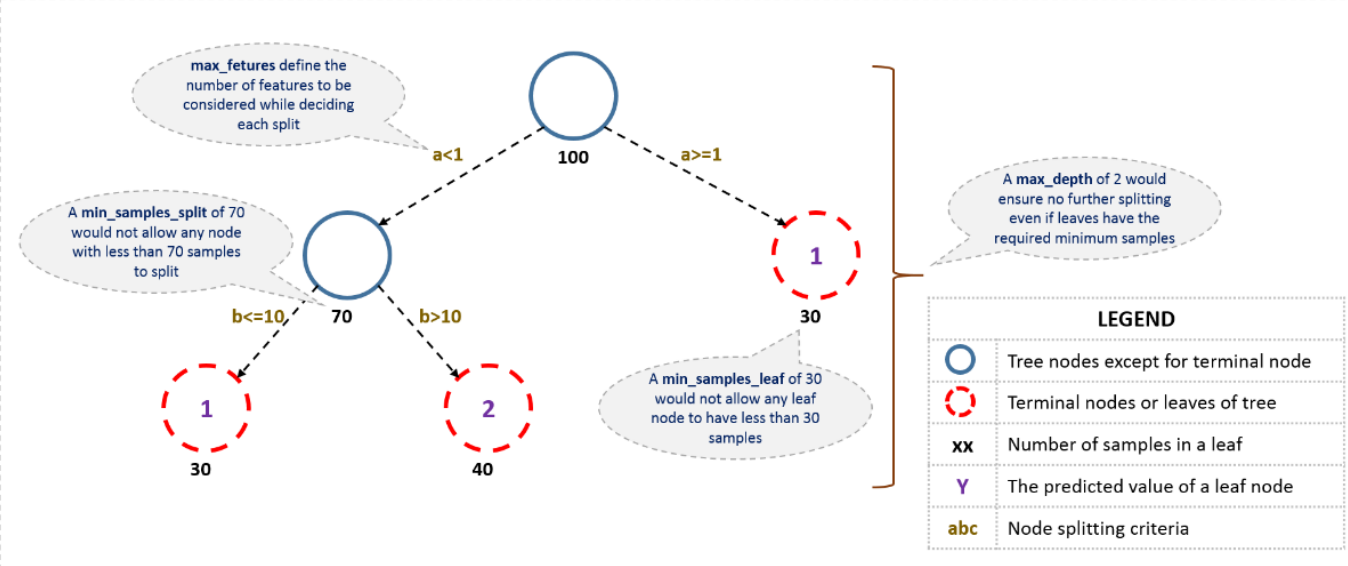


Figure 21 Gradient Boosting parameters

4.3. Tuning using GridSearchCV

Grid search is best described as exhaustive guess and check. It searches for the hyperparameters that result in the best cross validation score, and a set of values to try in the hyperparameter grid - the domain. The grid search method for finding the answer is to try all combinations of values in the domain and hope that the best combination is in the grid (in reality, we will never know if we found the best settings unless we have an infinite hyperparameter grid which would then require an infinite amount of time to run). Grid search suffers from one limiting problem: it is extremely computationally expensive because we have to perform cross validation with every single combination of hyperparameters in the grid!

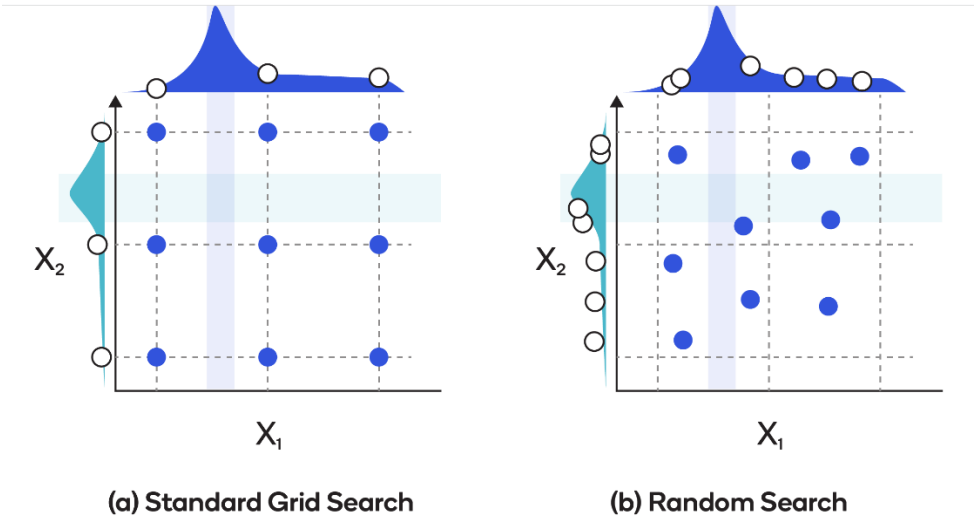


Figure 22 grid search vs random search

5. Application

5.1. Interfaces

Now that we have presented the types of plots that will be used in the application, we can move to presenting the interfaces of the application and how they automate the process of displaying visualizations of the game. Obviously we will not insert the whole code / large portions of code, but we will summarize the most important functionalities.Interfaces are created using Tkinter.

4.1.1 Main interface

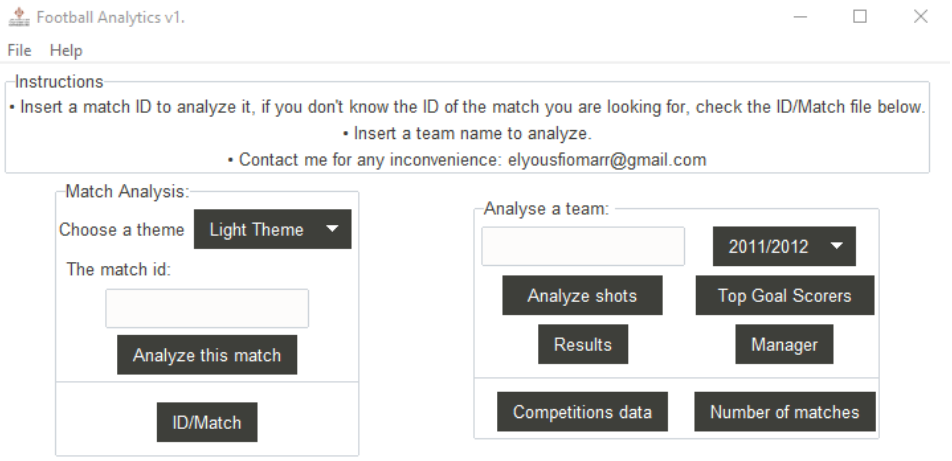


Figure 23 Light interface

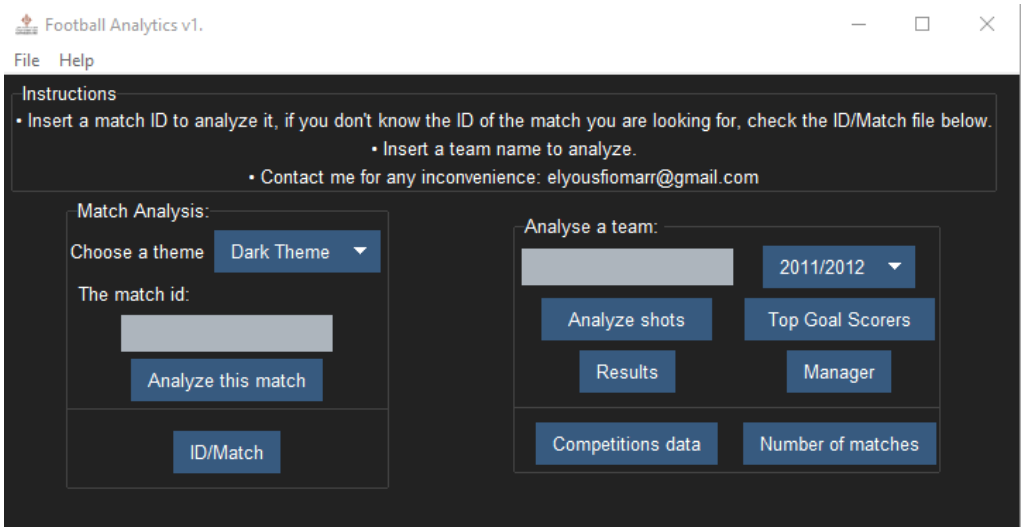


Figure 24 Dark interface

In order to create these themes, we used ttkbootstrap. ttkbootstrap is a collection of modern, flat themes inspired by Bootstrap for tkinter/ttk. First we had to create a customized theme using ttk creator:

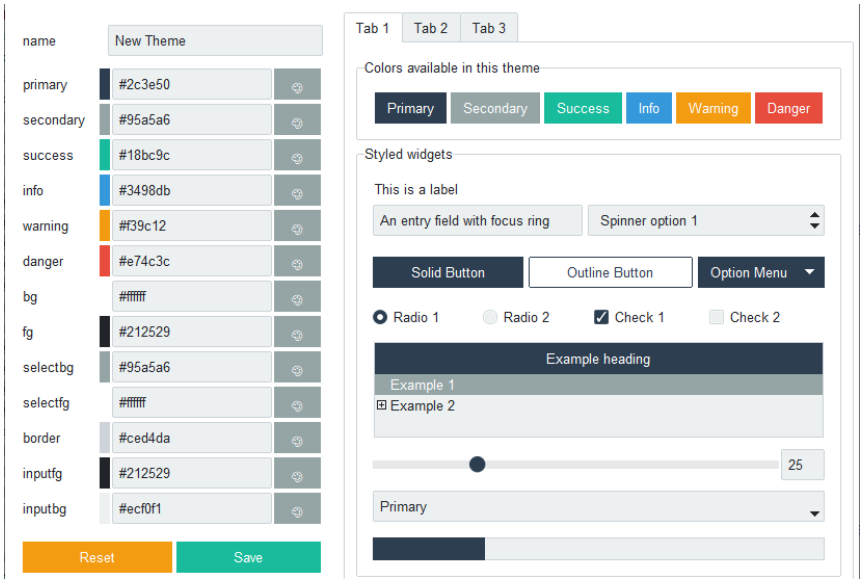


Figure 25 Tkinter creator

This interface of ttkbootstrap offers the possibility to edit and create any theme we want for our application. We created the two desired themes and left the choice to the client.

```
if tkvar.get() == "Light Theme":
    style = Style(theme="sandstone", themes_file =
"ttkbootstrap_themes.json")
elif tkvar.get() == "Dark Theme":
    style = Style(theme="darktheme", themes_file =
"ttkbootstrap_themes.json")
window = style.master
```

In the left, we have the match analysis frame, where we can insert the ID of the match we want to analyze, and in the right we have the team analysis frame. In the right side we find the team analysis frame, where we can insert a team's name and chose a season, then analyze the shots, display top goal scorers, results of the team in that season and the manager. Also, it has two more functionalities, the competitions data and number of matches of each team in the dataset.

4.1.2 Match analysis interface

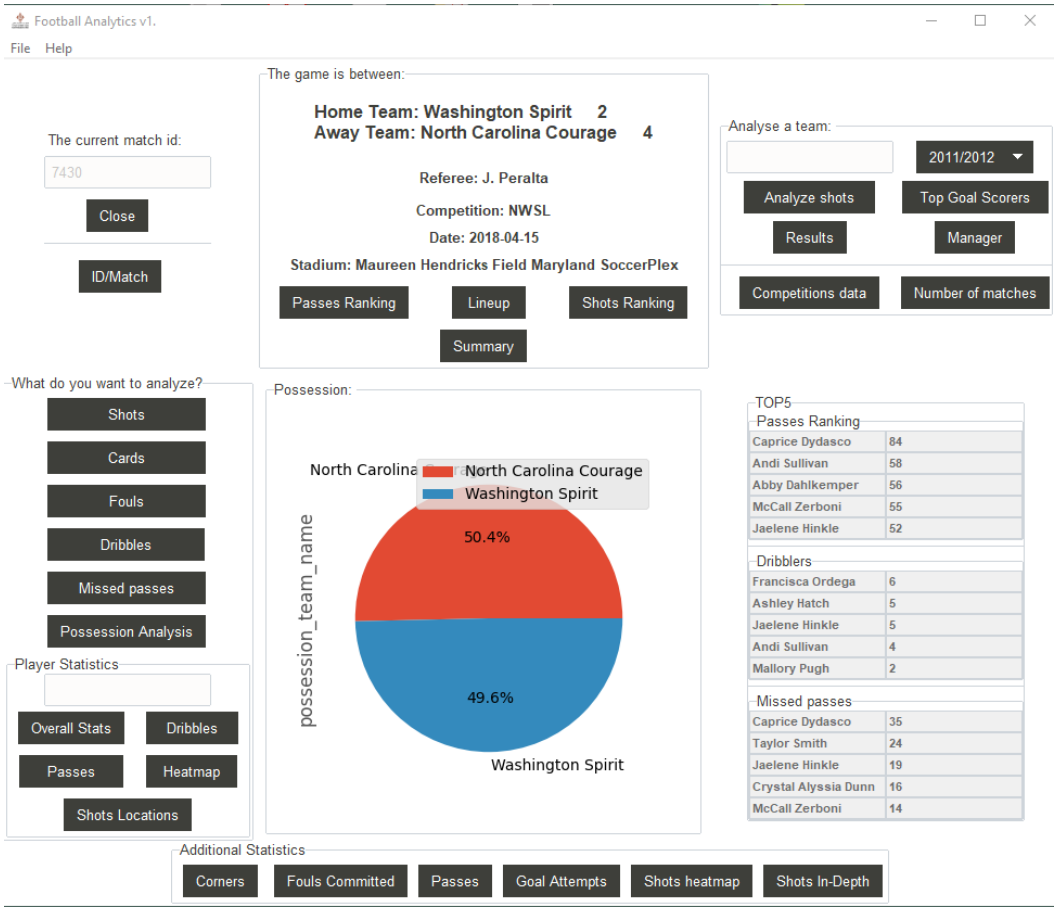


Figure 26 Match analysis interface

Once we submit the match id; 7430 as an example, this page pops out.

The top side contains three frames, the ones on sides are the same frames we have seen in the main interface, the frame in the middle is dedicated to displaying the overall information about the match, such as:

- The teams playing the match.
- The score of the match.
- Date of the match.
- Referee, stadium and competition of the match.

In addition to two buttons that display the lineup of both teams and the passes ranking sorted in descending order.

The middle of the interface contains multiple frames:

Left frame is for descriptive analysis of the match, it is divided to two portions, the first portion is a descriptive analysis of the performance of both teams in the match, the second portion is a descriptive analysis of the performance of individual players.

The middle frame is pie chart illustrating the possession rate.

The right frame displays top five players in: passing ranking, number of dribbles, missed passes.

The bottom of the interface has additional descriptive statistics of the match such as number of passes, number of goal attempts, shots heatmap, fouls committed and an analysis of the shots.

4.1.3 Team analysis interface:

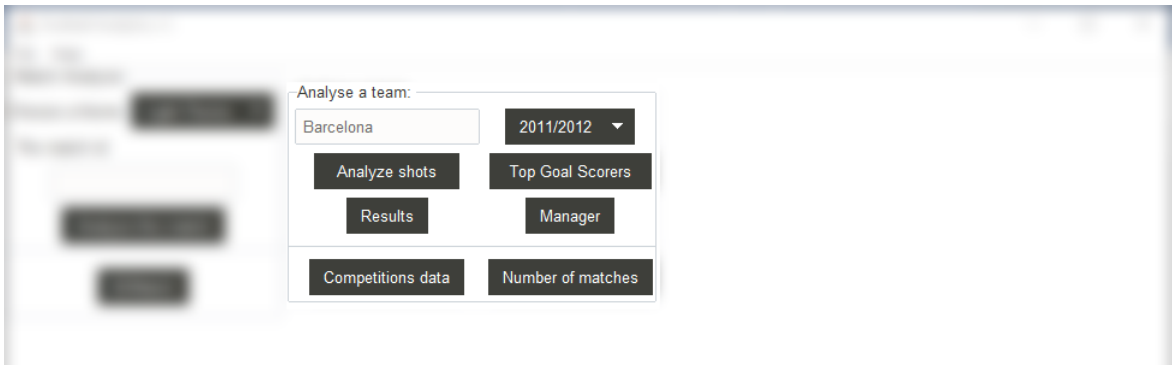


Figure 27 Team analysis interface

Team analysis allows us to understand the performance of a team not only in one match, but in a whole season. One of the important features we discussed earlier is the expected goal analysis, in our project we used multiple methods to analyze the shots of the team from different sides.

Let us take the example of Barcelona, once we write the name of the team and we choose the season we want to analyze, we can click on any of the buttons desired, we are going to start with Analyze shots button.

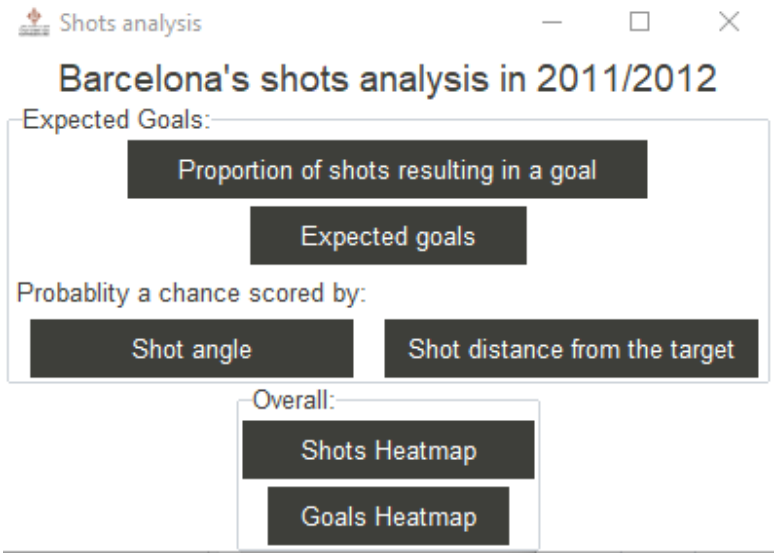


Figure 28 Barcelona shots analysis

6. Conclusion

In this chapter, we presented the different interfaces of the application, plots used in the application and expected goals model. In the next chapter we will go through the results obtained and their demonstration.

Chapter 5: Results and Demonstrations

1 Introduction

In this chapter, we will present the results of our work, demonstrate how we calculated the expected goals metric based on the machine learning model we implemented in the previous chapter and explain how each feature behaves.

2 Results

2.1 Match analysis

In order to understand how a team performed in a match, it is important to start by visualizing the most important events. The first thing a football team manager would like to know is did his team dominate the game? How did they perform with the ball? Did they have chances? And many other questions that a descriptive analysis could answer.

With start off with the analysis of shots, let us take an example of the match we selected earlier with the id 7430.

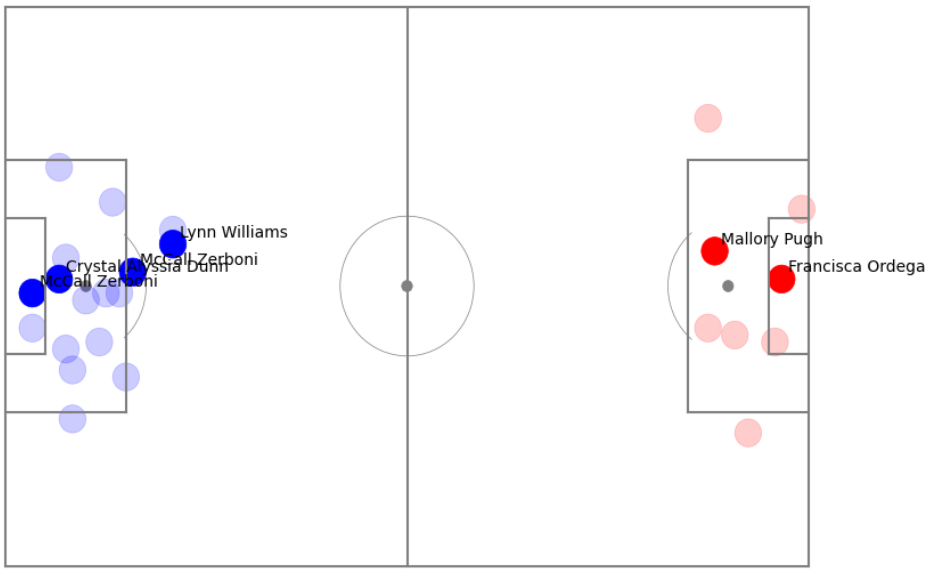


Figure 29 Shots analysis

In the first glance we can say that blue team (or the away team) had more chances and scored more goals, which does not correlate with the fact that the home team had more possession of the ball by 1%.

As we can see, just one good plot can give so much information about the performance of the teams, and it can describe what happened in a match.

To achieve that result, we first create a dataframe named ‘shots’ containing only the shots events, then we iterate through the rows of this dataframe, and assign the location of the events to x and y, x being the x-axis and y being the y-axis.

In order to highlight the shots that resulted in a goal, we created a noolean variable named ‘goal’, it equals true when the shot outcome is ‘Goal’.

Then we check the team of the event, if it is the home team, the circle would be red, if it is the away team, the circle would be blue. In order to display the home team shots on one side and the shots of the away team on the other side of the field, we placed the home team shots locations in (x, pitchWidthY – y), in our case the pitchWidthY is 80, and we placed the away team shots in the (pitchLengthX –x, y), in our case pitchLengthX is 120.

Inside this condition, we check if each shot results in a goal, if so, the circle would be darkened and the name of the player that shot the ball besides it. The example above is a good reference, where the away team scored four goals, and the home team scored two goals. The code is inspired by the work of ‘Friends of Tracking Data’. [14]

```
def shots_locations():
    shots = df.loc[df['type_name'] == 'Shot'].set_index('id')
    (fig,ax) = createPitch(120,80,'yards','gray')
    for i,shot in shots.iterrows():
        x=shot['location'][0]
        y=shot['location'][1]

        goal=shot['shot_outcome_name']=='Goal'
        team_name = shot['team_name']

        circleSize=2

        if (team_name ==home_team ):
            if goal:
                shotCircle=plt.Circle((x,pitchWidthY-y),circleSize,color="red")
                plt.text((x+1),pitchWidthY-y+1,shot['player_name'])
            else:
                shotCircle=plt.Circle((x,pitchWidthY-y),circleSize,color="red")
                shotCircle.set_alpha(.2)
        elif (team_name==away_team):
            if goal:
                shotCircle=plt.Circle((pitchLengthX-
x,y),circleSize,color="blue")
                plt.text((pitchLengthX-x+1),y+1,shot['player_name'])
            else:
                shotCircle=plt.Circle((pitchLengthX-
x,y),circleSize,color="blue")
                shotCircle.set_alpha(.2)
        ax.add_patch(shotCircle)
    fig.set_size_inches(10, 7)
    plt.show()
```

To create the plots for fouls, dribbles and passes we follow the same approach of shots.

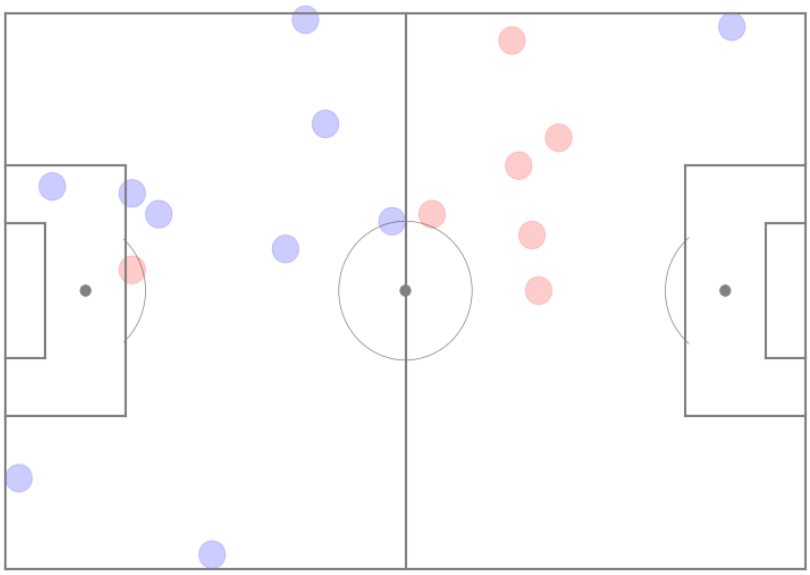


Figure 30 Fouls locations

Dribbles result:

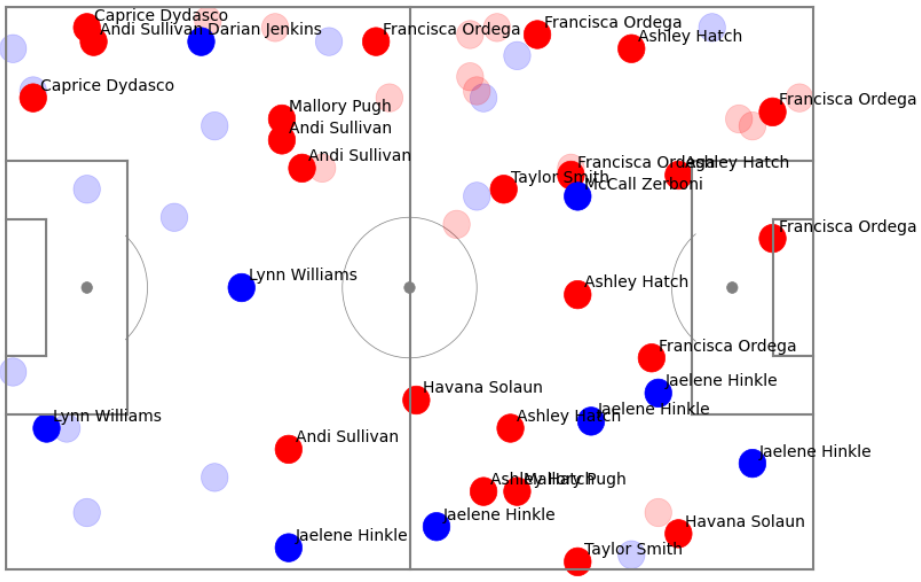


Figure 31 Dribbles locations

Here can see that the dark circles are for the players that made successful dribbles, meanwhile the light circle are for unsuccessful dribbles.

Until now, we have seen three plots, in each one we plot the location of an event and specify if it was successful or not. Now we are going to see a different type of analysis, the ranking of missed passes. This analysis is so important because it allows the manager to understand who are the players that waste the ball the most, therefore waste the ball possession of team.

Ranking the players based on missed passes does not always give the right analysis, since we should differentiate between a player that has made 50 passes in a game and missed 10, and another player that has made10 passes and missed 9 passes, that is why we are going to rank the missing passes by frequency:

$$Frequency = \frac{Missed\ passes}{total\ number\ of\ passes} * 100$$

We can see that in the following figure, Jessica McDonald has missed 12 passes out of 21 by a percentage of 57.1%, even though she missed less passes that other players like Taylor Smith and Crystal Alyssia, she still ranks first.

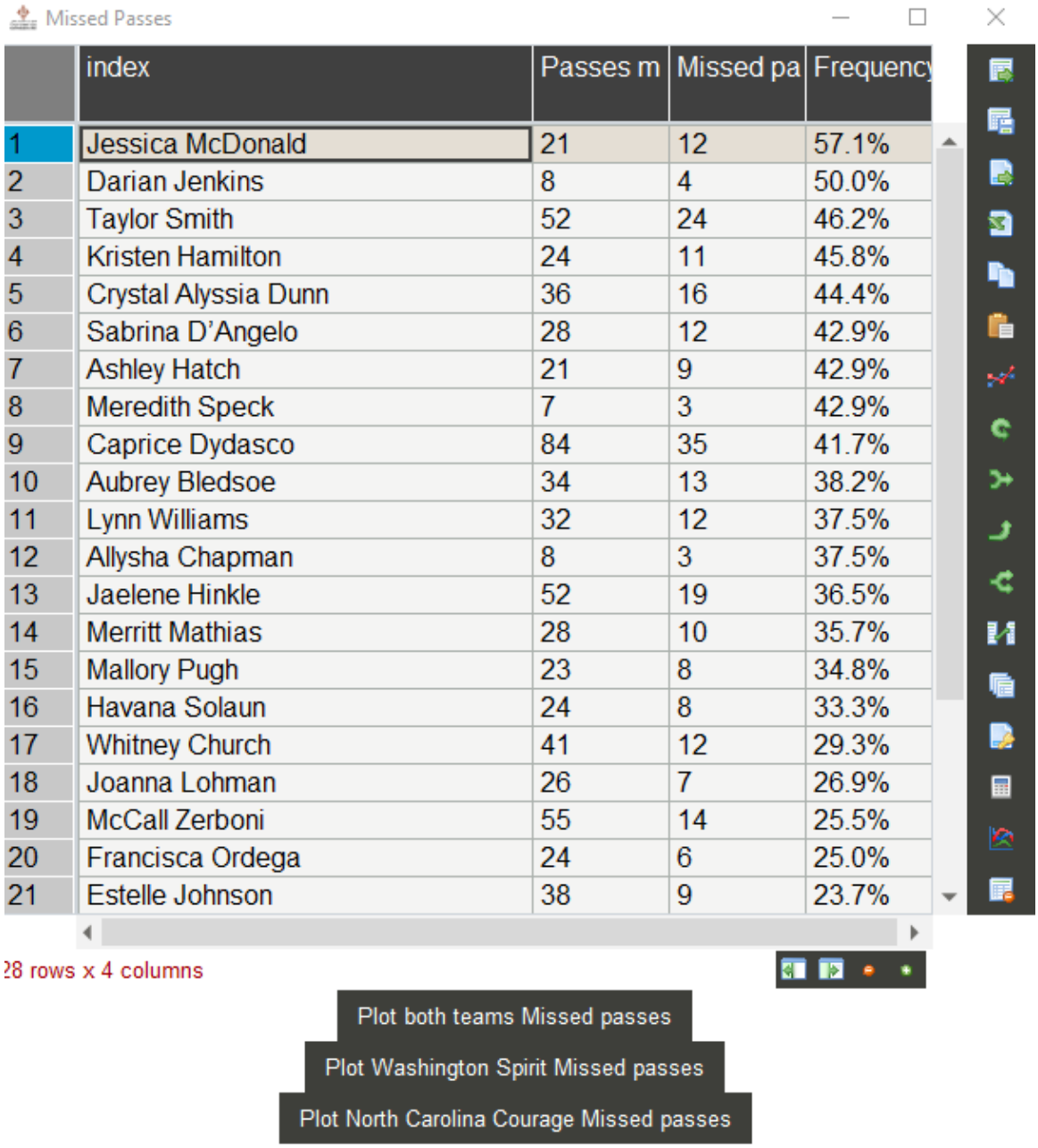


Figure 32 Missed passes ranking

The three plots below the table, allow the manager observe if there are certain areas of the field where players miss the ball the most. For example:

We can see here that the home team, Washington spirit, missed passes the most in their first half of the field, which explains why they had so many shots against them, since missing a pass in the first half of the field usually means a dangerous attack.

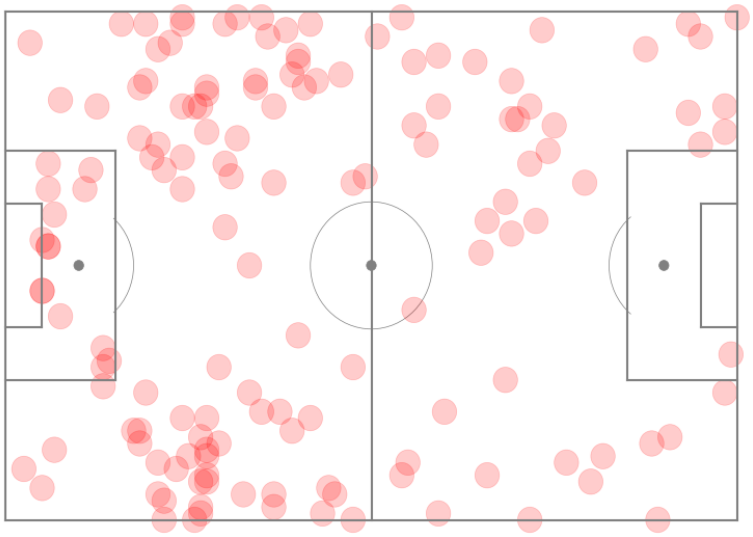


Figure 33 Fouls

The code behind the missed passes table is:

```
missed = df[df['pass_outcome_name'].isin(["Incomplete","Out","Pass Offside",
"Unknown"])]

# number of missed passes by players
m = pd.DataFrame(missed["player_name"].value_counts())
m.columns = [""]

# number of passes by players
passes = df.loc[df['type_name'] == 'Pass'].set_index('id')
p = pd.DataFrame(passes["player_name"].value_counts())
p.columns = [""]

# create a new column named frequency and sort the values in the descending
order
res = pd.concat([p,m], axis=1, keys=["Passes made", "Missed passes"])
res["Frequency"] = (((res["Missed passes"]/res["Passes
made"])*100).round(1).astype(str)+"%")
res = res.sort_values("Frequency", ascending=False)
res = pd.DataFrame(res).reset_index()

f = Frame(missedpass)
f.grid(row=0, column=0)
table = pt = Table(f, dataframe=res ,showtoolbar=True, showstatusbar=True)
pt.show()
```

The last feature is the possession chain. In the plotting section we have explained the code in detail, now let us see an example of the result. Instead of displaying the plot in the interface, the code of this create an html file that makes it easy to explore the plot dynamically, here is an example from our

application obtained from a match between Washington Spirit and North Carolina Courage. The code is inspired by the work of 'Friend of Tracking Data'. [14]

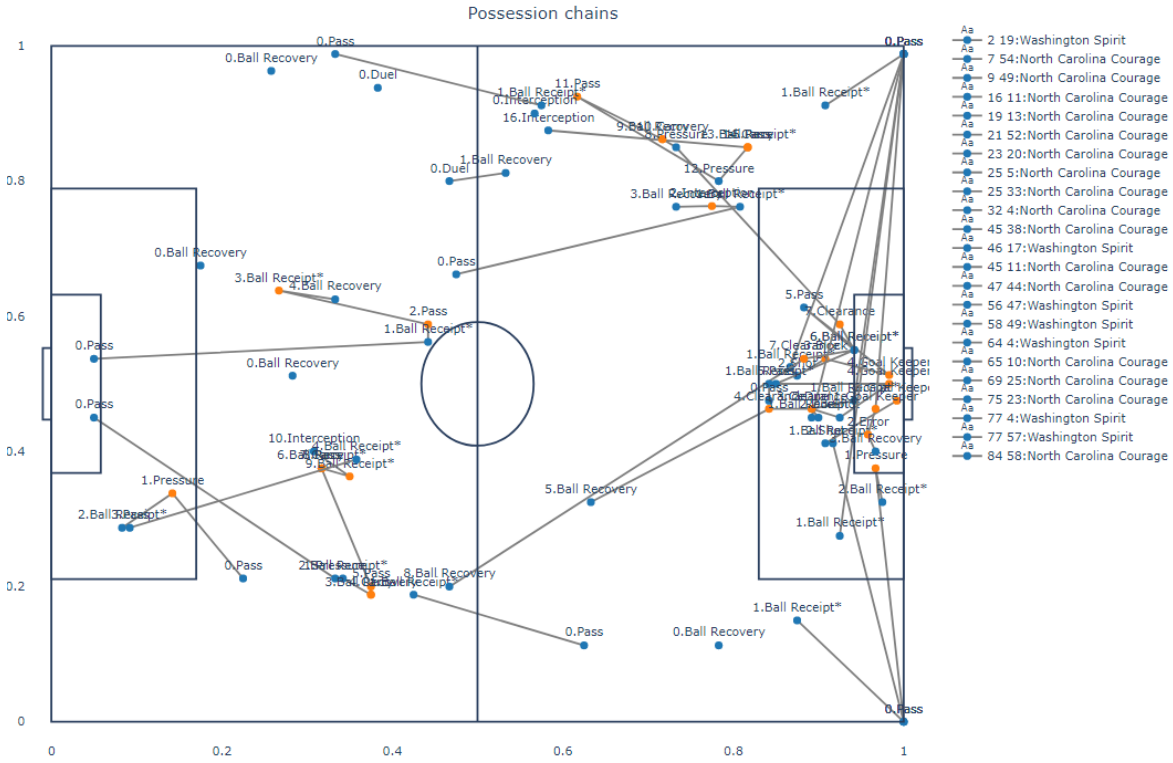


Figure 34 Possession chain

We can click on the buttons on the right side to enable or disable an event based on the timestamp of the event and the team name.

2.2 Team analysis

We cannot stress enough the importance of analyzing the performance of a team in a whole season, the application offers as a start, in depth shot analysis.

Before going to discuss the results and application, first we have to present the algorithm we used to merge hundreds of files into one file.

The goal is to obtain the locations of the shots of a team from all the event files, and then assign to each match id its season, since we do not have the season column in the events data, we have to loop through the events repository and match repository in order to merge these dataframes.

```
shots_df = pd.DataFrame(columns = ["Team", "Location", "Shot_Outcome",
"Season"])
path = os.listdir('C:\\Users\\OMAR\\Desktop\\PFE\\Interface\\events\\')
competitions_id = [2,11,16,37,43,49,72]
for file_name in path:
    with open('C:\\Users\\OMAR\\Desktop\\PFE\\Interface\\events\\' + file_name)
as data_file:
        data = json.load(data_file)
    df_t = pd.json_normalize(data, sep = "_").assign(match_id = file_name[:-5])
    mid = df_t["match_id"][0]
    shots = df_t.loc[df_t["type_name"] == "Shot"].set_index("id")
    shots = shots.loc[shots["team_name"] == "Barcelona"]
    for competi in competitions_id:
        files =
os.listdir('C:\\Users\\OMAR\\Desktop\\PFE\\Interface\\matches\\' + str(competi)
)
        for file in files:
            with open('C:\\Users\\OMAR\\Desktop\\PFE\\Interface\\matches\\'+
str(competi) + "/" + file) as f:
                temp = json.load(f)
            for i in range(len(temp)):
                if temp[i]["match_id"] == pd.to_numeric(mid):
                    shots = shots.assign(season =
temp[i]["season"]["season_name"])
    for i, shot in shots.iterrows():
        shots_df.at[i,"Team"] = shot["team_name"]
        shots_df.at[i,"Location"] = shot["location"]
        shots_df.at[i,"Shot_Outcome"] = shot["shot_outcome_name"]
        shots_df.at[i,"Season"] = shot["season"]
shots_df.to_csv("Barcelona.csv")
```

The first loop, goes through all the event files, at each event file, we store the match id in a variable named mid, and then we locate only the shots events of Barcelona, and now moving to assigning the season to the match, we iterate through the match repository, that has multiple folders, each folder has multiple files, so we have to create two other for loops.

After finishing this steps, now that we have obtained the data we want, we will keep only the columns we need: team name, location of the shot, the outcome of the shot and the season of the match. Finally, we save the obtained dataframe in a csv file.

Note: Refer to data architecture in specification and modeling chapter to better understand the architecture of repositories. The resulted dataframe is the following:

Team	Location	Shot_Outcome	Season
Barcelona	[111.7, 51.7]	Off T	2018/2019
Barcelona	[114.0, 27.0]	Off T	2018/2019
Barcelona	[92.0, 34.5]	Saved	2018/2019
Barcelona	[107.0, 25.0]	Off T	2018/2019
Barcelona	[108.1, 27.4]	Off T	2018/2019

Figure 35 Shots dataframe

- **Shots and goals:**

Now moving to create the heatmaps we need, we use the following code to adjust the dataframe:

```
for i,shot in shots_df.iterrows():
    shots_df.at[i,"X"] = 120 - shot["Location"][0]
    shots_df.at[i,"Y"] = shot["Location"][1]

# Create a Goal column, 1 == GOAL, 0 == NOT A GOAL
for i,shot in shots_df.iterrows():
    if shot["Shot_Outcome"] == "Goal":
        shots_df.at[i,"Goal"] = 1
    else:
        shots_df.at[i,"Goal"] = 0
```

We iterate through each row of the dataframe, and substract the position of the shot in the x-axis from the full length of the field in order to plot the results in one half of the field, then we create a new column named Goal, equals 1 if the shot outcome is a goal, it equals 1 otherwise.

To obtain meaningful results, we have to choose a team with many games in the dataset since we only have a sample data, checking the table in appendix 1 we can see that Barcelona has the most games, that is why we are going to analyze its performance in 2010/2011.

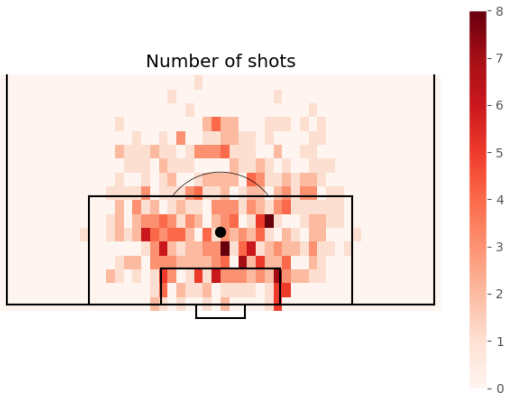


Figure 36 Number of shots

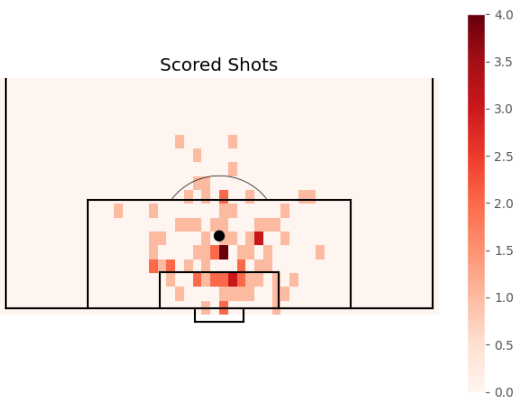


Figure 37 Scored shots

Comparing these two plots, we can see that in 2010/2011 season, most of Barcelona goals came from inside the penalty area.

To obtain these results, we have used the possession chain plot we mentioned in the plotting section, and to plot the results, we have used the following code:

```
H_Shot=np.histogram2d(shots_df['X'], shots_df['Y'],bins=50,range=[[0, 120],[0,
80]])
goals_only=shots_df[shots_df['Goal']==1]
H_Goal=np.histogram2d(goals_only['X'], goals_only['Y'],bins=50,range=[[0,
120],[0, 80]])
```

We created a two dimensional numpy histogram with the x-axis and y-axis position of the shots, and we do the same thing for the shots resulted in a goal.

We have used the histogram to highlight the number of occurrence of a shot in a certain position, and to display it on the field as a heatmap.

- **Proportion of a shot resulting in a goal:**

To obtain a proportion of shots resulting in a goal, we have divided the goal histogram by shots histogram, which gave the following result:

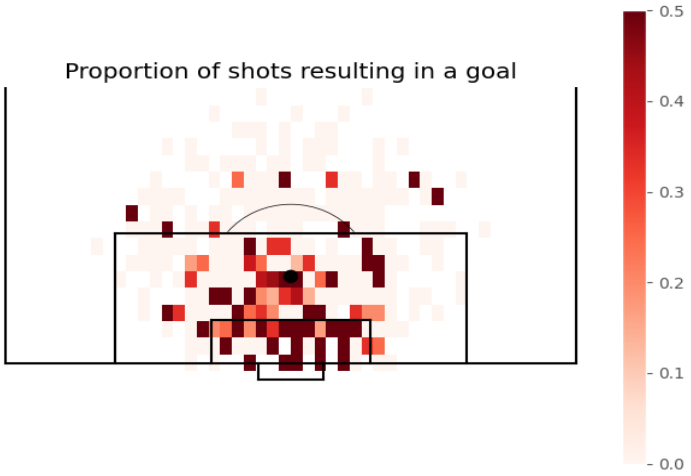


Figure 38 Proportion of shots resulting in a goal

- **Expected goals**

In order to calculate this probability, we have created a machine learning model, that predicts this probability based on the distance of the shot from the target and the angle of the shot. The dataset does not have these two metrics, so we have to calculate them.

- Distance of the shot from the target:

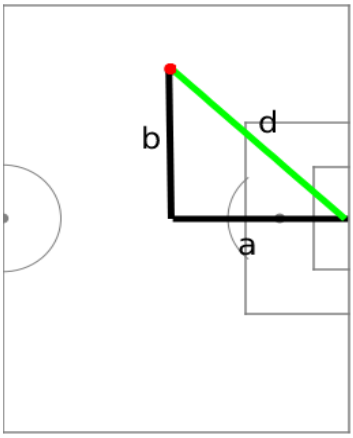


Figure 39 Distance between a shot and target

Let us consider this figure, the shot location is the red dot, so our goal is to calculate the green line: distance between the shot and target. First, we take half of the field where the shot occurred, so we

subtract the length of the field from the x-coordinate of the shot. We know that: $a = pitchLengthX - x$ and $b = |\frac{pitchWidthY}{2} - y|$

Based on 'Pythagoras theorem' we obtain the distance between the shot and the target:

$$d = \sqrt{a^2 + b^2}$$

The code applying these steps:

```
for i,shot in shots_df.iterrows():
    shots_df.at[i,'X'] = 120-shot["X"]
    x = shots_df.at[i,'X']
    y = shots_df.at[i,'Y'] - 40
    shots_df.at[i,'Distance'] = np.sqrt(x**2 + y**2)
```

We subtract the location of the shot on x-axis because by convention all the shots are on one side of the field (right side), and we subtract 40 from the location of the shot on the y-axis to determine if the shot was above, under or on the y-line center of the field.

Now we can calculate the percentage of shots being scored, we get the following graph:

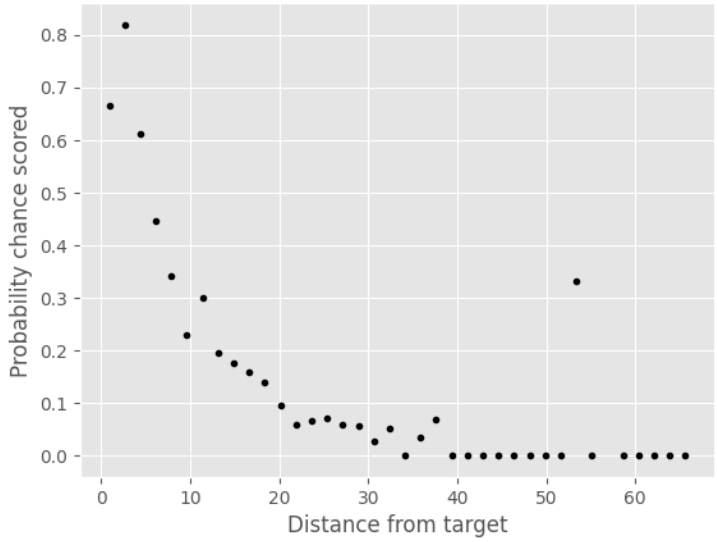


Figure 40 Probability chance scored

We can see that the bigger the distance between the shot and the target, the less that shot would end up as a goal. But distance is not the only factor that affects a shot being a goal or not, so let us calculate the angle of the shot and see its effect on the shot.

- Angle of the shot:

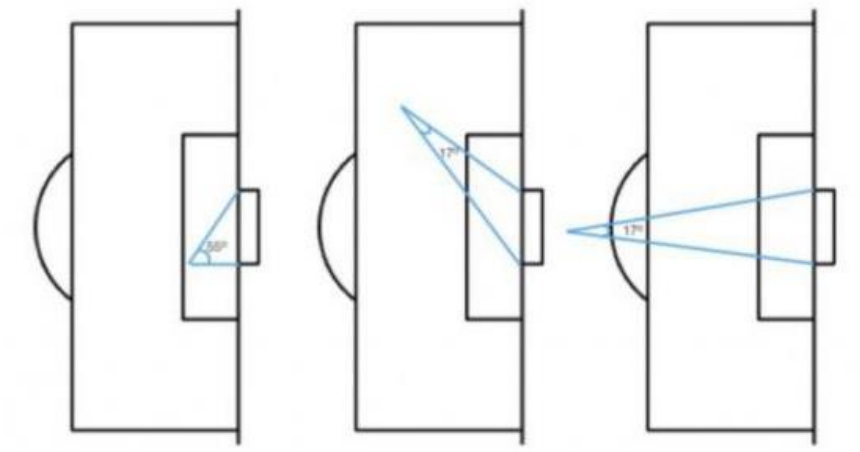


Figure 41 Angle of the shot

To calculate the angle of the shot we use cosines law, in this case, we know that:

$a = pitchLengthX - x$ and $b = |\frac{pitchWidthY}{2} - y|$. Based on Pythagorean theorem we obtain:

$$\tan(\alpha) = \frac{opposite}{adjacent}$$

So the angle of the shot is: (Refer to appendix 13 for the demonstration)

$$\alpha = \tan^{-1}\frac{t * x}{x^2 + y^2 - (\frac{t}{2})^2}$$

Note: t is the width of the target, in our data, the width of the target is 8 yards (Refer to appendix 8 for target coordinates).

The code to calculate the angle is:

```
for i,shot in shots_df.iterrows():
    a = np.arctan(8 * x /(x**2 + y**2 - (8/2)**2))
    if a<0:
        a=np.pi+a
    shots_df.at[i,'Angle'] =a
```

Now that we have calculated required attributes to create the model, and after tuning the hyperparameters of gradient boosting classifier using GridSearchCV, we have got the following results:

```
GradientBoostingClassifier(learning_rate=0.05, max_depth=6, max_features=2,
                           min_samples_leaf=2, min_samples_split=800,
                           n_estimators=1000, random_state=10, subsample=0.8)
```

With an accuracy of 0.88 and cv score of 0.74. Probability of a shot being a goal:

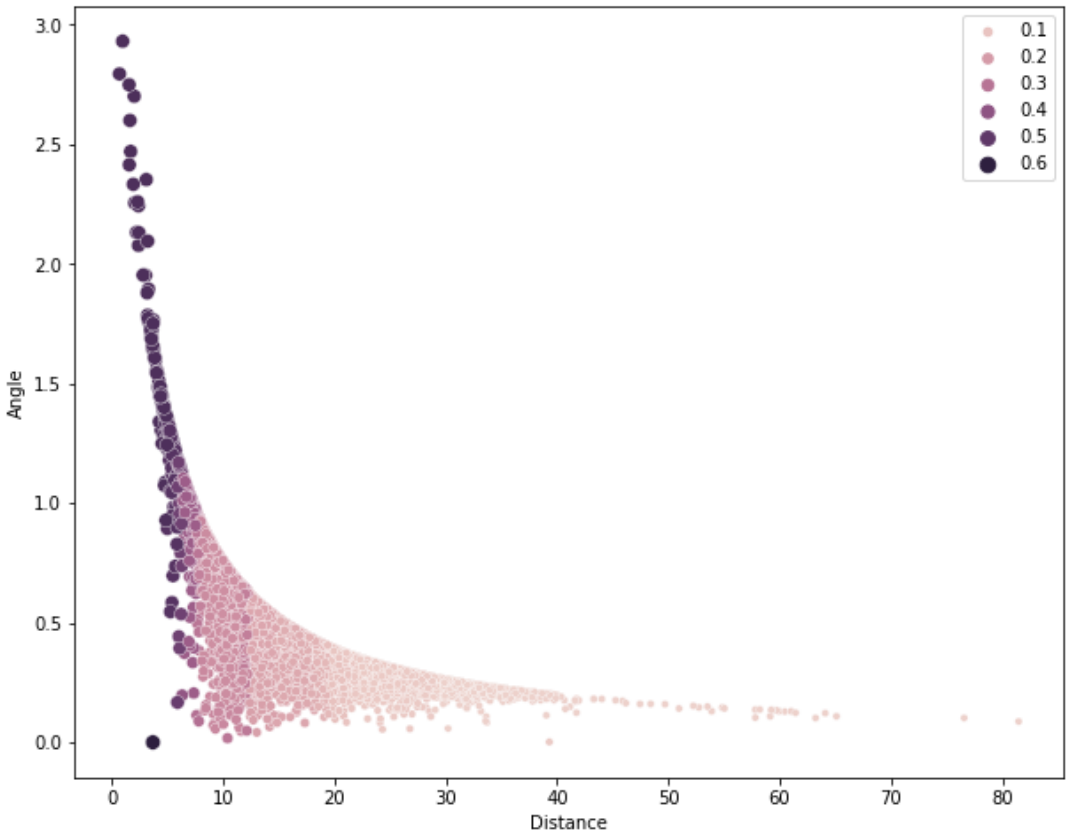


Figure 42 Expected goals predictions

We can see that the wider the angle of the shot and the closer the shot to the target, the higher probability of the shot being a goal.

- **Top Goal Scorers:**

We will follow the same algorithm we did when we merge the match repository with events repository, the difference is now we have to return the goal scorers instead of locations of shots. We can check the top goal scorers of Barcelona in 2019/2020:

Player name	Season	0
Lionel Andrés Messi Cuccittini	2019/2020	25
Luis Alberto Suárez Díaz	2019/2020	13
Antoine Griezmann	2019/2020	7
Arturo Erasmo Vidal Pardo	2019/2020	7
Anssumane Fati	2019/2020	5
Arthur Henrique Ramos de Oliveira Melo	2019/2020	2
Clément Lenglet	2019/2020	2
Sergio Busquets i Burgos	2019/2020	2
Frenkie de Jong	2019/2020	1
Ivan Rakitić	2019/2020	1
Jordi Alba Ramos	2019/2020	1
Martin Braithwaite Christensen	2019/2020	1
Nélson Cabral Semedo	2019/2020	1
Ousmane Dembélé	2019/2020	1
Sergi Roberto Carnicer	2019/2020	1

Figure 43 Top goalscorers

- **Results of a team in a season:**

This feature returns the following information in a certain season:

- Number of wins, draws and losses.
- Opponents and results of each match.
- Filtering by wins, draws and losses

Let us take the example of Barcelona in 2014/2015 season:

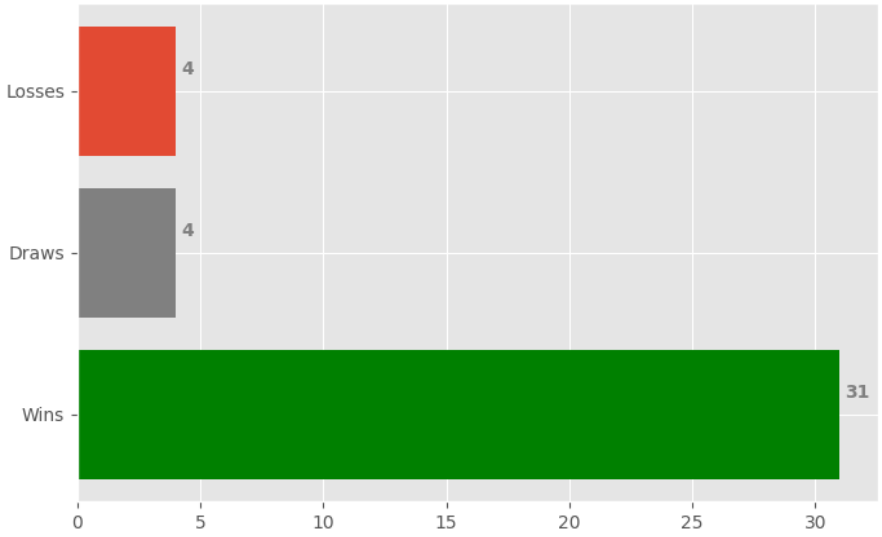


Figure 44 Wins, draws and losses

This graph show number of wins, draws and losses of Barcelona in 2014/2015 season. The following figure shows the results of Barcelona in that season.

Figure 45 Results

And we can filter these results by the buttons in the bottom.

- **Manager**:

We can also display the manager of the team in a given season:

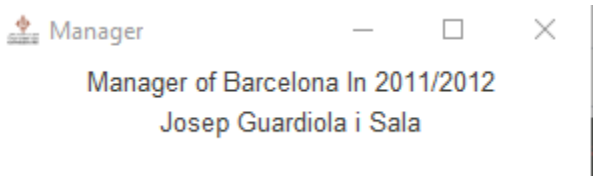


Figure 46 Manager

3. Conclusion

In this last chapter we have demonstrated the implementation of the application and the tools we have used in order to create a football analytics application. In addition to explaining each feature and the different methods we used to achieve our purpose.

General Conclusion:

Football analytics has been the focus of professional football teams and academic research in recent years. Enormous progress has been made thanks to the advances in several areas, such as data analytics tools, data collection, and machine learning. With all this progress, there are still many challenges facing analytics in football because of the complexity and variety of the game.

In this research thesis, we have created a desktop application that analyzes a football match; based on event data and match data, we have used sample data collected by StatsBomb. In order to create this application, we have used the most common data analysis language: python, due to its vast libraries support, also we have used a supervised learning model to predict the probability of a shot resulting in a goal; Gradient Boosting, and finally we tuned its parameters using GridSearchCV.

This application still has a big margin of improvement. There are still many columns in the data that could be used to derive more conclusions, such as timestamp of events, we used it to create possession chain but timestamp column could be used for further and deep analysis. The application could have the following enhancements:

- One of our main goals with the application is creating analysis using different data sources and not only StatsBomb.
- Add options of finding a match.
- Collect data and produce analysis in real time.
- Improve Expected goals model.
- Write the Expected assists metric.

At this level we have reached the end of our research thesis. We have presented the history of analytics in football and its importance. Then we discussed how it is used in professional football teams to improve performance, discover gem players and prevent injuries. After that, we presented the tools we have used to create the application and the machine learning model used to predict the expected goals metric. Finally, we have consulted the implementation of the application and the results.

References:

Sumpter, David. Soccermatics: mathematical adventures in the beautiful game. Bloomsbury Publishing, 2016.

StatsBomb Open Events Structure and Data Specification.

[1] Nature. (2019). *A public data set of spatio-temporal match events in soccer competitions.* https://www.nature.com/articles/s41597-019-0247-7 [Accessed 01 July 2021]

[2] Scisports. (2021). *State of the football analytics industry in 2021*. https://www.scisports.com/state-of-the-football-analytics-industry-in-2021/ [Accessed 22 June 2021]

[3] Soccerment. (2021). *The Growing Importance of Football Analytics*. https://soccerment.com/the-importance-of-football-analytics/ [Accessed in 5 June 2021]

[4] Ac Milan. VISMARA. *The base for the Rossonero Youth Sector, where the players of the future develop*. https://www.acmilan.com/en/club/venues/vismara/milan-lab [Accessed in 5 June 2021]

[5] FourFourTwo, (22 Dec 2017). *The Numbers Game | How Data Is Changing Football | Documentary* [Video]. Youtube. https://www.youtube.com/watch?v=lLcXH_4rwr4

[6] Tutorialspoint. (n.d.). *SDLC – V-Model*. https://www.tutorialspoint.com/sdlc/sdlc_v_model.htm https://www.bbc.com/news/business-56164159 [Accessed in 02 July 2021]

[7] Visual-paradigm. (n.d.). *What is Unified Modeling Language (UML)?* https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/ [Accessed in 15 July 2021]

[8] Statsbomb. (n.d.). https://statsbomb.com/ [Accessed in 02 April 2021]

[9] McKinney, Wes. (2021*). Python for data analysis: Data wrangling with Pandas, NumPy, and IPython.* O'Reilly Media, Inc.

[10] Python. (n.d.). *Tkinter Documentation.* https://docs.python.org/3/library/tkinter.html [Accessed in 01 May 2021]

[11] Seaborn (2020) https://seaborn.pydata.org/ [Accessed in 01 July 2021]

[12] Pypi. (24 Feb 2020). https://pypi.org/project/pandastable/ [Accessed in 15 May 2021]

[13] Wikipedia. (14 July 2021). *Sublime Text.* https://en.wikipedia.org/wiki/Sublime_Text

[14] Soccermatics Github repository. (25 Mar 2020). https://github.com/Friends-of-Tracking-Data-FoTD/SoccermaticsForPython

[15] StatsBomb Documentation. (2019). *Open Data Events v4.0.0* https://github.com/statsbomb/open-data/tree/master/doc

Appendices:

Appendix 1: Data Architecture

Column	Type	Child(/s)	Child(/s) Type	Description	Values	Value Description
id	Uuid (Universally unique identifier)			The unique identifier for each event	e.g.“0052d1b5-e2b0-4629-bbea-c18c884ab103”	
index	Integer			Order of the event	e.g.1-#of events	
period	Integer			The part of the match the timestamp relates to	1	1st Half
					2	2nd Half
					3	1st Half Extra Time
					4	2nd Half Extra Time
					5	Penalty Shootout
timestamp	timestamp			Time recorded to the millisecond	e.g., 00:00:06.293	
minute	integer				e.g., 40	
second	Integer				e.g., 15	
type	object	id / name	integer/text	Id / name of the event type	42/Ball Receipt	The receipt or intended receipt of a pass
					2/Ball Recovery	An attempt to recover a loose ball

					3/ Dispossessed	Player loses ball to an opponent as a result of being tackled by a defender without attempting a dribble
					4/ Duel	Duel between two opposing players
					5/Camera On	Camera stop(replay)
					6/ Block	Blocking the ball
					8 / Offside	
					9 / Clearance	Clearing the danger from the penalty area
					10/ Interception	Preventing an opponent's pass from reaching their teammates by moving to the passing lane/reacting to intercept it.
					14 / Dribble	An attempt to dribble.
					16 / Shot	Made with any legal part of the body.
					17 / Pressure	Applying pressure to an opposing player.
					18/Half Start	Signals referee whistle to start a match period.
					19/Substitution	
					20/ Own Goal Against	An own goal scored against the team.

					21/Foul Won	Won a free-kick or penalty after being fouled by an opposing player.
					22/Foul Committed	Any infringement that is penalized as foul play by a referee.
					23/Goalkeeper	Actions that can be done by the goalkeeper.
					24/Bad Behavior	When a player receives a card due to an infringement outside of play.
					25 /Own Goal For	An own goal scored for the team.
					26/Player On	A player returns to the pitch after a Player Off event.
					27 /Player Off	A player goes/ is carried out of the pitch without a substitution.
					28 / Shield	Player shields the ball going out of bounds to prevent the opponent from keeping it in play.
					30 / Pass	Ball is passed between teammates.
					33 / 50/50	2 players challenging to recover a loose ball.
					34/Half End	Signals the referee whistle to finish a match part.
					35/StartingXI	

					36/Tactical Shift	Indicates a tactical shift made by the team, shows the players' new positions and the team's new formation.
					37 / Error	When a player is judged to make an on-the-ball mistake that leads to a shot on goal.
					38/ Miscontrol	Player loses ball due to bad touch
					39/Dribbled Past	Player is dribbled past by an opponent.
					40/Injury Stoppage	A stop in play due to an injury.
					41/Referee Ball-Drop	Referee drops the ball to continue the game after an injury stoppage.
					43/Carry	A player controls the ball
possession	integer			Indicates the current unique possession in the game. A single possession denotes a period of play in which the ball is in play and a single team is in control of the ball.	e.g., 1 - # of unique possessions	New possessions are triggered after a team demonstrates they've established control of the ball.
possession_team	object	id	integer	The ID of the team that started this possession in control of the ball. Note that this will appear even on opposition events like tackles attempted during the possession.		
play_pattern	object	id / name	integer / text	Id /name of the play pattern relevant to this event.	1 / Regular Play	The event was not part of any of the following play_patterns

					2/From Corner	
					3/From Free Kick	
					4/From Throw In	
					5 / Other	
					6/From Counter	The event was part of a counter attack: ● The possession started with an open play turnover outside the counter-attacking team's final third. ● The possession was at least 75% direct towards goal (as measured by our possession chain metrics). ● The counterattack travelled at least 18 yards towards goal. ● This definition is not part of collection and is derived from the logic above.
					7/From Goal Kick	
					8/From Keeper	
					9/From Kick Off	
team	object	id / name	integer	Id / Name of the team this event relates to.	e.g., 1 / "Arsenal"	
player	object	id / name	integer / text	Id / Name of the player this event relates to.	e.g., 5079 / "Zlatan"	

position	object	id / name	integer / text	Id / Name of the position the player was in at the time of this event..	e.g., 1/Goalkeeper”	Refer to Appendix 3 below for more information.
location	array [x,y]			Array containing the x and y coordinates of the event.	e.g., the center of the field is [60,40]	
duration	decimal			If relevant, the length in seconds the event lasted.	Time in seconds.	
under_pressure	boolean			The action was performed while being pressured by an opponent.	TRUE	Refer to Appendix 4 below for more information.
off_camera	boolean			The event occurred while the camera was off.	FALSE, TRUE	
out	boolean			the ball is going out of bounds.	TRUE	
related_events	array			A comma separated list of the Ids of related events.		
tactics	object	formation	text	The formation item describes the formation being used.	e.g., 343	
		lineup	array	The lineup item describes the players and their positions.		

Table 3 Full data architecture

Appendix 2: Pitch Coordinates

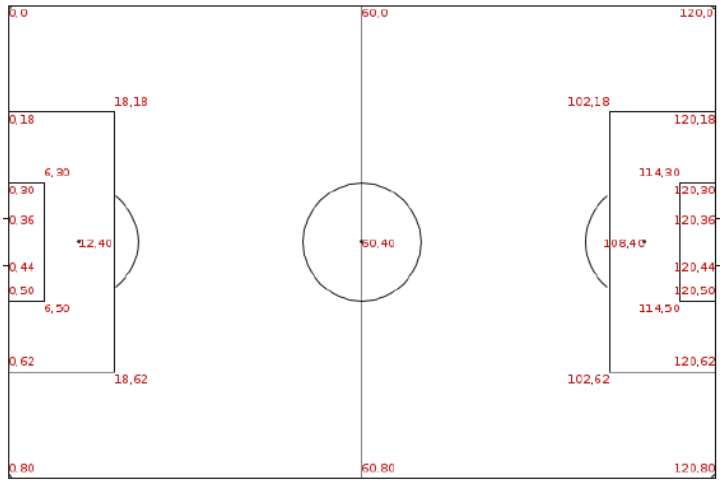


Figure 47 Pitch coordinates

Appendix 3: Tactical Position Guide

Position Number	Position Abbreviation	Position Name	Position Number	Position Abbreviation	Position Name
1	GK	Goalkeeper	15	LCM	Left Center Midfield
2	RB	Right Back	16	LM	Left Midfield
3	RCB	Right Center Back	17	RW	Right Wing
4	CB	Center Back	18	RAM	Right Attacking Midfield
5	LCB	Left Center Back	19	CAM	Center Attacking Midfield
6	LB	Left Back	20	LAM	Left Attacking Midfield
7	RWB	Right Wing Back	21	LW	Left Wing
8	LWB	Left Wing Back	22	RCF	Right Center Forward
9	RDM	Right Defensive Midfield	23	ST	Striker
10	CDM	Center Defensive Midfield	24	LCF	Left Center Forward
11	LDM	Left Defensive Midfield	25	SS	Secondary Striker
12	RM	Right Midfield	13	RCM	Right Center Midfield

14	CM	Center Midfield	

Table 4 Tactical position guide

Positions on the field:

Figure 48 Position guide

Appendix 4: Pressure

Calculated as every on-the-ball event that overlaps the duration of a pressure event. For example, if a pressure event appears before a pass, and the pressure's timestamp plus its duration encompasses the pass's timestamp, that pass is said to have been made under pressure. If a pressure event occurs after a pass, but before the end of the pass (as calculated by using its duration), that pass is said to have been received under pressure.

Appendix 5: Competition Stages:

Competition Stage ID	Competition Stage Name
1	Regular Season
2	Play-In Round
6	Europa League Play-offs - Semi-finals
8	MLS Cup - Conference Semi-finals
9	3rd Qualifying Round
10	Group Stage
11	Quarter-finals
12	Europa League Play-offs - Finals
13	16th Finals
14	Promotion Play-offs - Final
15	Semi-finals
18	Promotion Play-offs - Semi-finals

19	Preliminary Round
20	2nd Round
21	Europa League Play-offs - Quarter-finals
22	2nd Qualifying Round
23	MLS Cup - Conference Finals
24	Promotion Play-offs - 1st Round
25	3rd Place Final

Table 5 Competition stages

Appendix 6: Cutbacks:

Cutbacks are low or ground passes that originate in zone A and end in zone B.

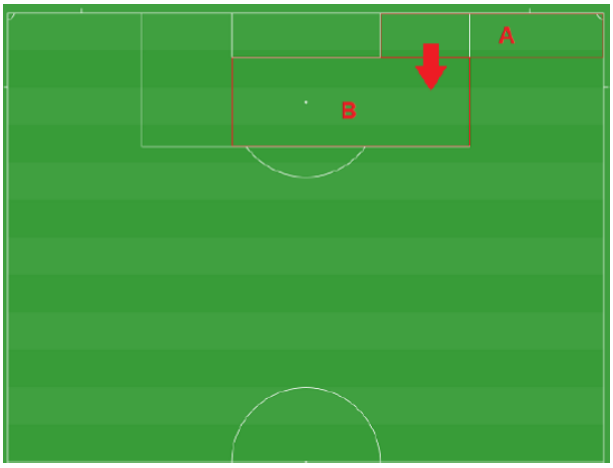


Figure 49 Cutbacks

Appendix 7: Cross:

A pass is marked as a cross if it originates from any of the following:

- Attacking zones (on either side of the pitch):

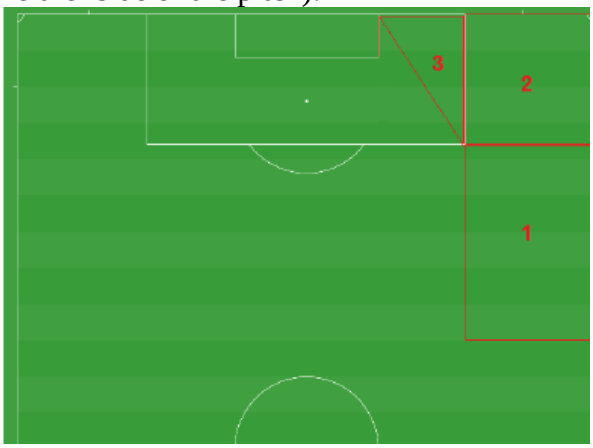


Figure 50 Cross 1

- Intersects the following zone:

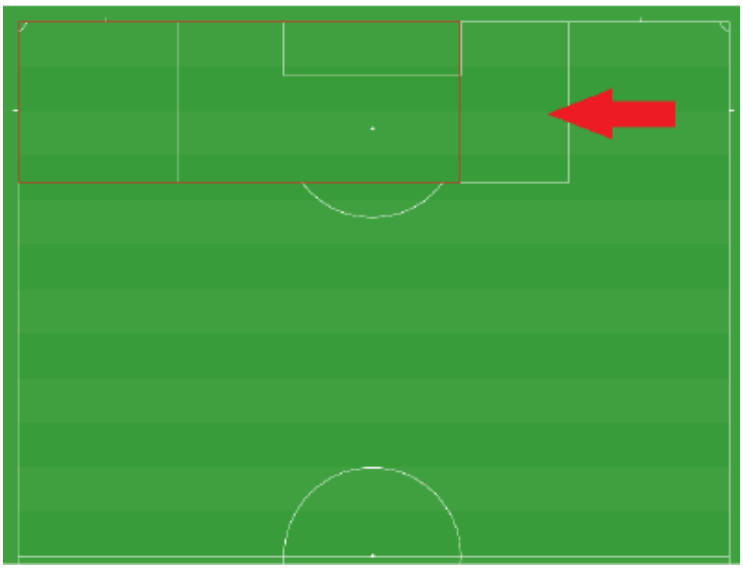


Figure 51 Cross 2

Appendix 8: Goal coordinates

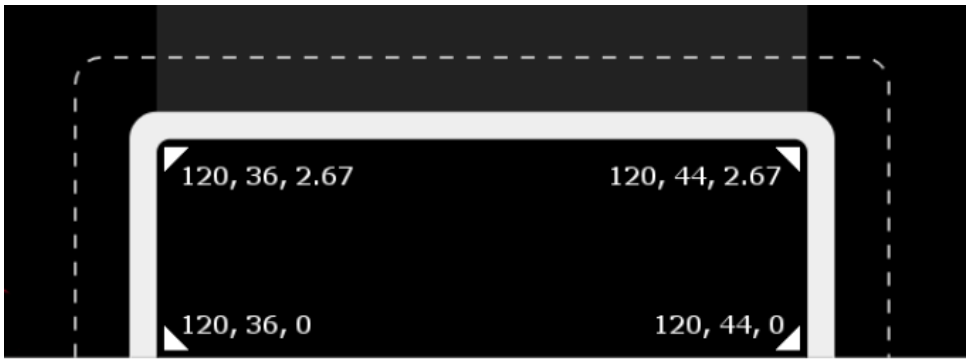


Figure 52 Goal coordinates

Appendix 9: Locations plot code, inspired by 'Friend of Tracking':

https://github.com/Friends-of-Tracking-Data-FoTD/SoccermaticsForPython

```
def createPitch(length,width,linecolor):
    fig=plt.figure()
    #fig.set_size_inches(7, 5)
    ax=fig.add_subplot(1,1,1)
    #Pitch Outline & Centre Line
    plt.plot([0,0],[0,width], color=linecolor)
    plt.plot([0,length],[width,width], color=linecolor)
    plt.plot([length,length],[width,0], color=linecolor)
    plt.plot([length,0],[0,0], color=linecolor)
    plt.plot([length/2,length/2],[0,width], color=linecolor)
    #Left Penalty Area
    plt.plot([18 ,18],[(width/2 +18),(width/2-18)],color=linecolor)
    plt.plot([0,18],[(width/2 +18),(width/2 +18)],color=linecolor)
    plt.plot([18,0],[(width/2 -18),(width/2 -18)],color=linecolor)
    #Right Penalty Area
    plt.plot([(length-18),length],[(width/2 +18),(width/2
+18)],color=linecolor)
    plt.plot([(length-18), (length-18)],[(width/2 +18),(width/2-
18)],color=linecolor)
    plt.plot([(length-18),length],[(width/2 -18),(width/2 -
18)],color=linecolor)
    #Left 6-yard Box
    plt.plot([0,6],[(width/2+7.32/2+6),(width/2+7.32/2+6)],color=linecolor)
    plt.plot([6,6],[(width/2+7.32/2+6),(width/2-7.32/2-6)],color=linecolor)
    plt.plot([6,0],[(width/2-7.32/2-6),(width/2-7.32/2-6)],color=linecolor)
    #Right 6-yard Box
    plt.plot([length,length-
6],[(width/2+7.32/2+6),(width/2+7.32/2+6)],color=linecolor)
    plt.plot([length-6,length-6],[(width/2+7.32/2+6),width/2-7.32/2-
6],color=linecolor)
    plt.plot([length-6,length],[(width/2-7.32/2-6),width/2-7.32/2-
6],color=linecolor)
    #Prepare Circles; 10 yards distance. penalty on 12 yards
    centreCircle = plt.Circle((length/2,width/2),10,color=linecolor,fill=False)
    centreSpot = plt.Circle((length/2,width/2),0.8,color=linecolor)
    leftPenSpot = plt.Circle((12,width/2),0.8,color=linecolor)
    rightPenSpot = plt.Circle((length-12,width/2),0.8,color=linecolor)
    #Draw Circles
    ax.add_patch(centreCircle)
    ax.add_patch(centreSpot)
    ax.add_patch(leftPenSpot)
    ax.add_patch(rightPenSpot)
    #Prepare Arcs
    leftArc = Arc((11,width/2),height=20,width=20,an-
gle=0,theta1=312,theta2=48,color=linecolor)
    rightArc = Arc((length-11,width/2),height=20,width=20,an-
gle=0,theta1=130,theta2=230,color=linecolor)
    #Draw Arcs
    ax.add_patch(leftArc)
    ax.add_patch(rightArc)
    #Tidy Axes
    plt.axis('off')
    return fig,ax
```

Appendix 10: Possession chain field, inspired by ‘Friend of Tracking’:

https://github.com/Friends-of-Tracking-Data-FoTD/SoccermaticsForPython

```
class Field ():
    def __init__(self):
        figure = graph_objects.Figure()
     figure.update_layout(width=900*1.388, height=900, autosize=False, plot_bgcolor="white'
        figure.update_xaxes(range=[-0.03, 1.03])
        figure.update_yaxes(range=[-0.03, 1.03])
        self.figure = figure
        self.__draw_full()
    def add_title(self, main_title: str):
        self.figure.update_layout(title={'text': main_title, 'x': 0.48, 'y':
0.91,'xanchor': 'center', 'yanchor': 'top'})
    def save(self, name: str):
        offline.plot(self.figure, filename=f"{name}", auto_open=False)
    def __draw_full(self):
        self.figure.add_shape(type="rect", x0=0, y0=0, x1=1, y1=1)
        self.figure.add_shape(type="line", x0=0.5, y0=0, x1=0.5, y1=1)
        circle_radius = 0.0915
        self.figure.add_shape(type="circle", x0=0.5-circle_radius*0.72, y0=0.5 -
circle_radius, x1=0.5+circle_radius*0.72, y1=0.5+circle_radius)
        self.figure.add_shape(type="rect", x0=0, y0=0.211, x1=0.170, y1=0.789)
        self.figure.add_shape(type="rect", x0=0, y0=0.368, x1=0.058, y1=0.632)
        self.figure.add_shape(type="rect", x0=-0.01, y0=0.447, x1=0.0, y1=0.553)
        self.figure.add_shape(type="rect", x0=0.83, y0=0.211, x1=1.0, y1=0.789)
        self.figure.add_shape(type="rect", x0=0.942, y0=0.368, x1=1.0, y1=0.632)
        self.figure.add_shape(type="rect", x0=1.0, y0=0.447, x1=1.01, y1=0.553)
```

Appendix 11: Half of the field

```
def createGoalMouth():
    #Create figure
    fig=plt.figure()
    ax=fig.add_subplot(1,1,1)
    linecolor='black'
    plt.plot([0,65],[0,0], color=linecolor)
    plt.plot([65,65],[50,0], color=linecolor)
    plt.plot([0,0],[50,0], color=linecolor)
    plt.plot([12.5,52.5],[16.5,16.5],color=linecolor)
    plt.plot([52.5,52.5],[16.5,0],color=linecolor)
    plt.plot([12.5,12.5],[0,16.5],color=linecolor)
    plt.plot([41.5,41.5],[5.5,0],color=linecolor)
    plt.plot([23.5,41.5],[5.5,5.5],color=linecolor)
    plt.plot([23.5,23.5],[0,5.5],color=linecolor)
    plt.plot([41.5-5.34,41.5-5.34],[-2,0],color=linecolor)
    plt.plot([23.5+5.34,41.5-5.34],[-2,-2],color=linecolor)
    plt.plot([23.5+5.34,23.5+5.34],[0,-2],color=linecolor)
    leftPenSpot = plt.Circle((65/2,11),0.8,color=linecolor)
    ax.add_patch(leftPenSpot)
    leftArc =
Arc((32.5,11),height=18.3,width=18.3,angle=0,theta1=38,theta2=142,color=linecolor)
    ax.add_patch(leftArc)
plt.axis('off')
    return fig,ax
```

Appendix 12: Possession chain

```
def add_possession_chain_sb(figure, chain, team_name: str, opacity: int = 1.0):
    this_team_color = "#1F77B4"
    other_team_color = "#FF7F0E"
    chain_x = []
    chain_y = []
    text = []
    colors = []
    outcomes = []
    times = []
    counter = 0
    try:
        for idx, row in chain.iterrows():
            event_name = row['type_name']
            event_loc = row['location']
            # Transform  coordinates
            x = round((event_loc[0] * (100 / 120)) / 100, 3)
            y = round(((80 - event_loc[1]) * (100 / 80)) / 100, 3)  # Reverse
            color = this_team_color
            if x in chain_x and y in chain_y:
                x = x + random.choice([-0.001, 0.001])
                y = y + random.choice([-0.001, 0.001])
            if row['possession_team_id'] != row['team_id']:
                team_name = row['team_name']
                color = other_team_color
                x = round(1.000 - x, 3)
                y = round(1.000 - y, 3)
            chain_x.append(x)
            chain_y.append(y)
            if row['possession_team_id'] == row['team_id']:
                team_name = row['team_name']
                text.append(f"{counter}.{event_name}")
            else:
                text.append(f"{counter}.{event_name}")
            colors.append(color)
            times.append(f"{row['minute']}{row['second']}")
            outcomes.append('circle')
            counter += 1
    except Exception as err:
        print(err)
    line_color = "#7F7F7F"
    figure.add_trace(
        {'mode': 'markers+lines+text',
         'textposition': 'top center',
         'type': 'scatter',
         'x': chain_x,
         'y': chain_y,
         'hovertemplate': "Time: %{hovertext}<br>Event: %{text}",
         'text': text,
         'opacity': opacity,
         'hovertext': times,
         'marker': {'color': colors, 'size': 8, 'symbol': outcomes},
         'line': {'color': line_color, 'dash': 'solid'},
         'name': "{} {}:{} ".format(chain.iloc[0]['minute'],
chain.iloc[0]['second'],team_name)
         })
```

Appendix 13: Angle of the shot demonstration

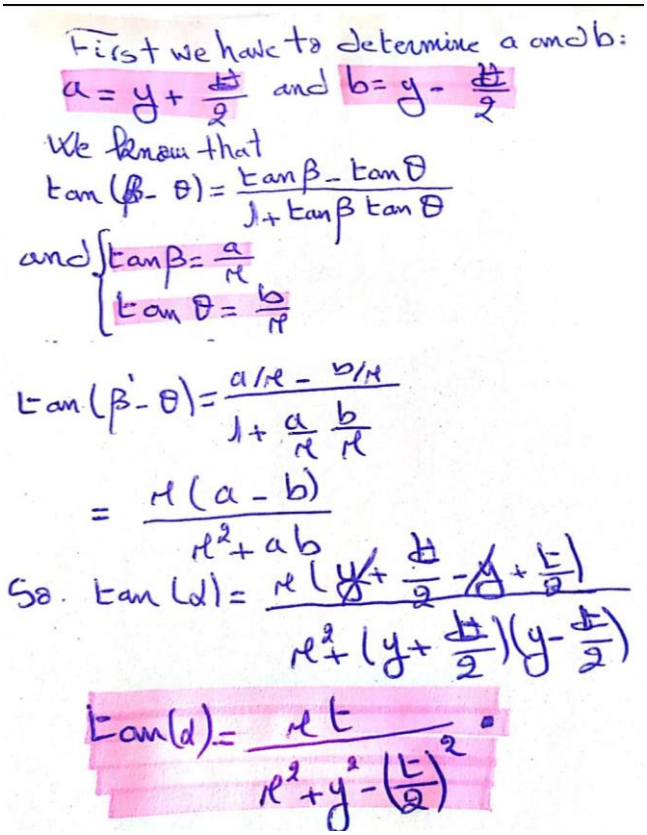


Figure 53 Angle demonstration

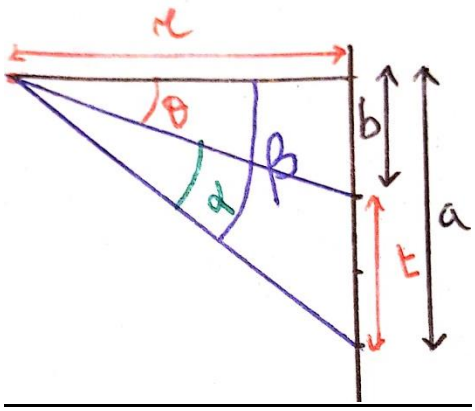


Figure 54 Angle

Appendix 14: Gradient boosting hyperparameters

Tree specific parameters:

- max_features
- min_samples_split
- max_depth
- min_samples_leaf

Boosting parameters:

- learning_rate
- n_estimators
- subsample

Miscellaneous parameters:

- loss
- init
- random_state
- verbose