1 2 90

UNIVERSIDADE Đ
COIMBRA

Miguel Paulo Martins Marques

SUBGROUP DISCOVERY IN SOCCER DATA

Dissertation in the context of the Master in Data Science and Engineering, advised by Professor Pedro Henriques Abreu (PhD.), Professor Carlos Soares (PhD.), Professor Maurice Van Keulen (PdD.), and René Hoevennar (Sports Scientist) and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

July of 2022

Faculty of Sciences and Technology

Department of Informatics Engineering

Subgroup Discovery in Soccer Data

Miguel Paulo Martins Marques

Dissertation in the context of the Master in Data Science and Engineering, advised by Professor Pedro Abreu (PhD.), Professor Carlos Soares (PhD.), Professor Maurice Van Keulen (PhD.), and René Hoevenaar (Sports Scientist) and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

July 2022

1 2 9 0

UNIVERSIDADE Ð
COIMBRA

This page is intentionally left blank.

Este trabalho foi desenvolvido em colaboração com:

This work was developped in collaboration with:

UNIVERSITY
OF TWENTE.

This page is intentionally left blank.

Without the element of enjoyment,
it is not worth trying to excel at anything.

MAGNUS CARLSEN

This page is intentionally left blank.

Agradecimentos

Agradeço em primeiro lugar aos meus pais e ao meu irmão por todo o apoio, motivação e confiança depositada em mim não só durante a realização desta tese, mas durante toda a minha vida académica. Sem vocês não seria metade do que sou hoje.

Agradeço do fundo do meu coração ao meu orientador Pedro Abreu pelo apoio, paciência, honestidade e pelas horas e horas de boa música que me foi enviando desde o início da Tese. Ao Carlos Soares por me ter arranjado esta oportunidade única que acabar a tese na Holanda. Ao Maurice pelo apoio e pela dedicação para que não me faltasse nada durante a minha estadia na Holanda. E ao René pelas inúmeras reuniões de última hora e constante paciência para me transmitir os seus conhecimentos na área de futebol (e pelos bilhetes para os jogos do FC Twente #RumoALigaEuropa).

Quero agradecer também principalmente ao Mega, Isabel, Cardoso, Rafa, Paulinho, Lecu, Bábá e JPS por serem a minha segunda família durante estes últimos anos. Tanto em Viseu, Coimbra, Porto, Lisboa ou à distância, o vosso constante apoio foi essencial no meu percurso.

Por fim, quero agradecer aos PDA 2019 por todos os momentos incríveis que passamos juntos e por me manterem a sanidade mental e momentos de maior stress. Sem vocês a Holanda não teria tanto encanto.

This page is intentionally left blank.

Abstract

Soccer is the most played sport in the world today. Due to its social impact and the vast investments involved, a victory in a game represents much more than three points. Therefore, any additional information provided to all the stakeholders (e.g., managers, players, agents) can be crucially important in winning a match.

Over the years, several machine learning techniques have been employed in soccer data to extract standard behaviours of players and teams, including Subgroup Discovery. Subgroup Discovery techniques aim to find subsets in which the distribution of a property of interest significantly differs from the whole population, i.e., extracting unusual patterns within frequent actions. The vast majority of Subgroup Discovery applications employed in soccer use tracking data and feature engineering. However, most works found in the literature only consider binary targets within the soccer domain. To address this limitation, the main goal of this work is to find subgroups (not only with binary targets) in the Spatio-temporal space of soccer actions that lead to a goal and to understand the characteristics and impact on the play assigned to each subgroup. To reach it, we proposed using two types of targets, a binary and a numerical one. The binary target consists of whether a play ended up in a goal or a miss, and the numerical target consists of the prediction of additional information called Expected Goals (xG).

In this thesis, two experiments were performed, a preliminary one where we performed a more technical subgroup discovery experiment (with different search strategies and quality functions) with event-stream data from the English Premier League in the 2017/2018 season. Then the main experiment was carried out with both tracking and event-stream data from seasons 2020/2021 and 2021/2022 of the Dutch Premier League Eredivisie.

In the main experiment, we tested multiple existing Subgroup Discovery approaches with Spatio-Temporal characteristics from soccer data. The best subgroups found increased the probability of scoring a goal from 11.5% to 20.0%. We also realised that there are team-specific subgroups, which leads us to conclude that Subgroup Discovery can detect different play styles from different teams.

Keywords

Subgroup Discovery, Soccer, Spatio-Temporal Data, Expected Goals, Data Mining, Data Visualisation

This page is intentionally left blank.

Resumo

Nos dias de hoje, o Futebol é o desporto mais praticado em todo o mundo. Devido ao seu impacto social e aos investimentos avultados que têm sido feitos, a vitória num jogo representa muito mais do que três pontos. Assim, qualquer informação adicional que se possa transmitir aos jogadores pode constituir uma importância vital para a conquista dessa vitoria.

Ao longo dos anos, várias técnicas de *Machine Learning* têm sido aplicadas a dados de futebol com vista à extração de comportamentos padrão de jogadores e de equipas. Incluídas neste conjunto encontram-se as técnicas de *Subgroup Discovery* que permitem encontrar subconjuntos cuja distribuição de uma propriedade de interesse varie comparativamente com a distribuição da população na integra, ou seja, extraindo assim padrões pouco comuns dentro de ações frequentes. Têm sido desenvolvidos vários trabalhos com técnicas de *Subgroup Discovery* ao futebol, onde a grande maioria desses trabalhos utiliza fundamentalmente *Tracking Data* e *Feature Engineering.*

Os trabalhos de *Subgroup Discovery* dentro deste contexto focam-se essencialmente em utilizar técnicas existentes com dados Espaciotemporais de Futebol com o objetivo de encontrar subconjuntos apenas utilizando *targets* binários, assim como jogadas que resultem em ataques perigosos e golos. Para colmatar esta limitação, o principal objectivo deste trabalho é encontrar subgrupos (não apenas com *targets* binários) nas dimensões espaciotemporais das acções de futebol que conduzem a um golo. Assim como compreender as características e o impacto no jogo atribuído a cada subgrupo. De modo a alcançar esse objetivo, propusemos a utilização de dois tipos de *targets*, um binário e um numérico. O *target* binário consiste em saber se uma jogada acabou num golo ou não, e o alvo numérico consiste na previsão de informação adicional chamada *Expected Goals* (xG).

Neste estudo, foram realizadas duas experiências, uma preliminar onde realizámos uma experimentação de *Subgroup Discovery* mais técnica (com diferentes *Search Strategies* e *Quality Function*) com dados de eventos da Premier League Inglesa na época 2017/2018. Finalmente, a experiência principal foi realizada com dados de *Tracking* e de eventos das épocas de 2020/2021 e 2021/2022 da Premier League Holandesa (Eredivisie).

Na experiência principal, testamos múltiplas abordagens de *Subgroup Discovery* existentes com características espaciotemporais a partir de dados de futebol. Os melhores subgrupos encontrados aumentaram a probabilidade de marcar golo de 11,5% para 20,0%. Além disso, também nos apercebemos de que existem subgrupos específicos para certas equipas, o que nos leva a concluir que *Subgroup Discovery* consegue detectar diferentes estilos de jogo de diferentes equipas.

Palavras-Chave

Subgroup Discovery, Futebol, Dados Espacio-Temporais, Expected Goals, Data Mining, Visualização de Dados

This page is intentionally left blank.

Contents

Acronyms

BFS Breadth-First Search. 16, 22

DFS Depth-First Search. 16, 22

FIFA Fédération Internationale de Football Association. 26, 38, 39

SPADL Soccer Player Action Description Language. 27, 36, 37

STARSS Spatio-Temporal Action Rating System for Soccer. 30

VAEP Valuing Actions by Estimating Probabilities. 30, 37

WRAcc Weighted Relative Accuracy. 6, 10, 19–21, 24, 32, 33, 40–42

xG Expected Goals. 30, 31, 55, 57–59, 61, 62, 65, 67

This page is intentionally left blank.

List of Figures

This page is intentionally left blank.

List of Tables

This page is intentionally left blank.

Chapter 1

Introduction

Soccer is a collective game where two teams of eleven players try to score at least one more goal than their opponent over a 90-minute period. Being the most popular sport globally, it is played by approximately 250 million players in over 200 countries [1] and due to its popularity, soccer has been used in multiple scenarios such as serving as a motivation for researchers to develop work in the field of artificial intelligence and robotics as is the case of RoboCup [2, 3, 4, 5, 6]. In spite of the social and economic importance of the sport, team management was until very recently not supported by analytic approaches.

Soccer Analysis initialized in 1950 with an English gentleman called Charles Reep who started to record the ratio between attacks and goals in his notebook during a soccer match [7]. His simple analysis escalated to the point where he would spend over 80 hours analysing plays, such as counting the number of passes, distributing the pass sequence among different categories, and reporting the number of goals in each category. When he was working for the Brentford team, his analysis contributed to double the goals scored and ended up winning 13 out of 14 games. This made him the first known soccer data scientist.

The Soccer industry in Europe generated 25.2 billion euros of revenue only in the season of 2019/20 [8]. This business involves investments such as purchasing players, infrastructures and/or technology, leading to high risk but high award profit. With the growth of soccer popularity and technology combined, the analysis of tactics and plays on this subject got much more complex. Nowadays, with the help of sensors and optical tracking camera systems, it is possible to gather most of the information regarding a soccer match for further analysis.

1.1 Context and Motivations

Soccer is a 50/50 game, where half of it is luck and the other half is skill. There is no secret recipe for success locked in data, there is neither a secret formula nor a right answer in soccer, but there is a way of making sure we are asking the right questions [9]. Thus, one the main motivations for this thesis is to find the right questions to ask.

When it comes to analysing soccer data, the complexity becomes much higher due not only to the size of the pitch (105 by 68 meters) but also to the number of players (twenty-two) performing actions simultaneously, with or without the ball and interacting with each other. As time and technology progressed, its complexity has increased also due to the increase of high quality of tracking data and post-game analysis. In addition, soccer in the most competitive leagues (e.g., the English Premier League and the Spanish league La Liga) is a sport where the number of goals is significantly low, which makes it harder to analyse patterns of plays that lead to a goal.

There are two main basic movements in soccer, these being passing and receiving the ball. The other actions, such as shooting and dribbling, derive from these two. Besides the two main movements, there are five time phases in the match: defensive and offensive organisations, the defense-attack and attack-defense transitions, and set-pieces.

To analyse these action and time movements, multiple companies such as Opta Sports [10], Wyscout [11], StatsBomb [12] provide data in the format of event stream logs. Each event-log has information about the action such as time elapsed since the game started, position on the field where the action started and ended, type of action performed by the player (pass, dribble, shot), and the outcome of the action. Nevertheless, handling with event-stream data can be challenging because it contains both continuous (elapsed time, position) and discrete information (action type, player name). However, using only event-stream data may not be enough, since it only provide us the information from the player with the ball's point of view. Tracking data fills that gap because it contains information about the position of every player and ball in the pitch during the whole game. This type of data is also provided by multiple companies such as Tracab [13] and Metrica Sports [14].

In addition, due to the popularity of soccer, there is a lot of literature concerning the analysis and methods to evaluate both teams/players and plays. These approaches range from clustering and dynamic time warping techniques to find a similar sequence of actions in a game and deep reinforcement learning to evaluate the goal impact in specific actions and evaluate players according to the action performed by them [15]. However, an emergent data mining technique called *Subgroup Discovery* can be employed in soccer data to extract knowledge.

Subgroup Discovery is a supervised data mining technique that discovers subgroups from data through a set of explainable rules (e.g., *Body_ Part* = *Right_ Foot* AND *Distance_ To_ The_ Goal* ≤ *10 m.*). Therefore, the subgroups discovered should follow two conditions. First of all, they must be interpretable by the user who is analysing the data. Finally, they need to be interesting according to the criteria chosen by the user (defined by a quality function). Its applications range from medical and marketing scenarios to finding patterns from data collected from children in the school recess in order to find interesting behaviour [16]. A surplus asset from this approach is that due to its explainability, it is possible to understand which characteristics within the actions and time movements in soccer have greater probability of leading to a goal from the attacking team.

Then, in order to employ Subgroup Discovery we need to define a target. We are using two different types of targets: a binary one which indicates if a play ended in a goal or a miss and a continuous one called Expected Goals.

Expected Goals (xG) is one of the most well-known metrics to evaluate the teams' and players' performance in a soccer game (replacing the goal scoring). This metric estimates to calculate a team's chances to score goals according to the features chosen to fit the model. The performance of this model varies not only with features available, but also with the type of data (discrete or continuous) used to train the model. Even though this can be seen as a regression problem, in this work a novelty approach has been proposed to adapt a binary target to an approximation coming from a machine learning model presented in Equation 1.1. We are using binary classifiers (where the classes are goal and miss), and then we only take into consideration the probability from the goal class. This way, given a specific play, we can evaluate the probability of it resulting in a goal and it can be represented as:

$$\dot{y} = model(train(x \to y))(x) \tag{1.1}$$

Where $model(train(x \to y))$ represents a machine learning model trained with the features x and the binary target y. The variable $\dot{y}$ represents the continuous target generated. Its values varies from 0 (no goal) to 1 (goal). However, a missed goal is usually worth more than 0, and a goal scored is usually worth less than 1, depending on the model that is used.

After researching the existing subgroup discovery applications in the domain of soccer, we found that most studies implement subgroup discovery in binary targets, such as, goal/miss or if an attack was successful or unsuccessful. In this thesis we decided not only to consider binary targets as it has been done until now but also to experiment with numerical targets using xG as a target.

1.2 Goals

Since the application of Subgroup Discovery techniques to Spatio-Temporal Soccer data is at a rather preliminary stage, we consider it relevant to expand this field for the purpose of finding reliable ways to extract knowledge that can be actionable by team coaches in order to improve the team's performance.

The main goal of this thesis is to **find subgroups in the Spatio-temporal space of soccer actions that lead to a goal and understand the characteristics and impact on the play assigned to each subgroup**. In order to reach this goal, four *Research Questions* have been formulated:

1. Does the use of different search strategies and quality measures lead to subgroups that represent substantially different knowledge?
2. Can we identify interesting subgroups combining information from different sources?
3. Is it possible to use a subgroup discovery approach to find knowledge that is specific to a particular team or league?
4. Are the subgroups discovered related with the five time phases (defensive and offensive organisations, the defense-attack and attack-defense transitions, and stop balls) in the game or with the time where those occurred (first or second half)?

1.3 Research Contributions

The work developed during this thesis resulted in the following contributions:

- Miguel Marques, Pedro H. Abreu, Carlos Soares, Maurice Van Keulen, René Hoevenaar."A 2-steps Pipeline Approach to explain soccer performance using Expected Goals". CIKM2022 - ACM International Conference on Information and Knowledge Management (Submitted on the 19th May 2022).
- Miguel Marques, Pedro H. Abreu, Carlos Soares, Maurice Van Keulen, René Hoevenaar, "Subgroup Discovery in Soccer Data: A Systematic Review" (in preparation to be submitted to European Journal of Sport Science).

1.4 Document Structure

The remaining contents of this document are organised as follows: Chapter 2 overviews the state of the art concerning Subgroup Discovery, Spatio-Temporal analysis in soccer data, and the review of the literature concerning these two areas. Chapter 3 shows the preliminary work done during the elaboration of the State of the Art. In Chapter 4 we present the experimental setup and its results are presented in Chapter 5. Finally, Chapter 6 concludes the work done in this thesis and also provides some directions for future work.

Chapter 2

State of the Art

This chapter provides an explanation about both Subgroup Discovery, and the analysis of spatial-temporal data in Soccer.

The chapter is organised as follows. Starting with Subgroup Discovery, the definition and its main components, such as search space and target concept will be presented in Section 2.1.1. Following this core concepts explanation, Section 2.1.3 overviews multiple quality functions found in the literature to be used by the Subgroup Discovery Algorithm (Section 2.1.8). Section 2.2 presents Spatio-temporal data analysis in soccer and how different approaches extract information from this type of data. Finally, the third Section of this Chapter presents the related work found in the literature concerning the employment of Subgroup Discovery techniques in soccer data.

2.1 Subgroup Discovery

The concept of Subgroup Discovery was initially formulated by Klösgen in EXPLORA [17] and by Wrobel in MIDOS [18]. It consists in a supervised data mining technique that aims at finding unusual statistical distributions in a user-defined target variable (or variables [19]) within a subset of data.

The goal of Subgroup Discovery is to find subsets of the population with a significantly different distribution in the target feature. E.g., with a dataset with multiple weather conditions where *Going for a bike ride* is the binary variable of interest, an hypothetical subgroup could be represented by the following selectors: *Weather = Clean* AND *Wind_ Speed <= 20 km/h.*

However, Subgroup Discovery does not focus on predicting the dependent variable (which we call *target*) based on the independent variables (also called *selectors*, once their purpose to the task is to describe the Subgroup with conditions understandable to the human). Instead, its main focus is to find subsets of the original dataset that satisfies the quality functions (Section 2.1.3).

2.1.1 Subgroup Discovery Main Components

The of Subgroup Discovery tasks' concept can be defined by the following four components [20]:

- The *Dataset* in which the variables can be numerical, categorical, or ordinal. Subgroup Discovery is not limited to the data in the form of a single relational table, Stefan Wrobel introduced this topic with the MIDOS approach in [18], proving that is possible to employ Subgroup Discovery in multiple tables;

- The *Search Space* is set by the descriptions in the database, i.e, the enumeration of the possible subsets within the dataset. The search space depends both on the size of the dataset and its type of variables. The size of the search space can be a bottleneck to the task of subgroup discovery since doing an exhaustive search in all the possible combinations of the descriptors of the search is a NP-Problem [21]. The most common setting considers only conjunctive combinations of selectors since it facilitates the user's understanding. It is crucial to avoid redundancy between selectors, e.i., in the scenario where a subgroup descriptor D can be defined by a set of selectors $S1, S2, .., Sn$. If S1 is $varA > x$ and S2 is $varA > y$, where $x \neq y$, if $D = S1 \wedge S2$, whatever the values of x and y are, one of the selectors will not bring any relevant information to the subgroup;

- The *Target* concept specifies the *property of interest* in the discovery task, i.e, the variable which depends on the remaining features. In most cases, the target is binary. However, it can be categorical, numerical, or complex in the cases where there are more than one target, such as in Exceptional Model Mining [19]. Note that a categorical target can be easily converted into a binary target in most cases;

- *Quality measures* define a set of selection criteria that depends on the target chosen by the user. Given a subgroup, the quality function quantifies the deviation in the subgroup's property of interest in relation to the whole population. The quality measure impacts directly the discovery of subgroups and must be chosen carefully. The type of target (binary, numerical, complex) has a significant impact on the quality measure, e.g., for binary targets, we can use Weighted Relative Accuracy (WRAcc) [22], however, if we are dealing with numerical targets, it is necessary to use Numeric weighted relative accuracy, an adaptation of WRAcc for numerical targets.

2.1.2 Domain Knowledge Concept

Regardless of the results discovered, the final decision has to be made by the domain specialist. Therefore, it is crucial to analyse the subgroups in order to understand if they are interesting and useful to the domain. This way, integrating background knowledge into the pipeline can improve the results. Within the pipeline, it can be applied in the post-processing, but can also be directly used in the mining process. Using background knowledge in the mining process has the advantage that it can reduce the complexity of the search space by focusing the search on variables considered more relevant for the expert and removing others that may be only adding noise to the solutions. There are three distinct classes of knowledge [23]:

- **Constraint Knowledge** specifies the kind of patterns that are relevant to the user. The given constraints defined by the user can be applied to the data in order to filter information. For example, removing "outliers" by restricting the value range of an independent variable in the search space or aggregating a variable in intervals that makes sense to the user and does not result in information loss;

- **Ontological Knowledge** specifies general properties of the object contained in the domain, such as weights of attributes denoting their importance, information about the relative similarity between attribute values, and abnormality/normality/ordinality information about attributes. Is it commonly used for the development of knowledge systems;
- **Abstraction Knowledge** is given by derived attributes that are inferred from basic attributes or other derived attributes. For example, we can infer the derived attribute *mean speed of a player* if we have the starting and ending position of a player in the field and the *time elapsed* of that action. It aims to increase the quality of the input data of the system.

There are multiple *Data Mining* pipelines found in the literature that integrate Subgroup Discovery in its steps.

It is essential to understand that Subgroup Discovery should not be seen as an isolated automatic mining task once its interestingness is heavily influenced by the end-user.

Taking this into account, a process model for Expert Guided Subgroup has proposed in [24] with the following eight steps: (1) problem understanding, (2) data understanding and preparation, (3) subgroup detection, (4) subgroup subset selection, (5) statistical characterisation of subgroups, (6) subgroup visualisation, (7) subgroup interpretation, and (8) subgroup evaluation (some steps might be performed multiple times in order to increase the quality of the subgroups). Also, in this domain, a more straightforward approach with only three steps was proposed in [23] that includes background knowledge: (1) Discover, (2) Inspect, and (3) Refine. Background Knowledge is used in the third step (Refine) to improve the performance of the next iteration.

2.1.3 Objective Measures

This section presents multiple objective approaches used to evaluate the subgroups discovered. *Objective Measures* can be computed only using the raw data. These measures can be used not only to gather information about the distribution of the property of interest in each subgroups but also to rank the subgroups according to its returned value. The measure used to rank the subgroups candidates is called *Quality Function.*

Due to Subgroup Discovery being a Data Mining approach, it is possible to use general interesting measures within this subject, since these measures can be used for any scenario, regardless of the kind of patterns being mined.

Sheikh L. et al in [25] proposed a total of eight different *objective measures*:

- Support
- Confidence
- Conviction
- Lift
- Leverage
- Coverage
- Correlation

- Odds Ratio

Support

Support measures how frequently a given item appears. Therefore it can be used as a measure of importance/significance.

$$Support(X) = P(X)$$

$Support(X)$ values are in the range $[0, 1]$. If X does not appear in the data, the value is 0, if is appears in all instances, the value is 1.

Confidence

Confidence measures the dependency between a given X and a given Y. It is important to mention that $Confidence(X \rightarrow Y) \neq Confidence(Y \rightarrow X)$

$$Confidence(X \rightarrow Y) = \frac{P(X \cap Y)}{P(X)}$$

$Confidence(X \rightarrow Y)$ values are in range $[0, 1]$. If X and Y are completely independent, the value is 0, on the other hand, if X occurs every time Y occurs, the *Confidence* value is 1.

Conviction

Conviction compares the probability that X appears without Y if they were dependent on the actual frequency of the appearance of X without Y.

$$Conviction(X \rightarrow Y) = \frac{P(X)P(\overline{Y})}{P(X \cap \overline{Y})}$$

$Conviction(X \rightarrow Y)$ values are in range $[0, +\infty]$. If X and Y are independent, the *Conviction* value is equal to 1. If X occurs every time Y occurs, the value is $+\infty$.

Lift

Lift measures how more often X and Y occurs than expected (if they where statistically independent).

$$Lift(X \rightarrow Y) = \frac{P(X \cap Y)}{P(X)P(Y)}$$

$Lift(X \rightarrow Y)$ values are in range $[0, +\infty[$.

If the *Lift* is lower than 1 means that the probability decreases when adding the rule. Consequently, if the *Lift* is higher than 1 means the probability increases when adding the rule. If X and Y are independent, the *Lift* is equal to 1.

Leverage

Leverage measures the difference of X and Y appearing together in the dataset and what would be expected if they were statistically dependent. Using a Leverage threshold is the same as using a frequency threshold. It suffers from the same problem as the *Support* metric, which may ignore rare items in the dataset.

$$Leverage(X \rightarrow Y) = P(X \cap Y) - P(X)P(Y)$$

$Leverage(X \rightarrow Y)$ values are in range $[-1, 1]$. The value is 0 if X and Y are independent.

Coverage

Coverage shows the percentage of values that are covered by a given rule.

$$Coverage(X \rightarrow Y) = \frac{P(X \cap Y)}{P(Y)}$$

$Coverage(X \rightarrow Y)$ values are in range $[0, 1]$. The value is 0 if none of the values are covered by the rule and 1 if all of them are covered.

Correlation

Correlation is a statistical technique also known as dependence that assesses whether and how strongly pair of variables is related. Correlation is able to detect if the relationship between two variables is positive, negative or if there is no relationship.

$$Correlation(X \rightarrow Y) = \frac{P(X \cap Y) - P(X)P(Y)}{\sqrt{P(X)P(Y)(1 - P(X))(1 - P(Y))}}$$

$Correlation(X \rightarrow Y)$ values are in range $[-1, 1]$. It is -1 if the relationship is perfect negative linear and 1 if the relationship is perfect linear. The value 0 means that there are no relationship between the variables.

Odds Ratio

An odds ratio is a statistic measure that quantifies the strength of the association between two variables X and Y.

$$Odds(X \rightarrow Y) = \frac{P(X \cap Y)P(\overline{X} \cap \overline{Y})}{P(X \cap \overline{Y})P(\overline{X} \cap Y)}$$

$Odds(X \rightarrow Y)$ values are in range $[0, +\infty]$. If X and Y are independent the *Odds* value is equal to 0. Otherwise, for strong associations, the value is equal to $+\infty$.

However, these genetic measures may not the best option to use as *Quality Function*, but rather to analyse the subgroups discovered.

Stefan Wrobel in [18] provides the following note when dealing with *Quality Functions*:

"If we want to find the k-best such subgroups, we need a way to measure the quality of candidate groups. To find such evaluation functions, we can build on the evaluation measures that have been defined for propositional algorithms."

The *Quality Function* primary purpose is to rank the subgroups discovered in an efficient and effective way. Subgroup size and the target's statistical distribution are the main factors that influence subgroup interestingness. It is used during the search strategy to rank the discovered subgroups when processing the defined search space. Since the subgroups are ranked according to their quality, subgroups with lower scores might not appear on the top-k subgroups selected by the algorithm.

However, this task can be challenging because most of the quality functions aim to maximise the coverage and unusualness (how different is the distribution from the subgroup's target and the original target) of the subgroup, and this two can be conflicting. As it is mentioned in [26], when a subgroup has extensive coverage, in most cases hinders the finding of unusual subgroups. On the other hand, when the coverage is small, it is easier to find more unusual subgroups.

There are several different approaches in the literature, both taking the type of target and problem domain into account. Our main focus is binary targets in the single-relational case. However, we consider it relevant to enumerate the most used techniques in this domain independently of the nature of the data.

Notations	
Variables	Meaning
P	Relative frequency of the target variable in the total population
s_i	Subgroup
p_i	Relative frequency of the target variable in the Subgroup
N	Size of the total population
n_i	Size of Subgroup
$\overline{p_i}$	Relative frequency of the target variable in the complementary Subgroup
$\overline{n_i}$	Size of the complementary Subgroup

Table 2.1: Formula's Notations

The notations in order to understand the formulas are found in the table 2.1. Notations specific to a certain method will be explained in the method's description.

Weighted Relative Accuracy

The Weighted Relative Accuracy (WRAcc) is by far the most used quality measure found in the literature to evaluate the quality of binary targets. Introduced in the MIDOS [18] approach by Stefan Wrobel and its main goal is to trade off the accuracy of a subgroup against its generality. According to [27], this measure can calculate the *Unusualness* of a rule.

$$q_{WRAcc}(s_i) = \frac{n_i}{N}(p_i - P)$$

Multi-class Weighted Relative Accuracy

Multi-class Weighted Relative Accuracy is an extension of the previous approach to deal with categorical targets. As we can see from the formula, this method basically calculates the mean value from the Weighted Relative Accuracy for each k target variable.

$$q_{MWRAcc}(s_i) = \frac{1}{k}\sum_{j=1}^{k} |q_{WRAccj}(s_i)|$$

There are more variants described in [28, 29], such as (one-vs-rest) weighted multi-class weighted relative accuracy and (one-vs-one) multi-class weighted relative accuracy.

Numeric Weighted Relative Accuracy

In [29] is presented another extension of the Weighted Relative Accuracy to deal with numerical targets, in this approach they use the mean from target in the subgroup $\mu(s_i)$ and from the entire population $\mu(Pop)$ in order to calculate this metric.

$$q_{NWRAcc}(si) = \frac{n_i}{N}(\mu(s_i) - \mu(Pop))$$

Another way to calculate this metric is through the probability distribution of the target, presented in [26].

$$q_{NWRAcc}(si) = \frac{n_i}{N}\int_x |f(x) - f_{sg}(x)|$$

Where $f(x)$ and $f_{sg}(x)$ correspond to the probability density function of a given real-value x for the given dataset and subgroup respectively.

Gamberger and Lavrac Approach [24]

Gamberger and Lavrac maintain that subgroups should have a sufficient large coverage, a positive bias towards the target class coverage and sufficient diverse for detecting most of the target population. Therefore, they proposed the following quality function:

$$q_{TP}(s_i) = \frac{p_i \cdot n_i}{(1 - p_i) \cdot n_i + g}$$

Where the g variable is a *generalisation* parameter defined by the user.

Squared Hellinger Distance

Nanlin Jin, et al proposed a new quality measures to deal with numerical target values called Squared Hellinger in [26]. The subgroups' unusualness is measured on the basis of the probability distribution of its real-valued target. Let x be the real value target and $F(x)$ be the density function of x in the dataset. Therefore, $F_{sgi}(x)$ is density function of x in the subgroup i.

$$q_{SHD}(si) = \frac{n_i}{2N}\int_x (\sqrt{F(x)} - \sqrt{F_{sgi}(x)})^2$$

Chi-squared

The Chi-squared approach can be used both for binary target [17] and categorical (multi-class) [29] targets. This approach relates the statistical significance according to the χ^2 statistical test.

Chi-squared formula for binary targets:

$$q_{\chi^2}(s_i) = \frac{p_i \cdot n_i}{N - p_i \cdot n_i}(p_i - P)^2$$

Here we introduce the notation p_{ij} and P_j that corresponds to the relative frequency of the target variable in the $subgroup_i$ according to the class j and relative frequency of the target variable class j in the total population. With the k variable being the number of classes within the target selected.

Chi-squared formula for Multi-class targets according to [28]:

$$q_{M\chi^2}(si) = \sum_{j=1}^{k} \frac{(n_i \cdot N - n_i \cdot N \cdot P_j)^2}{n_i \cdot N \cdot P_j \cdot (N - n_i)}$$

Binomial test

The Binomial test is a quality measure for targets with binary values. Martin Atzmuller in [30] explains this method as following: the gain in *accuracy* or *unusualness* is calculate by the difference between target shares $p_i - P$ and the *size*, *coverage* or *generality* are covered by the difference between subgroups size n and the size of the total population N. These are the basic parameters that are most frequent used to compare subgroups.

The binomial test formula is given by:

$$q_{BT} = \frac{p_i - P}{\sqrt{P(1-P)}}\sqrt{n_i}\sqrt{\frac{N}{N - n_i}}$$

Kolmogorov-Smirnov

The Kolmogorov-Smirnov significance test can be used as a quality metric for Subgroup Discovery if the property of interest is numerical. This approach basically takes the target distribution t_{dist} and its complement $\bar{t}_{dist}$ in order to verify if they are distributed significantly different. Note that according to Florian Lemmerich in [20], $\Delta_{t_{dist},\bar{t}_{dist}}$ is supremum of the differences in the empirical distribution function between the subgroup and its complement.

$$q_{ks}(si) = \sqrt{\frac{t_{dist} \cdot \bar{t}_{dist}}{N}}\Delta_{t_{dist},\bar{t}_{dist}}$$

Mutual Information

Mutual Information is an information-theoretic approach, it tells us how much knowledge about one variable would be increased by knowing the value of the other. The higher the mutual information, the more interesting the distribution of classes amongst the subgroup and its complement is [28]. This metric can be used both for numerical q_{NMI} [26, 29] and multi-class targets q_{MMI}[28].

$$q_{MMI}(si) = \sum_{j=1}^{k} \frac{p_{ij} \cdot n_i}{N} log(\frac{p_{ij} \cdot n}{N}) + \frac{P_j \cdot N - p_{ij} \cdot n_i}{N} log(\frac{P_j \cdot N - p_{ij} \cdot n_i}{N}) -$$

$$- \sum_{i=1}^{n} P_j log(P_j) - \frac{n_i}{N} log(\frac{n_i}{N}) - \frac{N - n_i}{N} log(\frac{N - n_i}{N})$$

$$q_{NMI}(si) = \frac{n_i}{N} \int_x f_{sg}(x) log \frac{f_{sg}(x)}{f(x)} dx$$

Mean/Variance/Median-based Quality Measures

This type of measure is used for scenarios where the target in question is numeric. We will not look too much into these quality metrics as they are not part of our focus since we will be dealing with binary target problems.

Mean Quality Measures, as the name implies, take into account the mean of the targets of both the subgroup and the dataset as a whole. Among these we highlight the following:

- Mean Test - Proposed by Kloesgen in [17], this metric takes into account the coverage/size of the subgroup n_i and the average targets of both the subgroups μ_{si} as well as the total population μ_{Pop}:

$$q_{MT}(si) = \sqrt{n_i} \cdot (\mu_{si} - \mu_{Pop})$$

- Impact - This metric is quite similar to the previous one, but it considers the size of the subgroup instead of its square root.

$$q_I(si) = n_i \cdot (\mu_{si} - \mu_{Pop})$$

- Z-score - is a metric that takes into account not only the means mentioned above, but also the standard deviation of the dataset target σ_{Pop}. It is a metric widely used in statistics, and can also be adjusted to the Subgroup Discovery topic.

$$q_z(si) = \frac{\sqrt{n_i} \cdot (\mu_{si} - \mu_{Pop})}{\sigma_{Pop}}$$

Variance-based Quality Measures are metrics that take into account the spread of the distribution, from which we highlight the T-Statistic.

- T-Statistic - proposed in [31] is a metric very similar to Z-score, however, takes into account the standard deviation of the subgroup target distribution σ_{si}. This makes this metric sensitive to the variance of the subgroups. According to [31] this metric should be applied to small subgroups.

$$q_t(si) = \frac{\sqrt{n_i} \cdot (\mu_{si} - \mu_{Pop})}{\sigma_{si}}$$

Following the same logic as the previous two approaches, Median-based Quality Measures take into account the Median from the numerical target. Here we be will emphasise the Median Statistic from [31], this approach uses the median of the dataset to calculate the difference in distributions between the target from the subgroup and from the whole dataset.

$$q_{med}(si) = \frac{(p_l - P_l)^2}{P_l} + \frac{(p_s - P_s)^2}{P_s}$$

Where p_l and p_s are the frequencies of individuals in the subgroup that are larger than median and equal or smaller than the median respectively. Therefore, P_l and P_s are the frequencies of individuals in the dataset that are larger than median and equal or smaller than the median respectively. The returned value from this quality function can range between zero and $+\infty$ and value is zero when individuals are distributed equally between the median.

ROC AUC

This *Rank-based measure* can be used for ordinal targets. This metric is more communally used to compare the performance of classifiers, however, Pieters B. et al proposed in [31] an extension of this approach in order to be applied in the context of subgroup discovery. Therefore , this measure is very useful to verify the sparseness of the subgroup individuals in the dataset.

$$q_{roc} = \frac{\sum_{j=1}^{k} \overline{t_j} - \frac{\overline{n_i} \cdot (n_i - 1)}{2}}{\overline{n_i} \cdot n_i}$$

Where n_i and $\overline{n_i}$ are the size of the subgroup and its complement respectively. $\sum_{j=1}^{k} \overline{t_j}$ is the sum of ranks in the $subgroup_i$ complement.

Finally, Geng L. et al. present two *Objective Measures* that do not take into account the distribution of the target variable, but rather the rules and the type of data within the subgroups candidates [32].

Peculiarity

Peculiarity claims that a pattern is possibly more interesting if it covers outliers in the data. Peculiar patterns may be unknown to the user, therefore, interesting.

Conciseness

Conciseness takes into account the number of selectors in the rule. The overall result must contain a limited amount of patterns in order to be concise.

2.1.4 Subjective Measures

According to Ken McGarry, data mining metrics can be either *Objective* or *Subjective* [33]. *Objective Measures* can be computed only using the raw data. *Subject Measures* like novelty, usefulness, and surprisingness, involve knowledge from sources beyond the dataset like the user/expert knowledge. These measures are subjective because they are user-driven and depend on the belief and analysis of the user, being more complex to calculate compared to the *Objective Measures*.

Geng L. et al introduced two interesting *Subjective Measures* in [32]:

Surprisingness

A pattern is surprising (or unexpected) if it contradicts a person's existing knowledge or previously discovered findings. Surprising patterns can help find scenarios where previous approaches failed to find patterns and then highlight the data subgroup that needs a more detailed analysis.

Novelty

A pattern is novel if it is not covered by the users' background knowledge and cannot be derived from other patterns. The difference between surprisingness and novelty is that a novel pattern is new and not contradicted by any pattern already known to the user. In contrast, a surprising pattern contradicts the user's previous knowledge or expectations.

Finally, Geng L. et al proposed another group of metric called *Semantics-based Measures* [32]. *Semantics-based Measures* take into account the knowledge extracted from subgroup descriptors and how it can make an impact in the application domain. We consider this group to be subjective as well since it varies according to the study domain.

The following two measure were proposed by Geng L. et al in [32].

Utility

The Utility of a pattern takes into account the contribution to a user's goal. This kind of interestingness is based on user-defined utility functions in addition to the raw data.

Actionability/Applicability

A pattern is actionable (or applicable) in some domains if it enables decision-making about future actions in this domain. Thereby, a pattern should influence future decisions in the application domain.

2.1.5 Data Structures

To iterate over the search space, it is required to store the results, such as selectors chosen and quality score. This subsection presents the most used data structures to support the efficient selection of subgroups.

Simple Tabular Form

This is the simplest data structure and is the default data structure used to store data in relational databases. However, is not the most efficient practice in Subgroup Discovery [20].

Since after evaluating certain subgroup it is always necessary to compare it to the original dataset and this operations can be time consuming.

Bitsets

This vertical data structure have been used in [17, 34]. In this data representation the instances covered by a subgroup are represented by a binary array. So, each bitset corresponds to a selector. To create conjunctive patters, it is needed to perform AND operations in the multiple candidate bitsets. Therefore, when a rule covers all the bits or none of them (bitsets with all values set to 0 ou 1), this denotes that the rule does not provide any relevant information.

FP-tree Structures

This *Divide-and-Conquer* approach is a very efficient data structure in particular to deal with large datasets. As it as mentioned in [20], FP-trees can be employed to the classical subgroup discovery setting. This can be accomplished in a FP-tree by storing the value of the quality function in the tree nodes. FP-trees are used in well-known algorithms such as FP-Growth, COFI-Tree, and CT-PRO [35] but can also be adapted to be used in Subgroup Discovery.

2.1.6 Search/Enumeration Strategies

Finding the most interesting subgroups requires to enumerate all the possible subgroups candidates to further iterate over them to find the best subgroups according to the quality measure selected. This step can be a bottleneck since the time complexity of the search space is exponential with the number of attributes and their domains. Therefore, multiple approaches are found in the literature to search for subgroup candidates. The most used ones are Exhaustive Search, Beam Search, and Genetic Models.

Exhaustive Search

In Exhaustive Search it is ensured that the best subgroup among the candidates is found. This is accomplished by traversing through the complete search space with only safe pruning techniques [20], i.e., pruning only those candidates which are guaranteed not to be within the best subgroups found. In order to perform this Exhaustive Search, multiple techniques can be used to iterate over the search space, such as Depth-First Search (DFS), Breadth-First Search (BFS) and Best-first-search. Notice that there are multiple variants in each one of the three approaches mentioned before, Florian Lemmerich provides a more detailed explanation of each one of them in [20].

Beam Search

Unlike exhaustive search tecniques, Beam Search uses a *Greedy* approach to find good subgroups. Therefore, it does not guarantee to find the global best subgroup, instead it tries to iterate over the most promising to groups. This way, makes it possible to apply Subgroup Discovery in scenarios with huge search spaces that would not me possible with exhaustive search.

Genetic Algorithm

This tecnique is well-known in Evolutionary Computing. Evolutionary computing uses Darwin principles to solve complex problems. The main idea behind this technique is that given a population of individuals where natural selection is made from one generation to the next one, thus increasing the population's fitness [34].

Berlanga and del Jesus proposed an extension of this technique [36] in order to be applied in Subgroup Discovery. This approach uses bit strings to define the subgroup descriptor where each binary bit corresponds to a selector. Then it is considered a conjunction between the selectors set to true by the genetic approach. It is relevant to mention that this bit strings has no relation with the Bitset data structure.

2.1.7 Pruning Strategies

As it was mentioned before, the size of the search space can be a bottleneck regarding time efficiency mainly because applying exhaustive search to large search spaces is not doable. Therefore, in order to improve efficiency in Exhaustive Searches, it is needed to employ pruning strategies. It is relevant to distinguish between *safe pruning* and *heuristic pruning.* Safe Pruning, only excludes parts of the search space that are guaranteed not to impact the solution. On the other hand, heuristic pruning is a more efficient approach but may ignore interesting parts of the search space due to its greediness.

Optimistic Estimate

In order to prune certain areas within the search space, interesting measures and quality functions can be used to establish the optimistic estimate bounds. However, most measures do not follow the anti-monotone property, i.e., if a set is infrequent, then its specialisations will also be infrequent. Thus it is crucial to take into account the metric used to prune the search space since the target distribution of a certain subgroup specialisation may change arbitrarily.

Florian Lemmerich in [20] outlines that popular anti-monotone constrains may include:

- Maximum number of selectors in a subgroup description
- Minimum subgroup size (coverage)
- Minimum number of positive target values

However, certain quality measures can be used to prune the search space, Martin Atzmüller suggests for example using the binomial test in [30]. If the goal is to find subgroups above a certain threshold from a quality function, every subgroup below that threshold be pruned. This threshold does not needs to be fix, it is possible to employ a dynamic approach where the goal is to find the $best - n$ subgroups, in this scenario, the $n - th$ increases every time a new subgroup is added to the best-n. This dynamic approach has the downside of being dependent on the order in which the search space enumeration is done.

Suppression Heuristics

Klösgen proposed in [17] two types of redundancy filters: Logical and Heuristic. These filters can be used to suppress the search space and can be extremely useful for Exhaustive Searches. This approach can also be called *subgroup suppression.* According to [37] the algorithm suppresses subgroups that are worse than, but not too different from another subgroup. To evaluate similarity between subgroups the balancing overlap degree and significance difference are calculated. This technique not only prunes the search space but also avoids redundancy in the solution.

Background Knowledge

As it was mentioned in 2.1.2, it is possible to employ specialist' background knowledge not only in the post-processing step, but also integrating it in the subgroup discovery pipeline since the subgroup's interestingness is heavily influenced by the end user.

According to Martin Atzmüller [30], there are several ways to employ background knowledge in this stage such as applying constraint and ontological knowledge in the approach's configuration to remove data elements from the search space dynamically as well as using attribute exclusion constrains.

2.1.8 Algorithms

In the previous sections, we discussed multiple concepts around the four main Subgroup Discovery components: Dataset, Search Space, Target concept and Selection Criteria. The algorithms proposed in the literature take into account most of the components mentioned above and several of them include also the data structure and pruning strategies used. When describing the following algorithms, the main focus is to perceive how each approach employs the multiple Subgroup Discovery components and understand the scenarios where they were applied.

We decided to split the algorithms into three different groups according to the approach used to enumerate the search space (Exhaustive Search, Beam-Search and Genetic Search) shown in the tables 2.2 and 2.3.

EXPLORA

The Explora approach proposed by Willi Klösgen in 1996 [17] was the pioneer in subgroup discovery. It applies exhaustive searches on the search space with Depth-First Search, Breadth-First Search, and Best-First search with suppression heuristics as pruning strategy. It also uses bitsets as data structures and deals with both binary and numeric targets. Explora used generality and redundancy as selection criteria when ranking the subgroups.

MIDOS

MIDOS was the first subgroup strategy to deal with multi-relational databases. Proposed by Stefan Wrobel [18] in 1997, employs exhaustive search in the search space (Depth-First Search, Breadth-First Search and Best-First search) as in the Explora approach proposed one year before. In order to improve the search performance, it applies optimistic estimate pruning using WRAcc as an interesting measure. There is little to none information as far as the data structure is concerned, but the authors lead us to believe that it is a tree structure.

CN2-SD

Lavrac et al. [22] proposed an extension of the CN2 classical learning rule for classification problems to cope with the challenge of subgroup discovery. It employs a Beam Search with WRAcc to rank subgroups. Since it uses a greedy search over the search space, no pruning techniques are used in this approach. Contrarily to the vanilla version of CN2, in CN2-SD, when a rule is created, the instances covered by it are not deleted.

SD-MAP

SD-MAP [38] is an exhaustive subgroup discovery algorithm that performs a Depth-First search over the subgroup candidates with a minimum support threshold to prune the search space i.e. only constrains the subgroup size and number of positive targets. It also employs the FP-tree data structure. Binomial test and unusualness are used as quality functions to rank the subgroups.

SDIGA

SDIGA [27] is a genetic algorithm approach for subgroup discovery proposed by Del Jesus et al. in 2007. The description of subgroups is represented in DNF (Disjunctive Normal Form). These fuzzy rules ease the use of numerical variables. Confidence and Support are the quality measures used to select the subgroup candidates. Since this is a genetic approach, there is no need for pruning strategies.

SubgroupMiner

SubgroupMiner proposed by [39] implements the binomial test as a quality measure and uses Beam Search to enumerate the subgroup candidates. It is also able to deal with multi-relational datasets and can handle binary, categorical, and numerical targets (for numeric targets, an on-the-fly discretisation is performed).

DSSD

DSSD [29] is a subgroup discovery approach that considers sets of subgroups instead of individual subgroups. Can deal with any target variables such as binary, categorical, numerical, and complex (multiple targets). Employs a Beam Search to enumerate the search space. Regarding criteria selection, DSSD uses a top-k approach with three additional selection strategies (Description-based subgroup selection, Cover-based subgroup selection, and Compression-based beam selection). In [29] these selection strategies are described in greater detail. Finally, no data structure is described.

SD

The SD algorithm [40] employs a Beam Search strategy and deals only with binary targets. However, it is always possible to convert categorical variables into binary variables after defining the target, as it is done in [40]. As quality measures, it employs the metric defined by Gamberger and Lavrac covered in 2.1.3 and unusualness. Simple support pruning is used to prune the search space.

Apriori-SD

Nada Lavrač and Dragan Gamberger defined this approach as an extension of the Apriori-C algorithm [41]. It performs a levelwise exhaustive search with optimistic estimate pruning to improve the search performance and uses the WRAcc metric as quality function to select the subgroup candidates.

RSD

As in the MIDOS approach, RSD [42] extracts subgroups from a relational database. It performs an exhaustive search in the search space with a Depth-First Search, and to improve the search performance, it employs constraint-based pruning with user-specified constraints. Regarding the data structure used, the authors seem to use a tree structure but do not specify if it is a FP-tree or not. To evaluate the subgroup's quality, the metric WRAcc is used.

Cascaded SD or DpSubgroup

DpSubgroup [43], presented by Henrik Grosskreutz, is a subgroup discovery approach that deals with numeric targets. Thus implements mean-based interesting measures to rank the subgroups. TP-tree is the data structure used by this approach, and to enumerate the search space an exhaustive Depth-first Strategy with Optimistic Estimate pruning was employed.

Exceptional Model Mining

Exceptional Model Mining (EMM) [44] is an innovative subgroup discovery framework that allows finding interesting subgroups with multiple targets. This is done by fitting models (both classification and regression) to subgroups targets that may be somehow exceptional. In order to enumerate the subgroup candidates, this approach uses the heuristic method Beam Search. Regarding the selection criteria, it varies according to the model chosen.

SD-MAP*

Martin Atzmueller and Florian Lemmerich presented the SD-Map* [45] algorithm enabling fast subgroup discovery for continuous target concepts. It performs an exhaustive search to cover the search space with a Depth-first Strategy with an Optimistic Estimate to prune the search space. As selection criteria, it implements a top-k approach with mean-based as interesting measure. The FP-tree is the data structure used.

NumBSD

The NumBSD approach [46] is very similar to the SD-MAP* algorithm. Both are used to deal with numeric targets, and both implement a top-k approach with mean-based as interesting measure. However, the data structure used differs once this approach uses a vertical data structure called bitsets (like EXPLORA). In order to enumerate the search space, this implementation employs a Depth-first Search with a look-ahead and optimistic estimate to prune the search space.

MESDIF

The MESDIF is a multi-objective genetic algorithm for inducing fuzzy rules in the Subgroup Discovery Task [36]. To enumerate the multiple subgroup candidates it employs an elitist approach called SPEA2 [47]. Regarding interesting measures it can use support, confidence, and unusualness (WRAcc).

NMEEF-SD

This Non-Dominated Multi-objective genetic algorithm is based on the well-known Non-dominated Sorting Genetic Algorithm II (NSGA-II) model but is oriented toward the subgroup-discovery [48]. Is able to deal deal both binary and categorical targets and employs unusualness, support and confidence as quality measures to rank subgroups.

GAR-SD

GAR-SD a new evolutionary multi-objective algorithm for Subgroup Discovery tasks [49]. This approach can deal with both discrete and continuous variables with the need of a pre processing step. The interesting measures applied are significance, support and confidence but it also considers the number of rules and number of variables.

Subgroup Algorithm	Dataset	Search Space	Pruning Strategy
Exhaustive Search Subgroup Algorithms			
EXPLORA [17]	Single-relational dataset	Exhaustive Search (DFS,BFS, Best-First)	Suppression Heuristics
MIDOS [18]	Multi-relational database	Exhaustive Search (BFS, Best-First)	Optimistic Estimate
SD-MAP [38]	Single-relational dataset	Exhaustive Search (DFS)	Minimum Support Threshold
APRIORI-SD [41]	Single-relational dataset	Exhaustive Search (Levelwise)	Optimistic Estimate
RSD [42]	Multi-relational database	Exhaustive Search (DFS)	Constraint-based Pruning with user-specified constrains
DpSubgroup [43]	Single-relational dataset	Exhaustive Search (DFS)	Optimistic Estimate
SD-MAP* [45]	Single-relational dataset	Exhaustive Search (DFS)	Optimistic Estimate
NumBS [46]	Single-relational dataset	Exhaustive Search (DFS with look-ahead)	Optimistic Estimate
Beam Search Subgroup Algorithms			
CN2-SD [22]	Single-relational dataset	Beam-Search	–
SubgroupMiner [39]	Multi-relational dataset	Beam-Search	–
DSSD [29]	Single-relational dataset	Beam-Search	–
SD [40]	Single-relational dataset	Beam-Search	Simple support pruning
EMM [44]	Single-relational dataset	Beam-Search	–
Genetic Search Subgroup Algorithms			
SDIGA [27]	Single-relational dataset	Genetic	No pruning
MESDIF [36]	Single-relational dataset	Multi-objective genetic algorithm	No pruning
NMEEF-SD [48]	Single-relational dataset	Non-Dominated Multi-objective genetic algorithm	No pruning
GAR-SD [49]	Single-relational dataset	Multi-objective genetic algorithm	No pruning

Table 2.2: Subgroup Discovery Algorithms

Subgroup Algorithm	**Selection Criteria**	**Target concept**	**Data Structure**
Exhaustive Search Subgroup Algorithms			
EXPLORA [17]	Generality, Redundancy, Simplicity	Binary and Numerical	Bitsets
MIDOS [18]	Novelty, Distributional Unusualness	Binary	Tree Structure, does not specify if it a FP-tree or not
SD-MAP [38]	Binomial test, Unusualness	Binary	FP-trees
APRIORI-SD [41]	WRAcc	Binary, Categorical (Transforms categorical in binary)	–
RSD [42]	WRAcc	Binary, Categorical	Tree Structure, does not specify if it a FP-tree or not
DpSubgroup [43]	Mean-based interesting measure	Numerical	FP-trees
SD-MAP* [45]	Top-k approach with Mean-based interesting measure	Numerical	FP-trees
NumBS [46]	Top-k approach with Mean-based interesting measure	Numerical	Bitset
Beam Search Subgroup Algorithms			
CN2-SD [22]	Modified WRAcc	Binary, Categorical	–
SubgroupMiner [39]	Binomial-test	Binary, Categorical and Numerical	–
DSSD [29]	Description-based subgroup selection, Cover-based subgroup selection, Compression-based beam selection	Binary, Categorical, Numerical and Complex	–
SD [40]	Unusualness, Gamberger and Lavrac Approach	Binary (converted from Categorical)	–
EMM [44]	Depends on the classes of models chosen	Complex/ Multiple targets	–
Genetic Search Subgroup Algorithms			
SDIGA [27]	Confidence, Support	Binary, Categorical	–
MESDIF [36]	Support, Confidence, Unusualness	Binary, Categorical	–
NMEEF-SD [48]	Unusualness, Support, Confidence	Binary, Categorical	–
GAR-SD [49]	Significance, Support, Confidence	Binary, Categorical	–

Table 2.3: Subgroup Discovery Algorithms

2.1.9 Environments/Frameworks

To apply the algorithms and quality measures mentioned previously in 2.1.8, multiple open-source environments allow an easy and yet effective usage of those approaches.

pysubgroup

Florian Lemmerich and Martin Becker implemented a Python package for subgroup discovery called pysubgroup [50]. This implementation allows the user to choose from a wide range of quality functions such as WRAcc and Chi-squared. Can handle binary, categorical, and numeric targets and has visualisation tools integrated to ease the analysis of the subgroups discovered.

VIKAMINE

The VIKAMINE system comes with a variety of established and state-of-the-art algorithms for automatic subgroup discovery [51]. Can deal with binary, categorical and numeric targets due to its large variety of interesting measures. Finally, it also includes an interface with visualisation tools.

rsubgroup

Released in 2021 by Martin Atzmueller, rsubgroup [52] is one of the newest packages to deal with the task of subgroup discovery. This R package contains a collection of efficient and effective tools and algorithms for subgroup discovery and analytics. It is able to deal with both numerical and binary targets and incorporates algorithms such as SD-MAP and BSD. It also integrates an R interface to the VIKAMINE system.

SDEFSR2

SDEFSR2 is a R package with all evolutionary fuzzy systems for subgroup discovery presented throughout the literature [53]. It includes three algorithms: SDIGA [27], MESDIF and NMEEF-SD. Can read data with multiple formats (ARFF, KEEL, CSV and data.frame). Integrates a wide variety of quality functions to choose from. However, the target variable must be categorical. It also includes an interface with visualisation tools.

Cortana

Developed by Marvin Meeng and Arno Knobbe, the Cortana framework simplifies and unifies the procedure of enriching a list, or ranking, of biological entities with background knowledge [54]. Deals with multiple data types, including nominal, numeric, ordinal and binary. The user is able to define everything in terms of search conditions and strategy in its user interface. The scope of this framework is the bioinformatics branch.

2.1.10 Applications

Subgroup Discovery has been applied in real-world situations in various fields. The Subgroup Discovery task is not dependent on a certain context. However, as it was mentioned previously, the validation of an expert in the given domain is crucial in the Subgroup Discovery pipeline. This subsection will present different applications of Subgroup Discovery found in the literature.

Medical Domain

Medical data is always a considerable target concerning data mining applications, and subgroup discovery was not an exception.

In [24] subgroup discovery was used to detect patient groups with a high risk of atherosclerotic heart disease where the target population consists of true positive cases of the disease and non-target class examples corresponding to the healthy population.

Another case shown in [23] uses cases from the SONOCONSULT system (a medical documentation and consultation system for sonography) that consists of detailed descriptions of examinations together with the diagnosis (binary attributes). They performed subgroup discovery only using essential attributes and general background knowledge.

Marketing

Subgroup Discovery has also been used in multiple real-life problems concerning marketing. The first-ever subgroup discovery approach EXPLORA used the data from financial data.

In [55] two marketing studies were performed with the help of Subgroup Discovery, the first was Decision support in a direct mailing campaign, and the second was Decision support in a public advertising campaign. The key question in these studies was how to help marketing analysts decide which client groups are the most suitable regarding the expected results.

In [27] the SDIGA algorithm was applied to the area of marketing concerning the planning of trade fairs in order to find relationship between the trade fair planning variables and the success of the stand.

Spatio-temporal Data Analysis

Subgroup discovery was also used in Station-temporal Data Analysis. Here we will present two articles found in the literature that we consider relevant for this thesis.

"Want to play with me?" by [16] had the objective of finding interesting behaviour from data collected from children in the school, where kids would wear sensors to track their movement in the school recess. Using subgroup discovery, they were able to find interesting patterns in children according to their gender, social skill, age, and others.

In [56] Subgroup Discovery is applied to tactics from spatio-temporal data in soccer to compare successful and unsuccessful attacks from two different teams.

2.2 Spatio-Temporal analysis in Soccer

In this section, a brief overview of the techniques used to analyse soccer data and how to extract relevant information from it is presented. According to Fédération Internationale de Football Association (FIFA) recommendations, the field dimension should be 105 meters in length and 68 meters in width. Another characteristic that hinders the analysis of soccer games is the number of players in-game. A soccer game started with 22 players, which is twice as much compared to other sports like Basketball or Futsal.

2.2.1 Data Format

Concerning the data format, it can be divided into three types that differ both in availability and granularity [57] as shown in Figure 2.1.

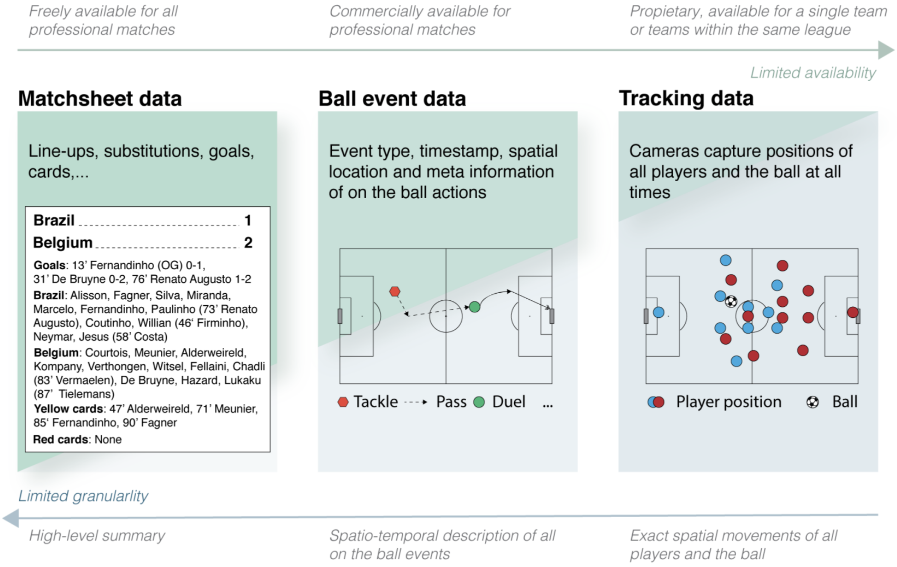

Freely available for all professional matches
Commercially available for professional matches
Propietary, available for a single team or teams within the same league
Limited availability
Matchsheet data
Line-ups, substitutions, goals, cards,...
Brazil 1
Belgium 2
Goals: 13' Fernandinho (OG) 0-1, 31' De Bruyne 0-2, 76' Renato Augusto 1-2
Brazil: Alisson, Fagner, Silva, Miranda, Marcelo, Fernandinho, Paulinho (73' Renato Augusto), Coutinho, Willian (46' Firminho), Neymar, Jesus (58' Costa)
Belgium: Courtois, Meunier, Alderweireld, Kompany, Verthongen, Witsel, Fellaini, Chadli (83' Vermaelen), De Bruyne, Hazard, Lukaku (87' Tielemans)
Yellow cards: 47' Alderweireld, 71' Meunier, 85' Fernandinho, 90' Fagner
Red cards: None
Ball event data
Event type, timestamp, spatial location and meta information of on the ball actions
Tackle
Pass
Duel
...
Tracking data
Cameras capture positions of all players and the ball at all times
Player position
Ball
Limited granularlity
High-level summary
Spatio-temporal description of all on the ball events
Exact spatial movements of all players and the ball

Figure 2.1: Match Sheet Data, Event Stream Data and Tracking Data [58]

- **Match Sheet Data** - Match Sheet Data is the most available type of data once it only gathers all the statistics concerning the soccer match, such as the number of passes per team, cards, shots, goals, line-ups, and substitutions. It is also the data type with less granularity once it only provides high-level summaries about the match. This data is freely available for all professional matches;
- **Tracking Data** - Unlike Match Sheet Data, Tacking Data gives the exact movements of all players and the ball, which is the highest level of detail possible. With tracking data, it is possible not only to analyse the player's formation on the field during the whole game but also to analyse the impact in the game of the players that do not have the ball. However, this data is the most difficult to obtain, since it is only available for a single team or teams within the same league;
- **Event Stream Data** - Event Stream Data is a balance between the previous two data types concerning availability and granularity. It is not as high-level as the Match Sheet Data and, at the same time, does not have the level of detail of Tracking Data. Event Stream Data has Tracking Data only for the player with the ball point of view, performing a certain action such as passing, dribbling, and shooting. Each

event contains a large amount of information, including the Spatio-Temporal data from the action.

To get a complete analysis of a sequence of events, it would be useful to have both the Tracking and Event Stream Data. Thus it would be possible to consider the pressure made by the defending team and analyse possible passing lines to teammates. Using only the Event Stream Data has the constrain that it ignores everything except the on-ball actions.

Soccer Player Action Description Language

One problem found when collecting data from multiple sources is that each companies has its own format to present the data. To bypass this problem, Tom Decroos proposed Soccer Player Action Description Language (SPADL) [59], a new language to describe individual players actions (Event Stream data) from multiple vendors. SPADL brings a unified strategy to load data from multiple sources (Opta, Wyscout, and StatsBomb) in the same format. Each action has a total of nine attributes:

- **StartTime** - Start time of the action in seconds;
- **EndTime** - End time of the action in seconds;
- **StartLoc** - Position where the action started;
- **EndLoc** - Position where the action ended;
- **ActionType** - Type of action performed (e.g., shot, pass), there are a total of twenty one different action types;
- **BodyPart** - Body part used in the action (e.g., foot, head);
- **Player** - Name of the player who performed the action;
- **Team** - Player's team name;
- **Resul** - Result of the action, (e.g., success, fail).

There is also an extension of this framework called Atomic-SPADL, this extension brings more detail information to the vanilla version of SPADL, e.g., if a pass is not successful. With Atomic-SPADL it is possible to understand if the cause was the player performing the pass ou the player receiving it.

2.2.2 Machine Learning and Data Mining Techniques

This subsection presents multiple approaches found in the literature with the purpose of extracting knowledge from soccer data. These techniques can be supervised or unsupervised and their aims can range from evaluating players, teams, players and actions.

Clustering

Clustering in an unsupervised machine learning approach that groups elements from a certain population with similar characteristics. Since in most scenarios there is no ground truth in soccer data, using clustering approaches can help gather information from unlabelled data such as similar plays. S. Hirano et. al. presented a method [60] with the purpose of grouping pass patterns with a clustering technique. This way they were able to find interesting pattern e.g., side-attack after complex pass transactions and zig-zag pass transactions.

Deep Reinforcement Learning

G. Liu et. al. proposed a new approach to evaluate all kinds of soccer actions from event stream data with Deep Reinforcement Learning model [15]. Their neural architecture approach fits continuous game context signals and sequential features within a play with one LSTM (Long Short Term Memory) tower to each team separately. This approach presents two new metric to evaluate actions called Goal Impact Metric (GIM) and Q-value-above-average-replacement (QAAR) that take into account the Q-function from the Deep Reinforcement Learning model.

Binary Probabilistic Classification

A Binary Probabilistic classifier is able to predict a binary output (1 if it is a positive example and 0 if it is a negative example) from a given input. In most approaches the models provide weights to the given features from the input to estimate the most likely output according the input values and its weights. The most common approaches in this field are logistic regression, decision trees and neural networks. This can be used in the context of soccer analytics to predict if a given play (a set of actions) can result in a goal or not.

Neural Network Emsemble Methods

T. Mendes-Neves et. al. presented three ensemble algorithms (Bagging, Simple Average Dropout Networks and Negative Correlation Learning) in [61] with the purpose of predicting the result of soccer games. It was concluded that using ensemble approaches proves to be a reliable way to reduce variance in the context of neural network. However, these approaches might not be optimal in computation time, therefore, its usage is still may depend on the situation being studied.

Non-Negative Matrix Factorization

Non-Negative Matrix Factorization is used to extract significant features from a set of non-negative data vectors. Its advantages compared to other factorisation techniques like Principal Components Analysis (PCA) is that its components are often more human-interpretable. It can be used in soccer analysis to decompose heatmaps of players like it has been done in [62] with the purpose of identifying the player based on his playing style in the previous season.

Mixture Models

Mixture models calculate the probability of a given x thought multiple probabilistic models fitted to the data. Mixture models can also be seen as a soft clustering variant of k-means according to [62]. The most commonly used distribution in mixture models is the Gaussian distribution, but it is possible to used any kind of distribution. This models are implemented in soccer data in order to model the location and direction of action as it has been done in [63].

2.2.3 Soccer Approaches

There are multiple approaches found in the literature concerning extracting valuable knowledge from soccer data. Different approaches are distinguished mainly by the types of data used and the purpose of the study (target). The three main data format considered are Match Sheet Data, Tracking Data and Event Stream Data, mentioned in 2.2.1.

Approach	Target	Data Type
ELO Rating System [64]	Rating Teams	Match Sheet Data
Pi-rating [65]	Rating Teams	Match Sheet Data
Sciskill index [66]	Rating Players	Match Sheet Data
Plus-minus rating [67]	Rating Players	Match Sheet Data
Player Rank Framework [68]	Rating Players	Event Stream Data
EA Sports Player Performance Indicator [69]	Rating Players	Event Stream Data
Data Envelopment Analysis [70]	Rating Players	Event Stream Data
POGBA [71]	Predict Highlights	Event Stream Data
Soccer Mix [63]	Rating Players & Teams	Event Stream Data
STARSS [72]	Rating Actions and Plays	Event Stream Data
VAEP [73]	Rating Actions	Event Stream Data
Expected Goals (xG) [62]	Predict Goals	Event Stream Data
GIM [15]	Rating Players	Event Stream Data
OBSO [74]	Predict Goals	Tracking Data
Expected Goals (xG) [62]	Predict Goals	Tracking Data

Table 2.4: Soccer Analysis approaches found in the literature

In Table 2.4 we present multiple approaches to extract knowledge from soccer data. As we can see, most approaches use event-stream data, probably because the type of data is mostly available, and at the same time, it also provides a good level of detail.

Approaches on Match Sheet Data

Regarding Match Sheet Data, to rate teams, approaches such as, the ELO Rating System and the Pi-rating can be used. The ELO Rating Systems is a well-known strategy to rank chess players and in [64] Hvattun and Arntzen present how it can be used to predict game results. The Pi-rating approach, according to [65] outperforms the ELO Rating strategy and can be used in other more sophisticated models. Both approaches aim to compare teams to further predict game results.

To rate players using Match Sheet Data, Sciskill [66] can be used. This approach only uses the following features to rate players: Line-up (including position), Substitutions, Type of match (e.g. league, cup, international), Competition strength, Goals scored and Minutes played. Other way to rate players is using the Plus-minus rating system, this metric is well-known in other sports such as basketball, however, it as only been introduced to soccer in 2015 in [67], this metric compares the team performance with and without a certain player.

Approaches on Event Stream Data

We outline the following methods for extracting knowledge from Event Stream Data.

Player Rank Framework presented by Luca Pappalardo et. al. in [68] is a algorithm for performance evaluation based on the action performed by the players and it consists of three steps (Rating Phase, Raking Phase and Learning Phase).

The EA Sports Player Performance Indicator [69] is a novel metric that attempts to rate all players using a single score. It is based on the player contribution to winning performances, regardless of the playing position. The final index metric is calculated by a weighted sum of multiple points achieved in the multiple sub-indexes. This metric was used in the top two tiers of soccer in England (the Premier League and the Championship) as an official metric to rank players.

Also in the context of ranking players with Event Stream Data, A. Charnes et. al. proposed Data Development Analysis in [70]. This metric is used to compute players' efficiency from their playing time, goals, assists, tackle win ratio, and pass completion ratio as it is described in [62].

Expected Goals (Expected Goals (xG)), is a very well-known metric in sports in general to measure the probability of a given game state leading to a goal. This metric uses Binary Probabilistic Classifiers and can applied to Event Stream Data.

Also in Event Stream Soccer Data, we highlight the multiple contributions from Tom Decroos, such as Spatio-Temporal Action Rating System for Soccer (STARSS), POGBA approach, Valuing Actions by Estimating Probabilities (VAEP) and Soccer Mix.

The STARSS presented in [72] aims to automatically rate the actions performed by soccer players by leveraging the outcome of a certain play based on past action with similar spatio-temporal characteristics. This approach splits the matches into phases (set of plays) based on a time window defined by the user or whenever the ball possession changes. Then proceeds to identify similar phases using Dynamic Time Warping to further rate them.

The POGBA (Prediction of Goals by Assessing Phases) approach [71] is an algorithm to predict soccer highlights. Given an event at a certain time, with a time window defined by the user, this algorithm predicts the probability of occurring an highlight/critical event within the time window defined. This approach considers the full spatio-temporal data and performs the indirect probability estimation.

In [73], Tom Decroos et. al. present a framework for valuing actions called VAEP, this framework estimates the probability of a given action leading to a goal. However, this approach enable the conversion of a set of action values to a player rating based on both the offensive and defensive contribution to the team. It is relevant to mention that this approach extracts extra information from the event stream data, such as game context features (number of goals scored by both teams at the time when the action occurred) and complex features (distance and angle to the goal in the beginning and the end of the action).

The Soccer Mix approach [63], also by Tom Decroos, aims to capture the playing style of both players and teams using multiple probability models. This approach aims at finding groups of similar actions of a certain type, locations and direction to further use the probabilistic models to encode each action as a probability distribution in a weighted vector. Then, it is possible to build a style vector for a certain player or team by summing the weighted vector of all actions performed by that player or team.

Finally, Guiliang Liu et. al. proposed in their deep reinforcement learning approach [15] a new metric to evaluate players, the Goal Impact Metric (GIM) which ranks a player by aggregating the impacts of all his actions. The impact of an action is the change of consecutive Q values due to this action, based on the learned Q-function from the deep reinforcement learning approach).

Approaches on Tracking Data

Regarding Tracking Data we highlight two metrics to predict goals, OBSO and xG. In [74], it is presented a probabilistic model to compute the probability of a player that does not have the ball in his possession to score a goal, this is called Off-ball Scoring Opportunity (OBSO). The xG metric can be also applied to Tracking Data since it can provide more detail compared to Event Stream Soccer Data, such as open passing lines and pressure by the defences, and those features may add reliability to this metric.

2.3 Related Work

In the previous section, we enumerated several techniques to extract knowledge from soccer data in multiple formats (Match Sheet Data, Tracking Data, and Event Stream Data). Therefore the main focus of this section is to analyse only the Subgroup Discovery approaches in the domain of Soccer. For each work, a detailed analysis has been performed attending to a set of criteria such as the type of data in use, the subgroup group quality measures, and the search strategy. Finally, we present the results provided by each paper.

Google Scholar was the main search engine used for the research of these academic articles, where keywords such as "Subgroup Discovery","Soccer" and "Football" were used. We concluded that applying subgroup discovery in soccer is a fairly recent area since we did not use any filter in the article's dates, and every single one range from 2015 to 2019. The various works studied are presented in Table 2.5.

Meerhoff, L.A. et al. proposed in [56] a data-driven approach that uses Subgroup Discovery strategies to describe tactics from multiple game events aggregations according to their spatio-temporal properties, with varying time windows of interest (5, 10, 15, 20, 25 and 30 seconds).

José Carlos Coutinho et al. proposed in [75] a framework called UnFOOT that measures the performance of both player and teams from position data. That framework includes pre-processing, feature extraction and visualisation tools. Finally, it also employs subgroup discovery to look for frequent distributions in the data.

Meerhoff, L.A. et al. proposed in [76] an automatized methodological approach to identify the key events from the spatio-temporal soccer data to further build explainable spatial relations between the players and define the success of a set of events. To employ Subgroup Discovery to find out explainable and actionable patterns that might be useful to the team's coach.

Jan Van Haaren et al. proposed in [77] an approach that aims at discovering patterns in attacking strategies in professional soccer matches, taking into account interactions among players (passes) in the multiple areas of the field. To do this, they discretised the continuous spatial dimension in the multiple field areas.

Lars Tijssen proposed in [78] an approach, using Subgroup Discovery, to identify if the Link's dangerousity has any impact on the outcome of an attack.

Concerning the type of data, four out of the five articles ([56, 75, 76, 78]) use Tracking data, to know the position of the players throughout the game. In [56] the tracking data is from 100 Dutch premier league games, in [75] due to privacy issues, they do not specify the teams of the league in which is played, they only say the data is from 6 games. In [76] it is used tracking data from 48 Dutch premier league games and in [78] the tracking data is from from 31 matches of the Dutch national soccer team. Finally, [77] uses Event Stream data from a Belgian team during 70 soccer matches.

Regarding the environment used, the papers [56, 77] do not specify any environment, the [75] used pysubgroup and [76, 78] used Cortana.

Also on the subject of subgroup discovery, the Quality Function WRAcc was employed in [76, 77, 78] and in [56, 75] the quality measure to select the subgroup candidates is not specified. Concerning the Search Strategy to enumerate the search space, the Beam Search approach was used in [76, 78] and in the [56, 75, 77], the Search Strategy is not specified. Which leads us to conclude that regarding the Quality Function and the Search Strategy, if we only consider the articles that provided us that information, only the Quality Function WRAcc and the Search Strategy Beam Search were used.

In [56], during the experimentation, they compared successful with unsuccessful attacks between the two Dutch teams. An attack was considered successful when a turnover (lost of ball possession) occurred in the inside the opponents penalty box. They concluded that if the team's length was large, the success rate of attacks would increase. Also, one of the teams had more successful attacks with a low number of passes in a time window of 25 seconds, and the other team was more successful at defending attacks with a high number of passes. Still, there are no information concerning the quality function used and the search strategy.

In [75], they found two interesting subgroups, the first one indicates that players playing in attacking positions tend to have higher agility score, the other states that player having low or high stamina (non-intermediate) are likely to a have a high speed score. Also, besides subgroup discovery, the UnFOOT tool also includes pre-processing and visualisation tools to deal with soccer data.

In [76], with the data provided, 72 features were created using distance-based metrics, Potential Danger indicators and temporal aggregations (with windows of 5 seconds). Finnaly, they found 24 significant (p-value lower than 0.05) subgroups were found with an ROC AUC of 0.627.

In [77], during experimentation, they split the game into phases (set of actions), and considered positive examples phases that ended in a shot. They ended up up 3803 phases having 13.8% of positive examples. Finally, they were able to find interesting patterns within the soccer data, both on passes between certain players, but also passes along certain areas in the field.

In [78], they only considered the attacking plays, where the attack could have three outcomes: no shot, shot off target, shot on target, goal scored. The employing subgroup discovery, they converted the multiple targets into binary targets, where each one of them had different thresholds for the subgroup candidates. They applied subgroup discovery with 1 and 2 descriptors (according to them, more than two descriptors are too difficult to interpret). Subgroups with two descriptors showed better results according to the WRAcc and ROC AUC metric.

Article's Title	Data Type	Environment	Quality function	Search Strategy
Mining Soccer Data: Subgroup discovery of tactics from spatio-temporal data [56]	Tracking data from 100 Dutch premier league games	-	-	-
UnFOOT: Unsupervised Football Analytics Tool [75]	Tracking data from 6 soccer games	pysubgroup	-	-
Exploring Successful Team Tactics in Soccer Tracking Data [76]	Tracking data from 48 Dutch premier league games	Cortana	Wracc	Beam Search
Automatically Discovering Offensive Patterns in Soccer Match Data [77]	Event-stream data from a Belgian team during 70 soccer matches	-	Wracc	-
Analyzing Offensive Player and Team Performance in Soccer Using Position Data [78]	Tracking data from 31 matches of the Dutch national soccer team	Cortana	Wracc	Beam Search

Table 2.5: Summary of reviewed works

Chapter 3

Preliminary Work

In this chapter, we present the preliminary work done concerning the implementation of soccer analysis approaches, visualisation tools and feature extraction to further employ Subgroup Discovery on the pre-processed data. The Figure 3.1 presents the workflow of this preliminary work.

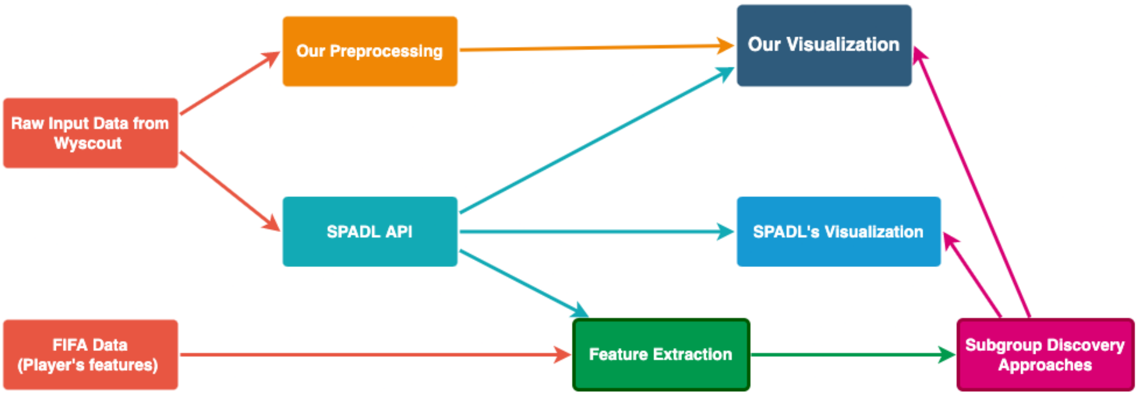

Raw Input Data from Wyscout
Our Preprocessing
SPADL API
FIFA Data (Player's features)
Our Visualization
SPADL's Visualization
Feature Extraction
Subgroup Discovery Approaches

Figure 3.1: Workflow

At the beginning of the work, we started to analyse the raw data from the Wyscout vendor [11], more specifically the English Premier League from the 2018/2019 season. Each event have the following format:

```
event = {
    'eventId': 8,
    'subEventName': 'Simple pass',
    'tags': [{'id': 1801}],
    'playerId': 9637,
    'positions': [{'y': 50, 'x': 50}, {'y': 45, 'x': 40}],
    'matchId': 2500089,
    'eventName': 'Pass',
    'teamId': 1659,
    'matchPeriod': '1H',
    'eventSec': 2.7635970000000043,
    'subEventId': 85,
    'id': 251700146
}
```

However, we changed our approach after coming across the Soccer Player Action Description Language (SPADL) API from Tom Decroos [59]. This API already pre-processed the data shown above and merged the event's information with the dataset from events, teams and players (this way, there is no need to deal with IDs) and it already includes some helpful visualisation tools.

3.1 Exploratory Data Analysis

First started our analysis by understanding the distribution of the multiple events. Each game has an average of 1284.2 events, the maximum number of events in a game is 1654 and the minimum is 926. In total we have 481 574 events from 375 Premier League games.

By analysing Figure 3.2, it is clear that the passing event is by far the most common one. Moreover, we proceed to analyse the success rate of the pass, dribble, and shot events presented in the Table 3.1.

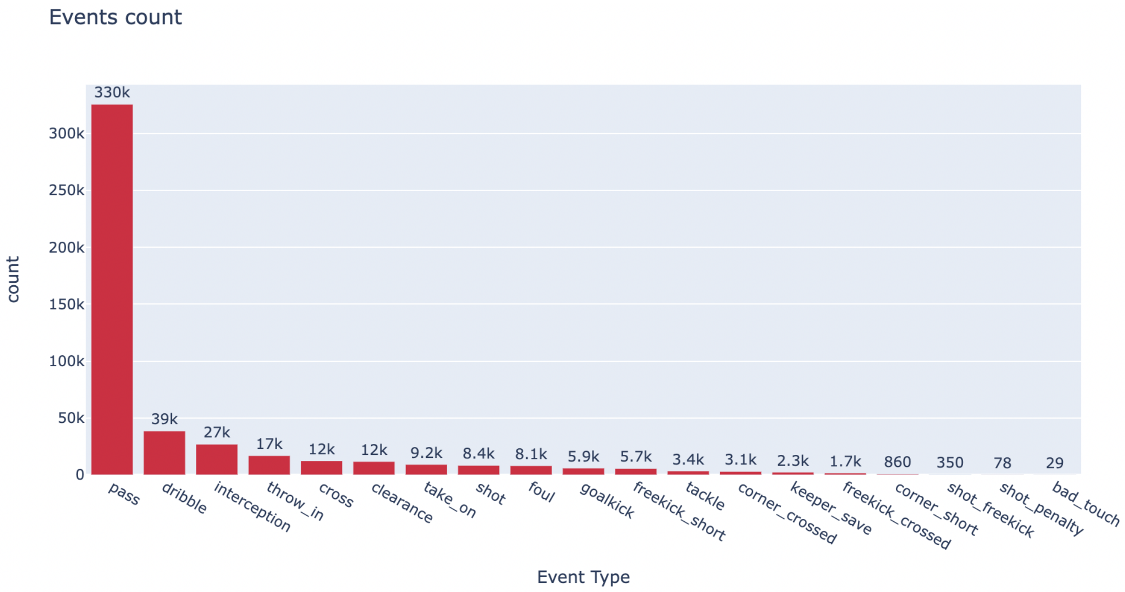


Figure 3.2: Events Distribution

Action	Success Outcome Percentage
Pass	83.0%
Dribble	99.2%
Shot	10.8%

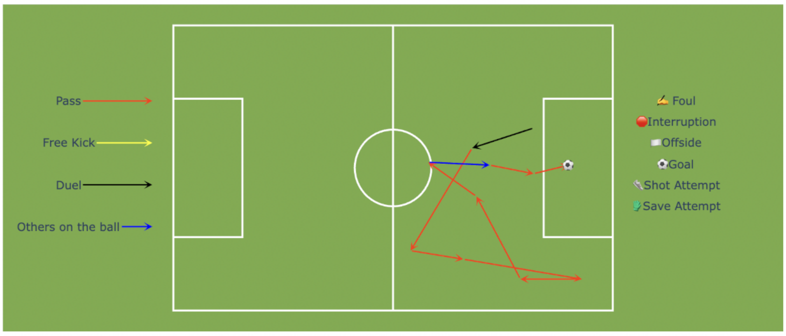
Table 3.1: Success Rate of passing, dribbling and shooting

It is relevant to mention that the dataset's size is **8406** and the number of positives (shots ending in a goal) in the whole dataset is **908**, i.e. **10.8%** of the shots resulted in a goal.

3.2 Spatial Analysis

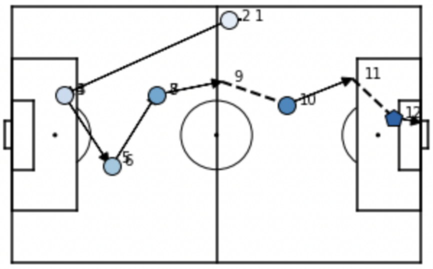
To analyse the event-stream data provived by the Italian sports analytics company Wyscout, we started by the development of a visualisation tool from scratch with the Python library Ploty [79] to analyse consecutive events. Our first visual analysis is presented in Figure 3.3. However, when we change our approach from dealing with the raw data from Wyscout to the SPADL API from Tom Decroos, we also came across the socceraction package [80]. Socceraction is a Python package that uses the SPADL action type conversion with visualisation tools (shown in Figure 3.4), integrates the Valuing Actions by Estimating Probabilities (VAEP) framework mentioned in 2.2.3 and the xT framework to value ball-progressing actions using a possession-based Markov model.

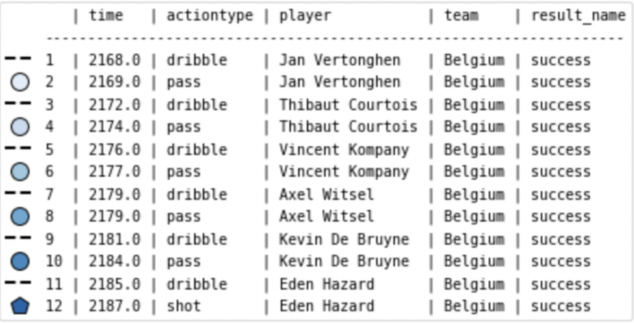
We decided to use the socceraction's visualisation tool to analyse plays, since we consider their approach more clear and intuitive.



```
[HOME] 0 - Time: 24:53, Event: Duel, SubEvent: Air duel and Tags: Won & Accurate
[HOME] 1 - Time: 24:56, Event: Pass, SubEvent: Simple pass and Tags: Accurate
[HOME] 2 - Time: 25:03, Event: Pass, SubEvent: Simple pass and Tags: Accurate
[HOME] 3 - Time: 25:08, Event: Pass, SubEvent: Simple pass and Tags: Accurate
[HOME] 4 - Time: 25:12, Event: Pass, SubEvent: Simple pass and Tags: Accurate
[HOME] 5 - Time: 25:15, Event: Pass, SubEvent: Simple pass and Tags: Accurate
[HOME] 6 - Time: 25:16, Event: Pass, SubEvent: Simple pass and Tags: Accurate
[HOME] 7 - Time: 25:17, Event: Others on the ball, SubEvent: Touch and Tags:
[HOME] 8 - Time: 25:20, Event: Pass, SubEvent: Simple pass and Tags: Accurate
[HOME] 9 - Time: 25:22, Event: Pass, SubEvent: Simple pass and Tags: Assist & Accurate
[HOME] 10 - Time: 25:22, Event: Shot, SubEvent: Shot and Tags: Goal & Left foot & Opportunity
```

Figure 3.3: Visual Analysis





```
       | time   | actiontype | player          | team    | result_name
   --------------------------------------------------------------------
--  1  | 2168.0 | dribble    | Jan Vertonghen  | Belgium | success
○   2  | 2169.0 | pass       | Jan Vertonghen  | Belgium | success
--  3  | 2172.0 | dribble    | Thibaut Courtois | Belgium | success
○   4  | 2174.0 | pass       | Thibaut Courtois | Belgium | success
--  5  | 2176.0 | dribble    | Vincent Kompany | Belgium | success
●   6  | 2177.0 | pass       | Vincent Kompany | Belgium | success
--  7  | 2179.0 | dribble    | Axel Witsel     | Belgium | success
●   8  | 2179.0 | pass       | Axel Witsel     | Belgium | success
--  9  | 2181.0 | dribble    | Kevin De Bruyne | Belgium | success
●  10  | 2184.0 | pass       | Kevin De Bruyne | Belgium | success
-- 11  | 2185.0 | dribble    | Eden Hazard     | Belgium | success
⬟  12  | 2187.0 | shot       | Eden Hazard     | Belgium | success
```

Figure 3.4: Visual Analysis by SoccerAction package

3.3 Feature Engineering

Event Stream Data contains both discrete and continuous data in each event. Regarding discrete data, the event has information about the type of action, which can be one of those shown in Figure 3.2 and it also contains the outcome of the event, which can be success, failure, among others. Finally, it also contains the name of the player who performed the action and the team he plays for. Regarding continuous information, it contains information regarding the game-time when the event occurred (in seconds) and the position on the field where the action started and ended (2 dimensions).

However, using the data in its raw format can compromise the data mining algorithms' performance, so we decided to apply Abstraction Knowledge mentioned in section 2.1.2 on the data and extract spatial features, temporal features and spatio-temporal features. Besides the event data we also decided to add Fédération Internationale de Football Association (FIFA) data (regarding players' rating) to put some weight on the type of player that did a given action.

Finally, we also extracted three extra features, namely the distance from the shot to the goal, the angle from the shot to the goal (in degrees) and the count of unique players that participated in the play.

3.3.1 Spatial Data

When extracting the following features we took into account only the spatial data of the events, i.e., the beginning and end position of each events. It is relevant to note that the distances in question are an approximation, since the events only indicate the beginning and end of the action, they do not take into account the trajectories of the plays. Thus, we calculated the Euclidean distance between the beginning and the end position of each action. The events have the coordinates x and y, where x varies between 0 and 105 meters and y varies between 0 and 68 meters.

Before calculating these features a pre-processing step was done so that all the plays are played from the left to the right of the field.

- *play_distance* (m) - Distance covered in the play (sum of the distance travelled in each action);
- *play_distance_towards_goal* (m) - Distance covered in the play towards the adversary goal, to calculate this feature we only took into account the x variable from the events. This feature can range from -105 to 105;
- *play_mean_distance_to_the_goal* (m) - Mean distance of each action to the adversary goal, this feature is calculated by dividing the play_distance_towards_goal with the number of actions in the play. With this feature it is possible to understand the mean distance progression toward the opponent's goal;
- *play_std_distance_to_the_goal* (m) - Standard Deviation from the distance of each action to the adversary goal;
- *ratio_distance* (m/m) - Ratio between the features play_distance and play_distance _towards_goal, with this feature is possible to analyse direct attacks. If every action is towards the opponent's goal in a straight horizontal line, the ratio_distance is equal to 1. This feature can range between -1 and 1;

- *attacking_action_count* - Number of actions towards the adversary's goal. It is considered that the action was towards the goal of the opposing team if the difference between the variable x at the beginning of the action and at the end of the action is negative;
- *defending_action_count* - Number of actions towards the team's goal. It is considered that the action was towards the goal of the defending team if the difference between the variable x at the beginning of the action and at the end of the action is positive.

3.3.2 Temporal Data

When extracting these features we took into account only the temporal characteristics of the events.

- *total_time (s)* - Time elapsed during the play. It is calculated by subtracting the timestamp when the first action of the play started to the timestamp when the shot occurred;
- *total_time_per_play (s)* - Mean time by action in the play. This feature is calculated by diving the total_time feature by the number of action in the play.

3.3.3 Spatio-temporal Data

In order to calculate these spatio-temporal features, we used the time and spatial features calculated above.

- *play_speed (m/s)* - Ratio between play_distance and total_time;
- *play_speed_towards_goal (m/s)* - Ratio between play_distance_towards_goal and total_time.

3.3.4 FIFA Data

The Fédération Internationale de Football Association (FIFA) is the highest governing entity of association football. Therefore, every year they release data concerning the evaluation of all players worldwide playing in high level leagues. We decided to use the open-source data provided by the FIFA relatively to the year of 2018 (since it is the year of our data) in order to measure abilities from the players such as overall score, passing score and shooting score.

We also decided to normalise this feature with the z-score technique:

$$x_{normalized} = \frac{x - \mu_{feature}}{\sigma_{feature}}$$

From the FIFA data, so far we are just using two features, the average overall score from all the players who took part in the play and the shooting score from the player performing the final shooting action in each play.

3.4 Subgroup Discovery

During this first stage, we tested three different approaches (three different datasets), all of them with Beam Search with a depth of 10 as search strategy and Weighted Relative Accuracy (WRAcc) as quality measure.

Concerning the implementation of Subgroup Discovery, we used the Python Package *pysubgroup* [50] since it provided a lot of freedom when it came to choose the quality function and search strategy.

Concerning the target, we considered the shot resulting in a goal or not, thus our target in binary.

In the first two approaches we considered a window of 20 consecutive events ending in a shot. This set of events includes events from both the attacking and defending team. In the first approach we ignored the temporal component since we only did a count of the events, where all the features are labelled with the event type name and the team that executed it (all features are numerical) as shown in Table 3.2. In the second approach we labelled the features from *event_1* to *event_20* and each instance have the type of event performed and the team that executed it (all features are categorical) as shown in Table 3.3. Finally, in both approaches we included the features related to the distance from the shot to the goal and the angle from the shot to the goal. In the third and final approach of this preliminary stage, we decided to employ all the features mentioned in section 3.3 to the sets of actions that started when the attacking team gain possession of the ball and ended in a shot at the opponent's goal (which can be either a goal or not) as shown in Table 3.4.

Pass [Attacking Team]	**...**	**Shot [Attacking Team]**	**Goal**
3	...	1	1
...	...	...	...
0	...	2	0

Table 3.2: First Approach Dataset

Event 1	**...**	**Event 20**	**Goal**
Pass [Attacking Team]	...	Shot [Attacking Team]	0
...	...	...	...
Dribble [Attacking Team]	...	Shot [Attacking Team]	1

Table 3.3: Second Approach Dataset

play_distance	**...**	**ratio_distance**	**Goal**
30.7	...	-0.3	0
...	...	...	...
19.1	...	0.47	1

Table 3.4: Third Approach Dataset

3.5 Preliminary Results

In the first two approaches, we did not consider the subgroups found to have any relevant information since the subgroup found were as *defendingTeam_offsides = 0* and *event_16 = pass*. The exception was the subgroup *distance_to_goal < 9.63*, which reveals that shots made from close distance to the goal have a higher probability of resulting in a goal.

However, in the third approach, with the features shown in the section 3.3, we considered to have found some interesting subgroups. The subgroups are shown in Table 3.5, where we show the value provided by the quality function WRAcc to rank the subgroups. Also the descriptors which enable us to interpret the subgroups space, the size of the subgroup, the number of positive instances in the subgroup and finally the percentage of target share in the subgroup, which must not be lower than 10.8%.

Quality	**Subgroup Descriptors**	**Subgroup's Size**	**Positives in Subgroup**	**Subgroup Target Share**
0.014920	play_mean_distance_to _the_goal (m)<21.38	1681.0	307.0	0.182629
0.013778	defending_play: [0:1[AND play_mean_distance_to _the_goal (m) <21.38	1233.0	249.0	0.201946
0.010863	ratio_distance (m/m)>=0.74	1682.0	273.0	0.162307
0.010489	defending_play: [0:1[AND ratio_distance (m/m) >=0.74	1563.0	257.0	0.164427
0.010246	Shooting Score>=1.83	1767.0	277.0	0.156763

Table 3.5: Subgroup Discovery Results

To analyze the subgroups found, we decided to visualize the plays pertaining to each subgroup. To do that, we analyzed the position where the shot was made (Figure 3.5) and the trajectory of the play from the moment the team gained the possession until executing the shot (Figure 3.6).

Here we decided to analyse only the *play_mean_distance_to_the_goal (m)<9.31* and *ratio_distance (m/m)>=0.74* subgroups since they are the only subgroups that are entirely unrelated to each other. In spite of that, these two subgroups present a similar shot position, since most shots are made inside the penalty area.

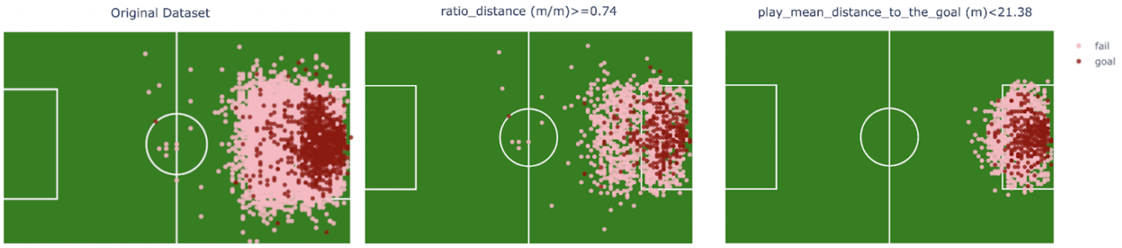


Figure 3.5: Shot Location Comparison

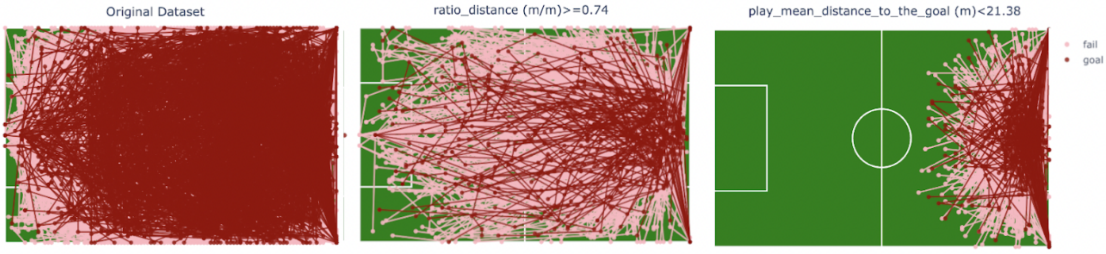


Figure 3.6: Play Trajectory Comparison

Finally, we decided to analyse how the search strategy and the quality function impact the subgroups discovered. To test this hypothesis, we used the features from the third approach mentioned in section 3.3.

To determine the search strategy's impact on the subgroups discovered, we fixed the WRAcc as quality measure and experimented with the following search strategies: Beam Search, Depth-First-Search with Optimistic Estimate as Pruning Strategy and Best-First-Search with Optimistic Estimate as Pruning Strategy.

After analysing the results from each search strategy we concluded that all of them provided the same top-5 subgroups, however, the elapsed time for each approach to discover the subgroups is rather different. While the Depth-First-Search and Best-First-Search approaches took 7.55 seconds and 4.32 seconds respectively, the Beam Search approach took 68.2 milliseconds. This experiment leaded us to conclude that the Beam Search is the best approach to deal with this dataset, since it discovered the same subgroups as the other approaches in a shorter execution time.

To verify the quality function's impact on the subgroups discovered, we decided to fix the Beam Search as search strategy and test the following quality functions: WRAcc, Chi-Squared, Coverage and Binomial test.

Unlike what happened when testing with the multiple search strategies, the top-5 subgroups found with the multiple quality function are all distinct from each other, which makes sense since the function that evaluates and ranks the subgroups is different. Regarding the execution time, all approaches took about the same time (average of 101 milliseconds) to find the best 5 subgroups.

These experiments lead us to conclude that with our dataset, the quality functions have an huge impact on the subgroups discovered while the search strategies only impacts the execution time.

Chapter 4

Experimental Setup

In this chapter, a description of the experimental setup is provided. First, we describe the data used in Section 4.1 followed by its preprocessing stage in Section 4.2. Then, we explain the features extracted from the Event-Stream and Tracking Data in Section 4.4. The analysis regarding both the plays and the features are presented in Sections 4.3 and 4.5. Finally, we go through the experiments done concerning Expected Goals and Subgroup Discovery (Sections 4.6 and 4.7).

4.1 Dataset

The data was provided by two companies, Opta [81] and Tracab [13]. Opta provided the event-stream data, and Tracab provided the metadata and the tracking data. All this data was gathered by SciSports [82]. The metadata file contains information about the players in the game, such as name, team, and frame when they entered and left the game. The event file has all the events (such as pass, dribble and shot) that happened in the game with also the spatial (start and end position where the event occurred) and temporal (event's timestamp) information from the event. Finally, the tracking data file has the position (x and y) of every player and referees on the pitch with a frequency of 25 Hz. To extract the information, we are using four files, the F7 and F24 files from Opta and the TracabMetadata and TrababDat from Tracab. Both F7 and TracabMetadata files contain the metadata information about the game, while the F24 and TrababDat contain the event and tracking data from the game, respectively. In this work, we are using the data from Eredivisie, the Dutch Premier League, from the 2020/2021 and 2021/2022 seasons, which gives us a total of 503 games (265 from the 2020/2021 season and 238 from the 2021/2022 season).

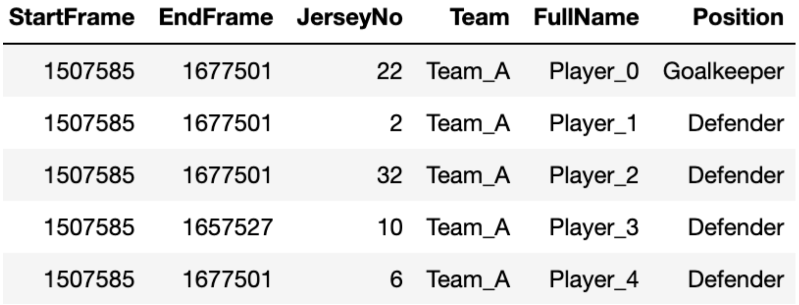
4.2 Preprocessing

After gathering the data, a conversion process was conduced since both the metadata and event data were in .xml format, while the tracking data was in .dat format. In Table 4.1 we present the major transformations performed on the data.

	X-Axis Range	Y-Axis Range	Playing Direction
Tracking Data	-5500 to 5500	-3400 to 3400	Both Direction
Event Stream Data	0 to 100	0 to 100	Left to Right
Preprocessed Data	0 to 105	0 to 68	Left to Right

Table 4.1: Data Preprocessing

From the TracabMetadata file, we extracted the following features: PlayerID, First Name, Last Name, Full Name, StartFrame, EndFrame, and Jersey No. We used the F7 file from Opta only to get the player's positions (Goal Keepers, Defender, Midfielder, and Striker). Then, we merged these two to get the following table containing all the metadata information we considered relevant to our study, as it is shown in Figure 4.1.



StartFrame	EndFrame	JerseyNo	Team	FullName	Position
1507585	1677501	22	Team_A	Player_0	Goalkeeper
1507585	1677501	2	Team_A	Player_1	Defender
1507585	1677501	32	Team_A	Player_2	Defender
1507585	1657527	10	Team_A	Player_3	Defender
1507585	1677501	6	Team_A	Player_4	Defender

Figure 4.1: Metadata preprocessed

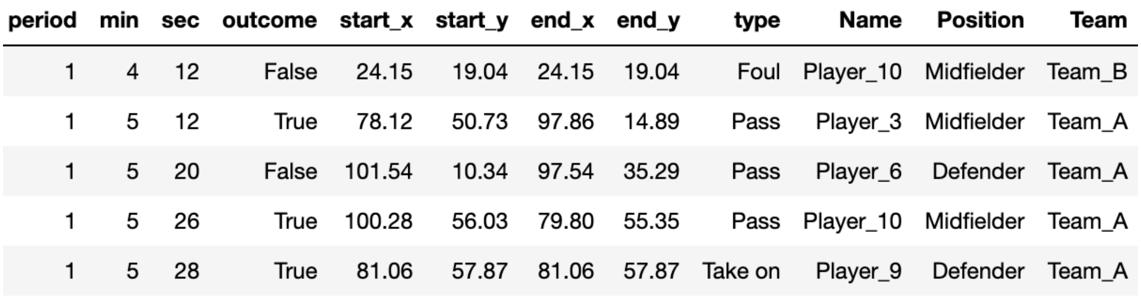
It is important to mention that the variables StartFrame, and EndFrame refer to the moment when a certain player entered and left the game, respectively. Also, we extracted the moment when the first and second halves started and ended from the metadata files.

Concerning the event's file (F24), we converted it from the .xml format into the same format as the metadata, shown in Figure 4.2. Opta provided a wide range of 79 different event types. However, after analyzing the event types, we noticed that not all of them are relevant to our study. Thus we ended up only using 24 events out of the 79. The events within a soccer game are aperiodic. Therefore, each event contains the timestamp from the moment that event occurred (minutes and seconds).

For each event, we also have the start and end locations from the event. Those tracking locations from the events range from 0 to 100 both for the x-axis and y-axis. Also, all plays are played from left to right according to the team which has the ball possession. So, to match the tracking data, we needed to change the scale to 0 to 105 meters on the x-axis and 0 to 68 meters on the y-axis.

Besides the Spatio-temporal information, each event also has the player's and team's identifier from who performed the event. So, we merged it with the metadata table to get the player's and team's names and also the player's position.

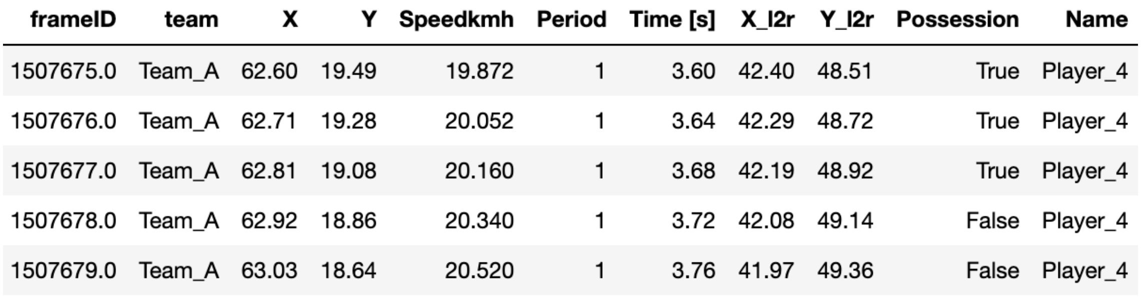
Finally, regarding the tracking data, the TracabDat file provided the players' and ball's position for every frame in the game with a frame rate of 25Hz, this way, we know their exact position every 0.04 seconds. Since we considered having all the player's tracking information in the same file to be unclear, we decided to create a table for each player and the ball. The tracking position data was provided in centimeters, and the values ranged from -5500 to 5500 on the x-axis and from -3400 to 3400 on the y-axis. Therefore, we rearranged it to be in meters and range from 0 to 105 on the x-axis and from 0 to 68 on the y-axis. Thus, the positioning information from both event data from Opta and



period	min	sec	outcome	start_x	start_y	end_x	end_y	type	Name	Position	Team
1	4	12	False	24.15	19.04	24.15	19.04	Foul	Player_10	Midfielder	Team_B
1	5	12	True	78.12	50.73	97.86	14.89	Pass	Player_3	Midfielder	Team_A
1	5	20	False	101.54	10.34	97.54	35.29	Pass	Player_6	Defender	Team_A
1	5	26	True	100.28	56.03	79.80	55.35	Pass	Player_10	Midfielder	Team_A
1	5	28	True	81.06	57.87	81.06	57.87	Take on	Player_9	Defender	Team_A

Figure 4.2: Event-Stream Data preprocessed

the tracking data from Tracab would match. This file also provides the speed from the multiple entities within the pitch in meters per second, the moments when the ball is dead or alive, and finally, which team has the ball possession in each frame. Also, since the event's positioning data is always playing left to right, we found it relevant to detect when each team was playing left to right and right to left. To detect that, we analyzed the goalkeepers' average position for each team and each half. This way, we know which side the teams are attacking. So, we added the X and Y position as if they were playing left to right. The final result is shown in Figure 4.3.



frameID	team	X	Y	Speedkmh	Period	Time [s]	X_l2r	Y_l2r	Possession	Name
1507675.0	Team_A	62.60	19.49	19.872	1	3.60	42.40	48.51	True	Player_4
1507676.0	Team_A	62.71	19.28	20.052	1	3.64	42.29	48.72	True	Player_4
1507677.0	Team_A	62.81	19.08	20.160	1	3.68	42.19	48.92	True	Player_4
1507678.0	Team_A	62.92	18.86	20.340	1	3.72	42.08	49.14	False	Player_4
1507679.0	Team_A	63.03	18.64	20.520	1	3.76	41.97	49.36	False	Player_4

Figure 4.3: Tracking Data preprocessed

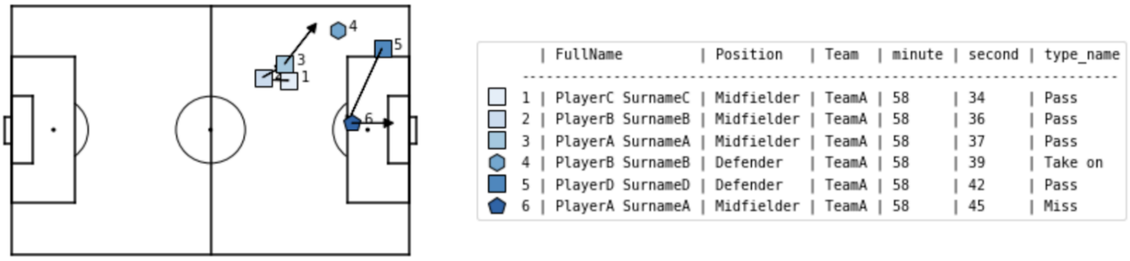
4.3 Analysing Plays

In this work, a play is a sequence of events that starts when a certain team gains ball possession and ends when that same team shots the ball into the opponent's goal. That shot can result in a goal or a miss (which can also be a save from the goalkeeper). Therefore, the length of the plays varies, and within each play, we know that the ball possession is always on the same team, which would not happen if we consider a fixed-size time window for each play.

After gathering every play that ended in a shot for each game, the next step is to merge it with the tracking data. Since the event and tracking data providers are different, we cannot use identifiers to merge the data because they do not match. Thus, the only way to merge the different data sources is to use the timestamp. However, since the sample rate from the tracking file is 25 Hz, the granularity is higher than the event's file because there, we only have limited information up to the seconds. Thus, in scenarios where we want to analyze precise moments in the game, e.g., the moment a certain shot was taken, we calculated the average position of the players and ball in that second.

To validate the analyses, we decided to use and develop visualization tools to see the events and tracking data within a play. Concerning the event data, we used Socceraction [80], a Python package that includes visualizations using event stream data, as we see in Figure 4.4. Regarding the tracking data visualizations, we decided to create our own visualizations. This way, we have more control over our analysis, as well as visualizing the flow of the game in a certain time window and forcing the play to be left to right based on the team that is attacking. Besides the player's positions, we also added their speed and direction, as we can see in Figure 4.5.

From the whole dataset, we were able to extract a total of 11375 plays that ended in a shot. Within these plays, only 1311 resulted in a goal, leading us to conclude that the probability of scoring a goal in our dataset is 11.5%.



```
    | FullName          | Position   | Team  | minute | second | type_name
    ---------------------------------------------------------------------
□   1 | PlayerC SurnameC | Midfielder | TeamA | 58     | 34     | Pass
□   2 | PlayerB SurnameB | Midfielder | TeamA | 58     | 36     | Pass
□   3 | PlayerA SurnameA | Midfielder | TeamA | 58     | 37     | Pass
⬡   4 | PlayerB SurnameB | Defender   | TeamA | 58     | 39     | Take on
■   5 | PlayerD SurnameD | Defender   | TeamA | 58     | 42     | Pass
⬟   6 | PlayerA SurnameA | Midfielder | TeamA | 58     | 45     | Miss
```

Figure 4.4: Event-Stream Data visualization

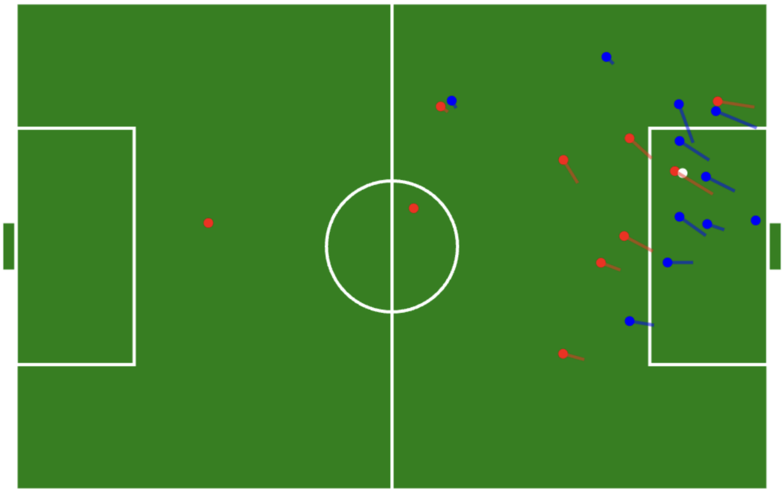


Figure 4.5: Tracking Data visualization

4.4 Feature Extraction

Up to now, the raw data is only being used in the visualization tools because, to extract some knowledge from the plays, we need to extract features from both the event stream and tracking data. It is relevant to mention that all plays were preprocessed in order to be played left to right from the attacking team's perspective and right to left from the defending team's perspective. This way, it is easier to calculate some features such as distance and angle from the shooter to the goal, and when analyzing the plays, there is no need to guess the side that is being attacked. We have extracted 38 features, 11 from the event stream data and 27 from the tracking data presented in Table 4.2. First, we will present the features extracted from the event data and then from the tracking data.

Feature	**Data Source**	**Data type**	**Units**
ShotDistanceFromTheGoal	Event Data	Continous (float)	Metres
ShotAngleFromTheGoal	Event Data	Continous (float)	Degrees
PlayDuration	Event Data	Discrete (int)	Seconds
NumberEvents	Event Data	Discrete (int)	Count
NumberPasses	Event Data	Discrete (int)	Count
NumberDribbles	Event Data	Discrete (int)	Count
HorizontalStartPos	Event Data	Categorical (string)	-
VerticalStartPos	Event Data	Categorical (string)	-
TimeBin15min	Event Data	Discrete (int)	-
TimeBin5min	Event Data	Discrete (int)	-
ExtraTime	Event Data	Binary (bool)	-
StrikerSpeed	Tracking Data	Continous (float)	Km/h
BallSpeed	Tracking Data	Continous (float)	Km/h
BallStrikerSpeedRatio	Tracking Data	Continous (float)	-
DefendersAt3MetersRadius	Tracking Data	Discrete (int)	Count
DefendersAt7MetersRadius	Tracking Data	Discrete (int)	Count
AttackingConvexHullArea	Tracking Data	Continous (float)	M^2
DefendingConvexHullArea	Tracking Data	Continous (float)	M^2
HeightAttacking	Tracking Data	Continous (float)	Metres
WidthDefending	Tracking Data	Continous (float)	Metres
AttackingTeamSpread	Tracking Data	Continous (float)	Metres
DefendingTeamSpread	Tracking Data	Continous (float)	Metres
AttackingCentroidDistanceToGoal	Tracking Data	Continous (float)	Metres
DefendingCentroidDistanceToGoal	Tracking Data	Continous (float)	Metres
AttackingCentroidAngleToGoal	Tracking Data	Continous (float)	Degrees
DefendingCentroidAngleToGoal	Tracking Data	Continous (float)	Degrees
DistanceBetweenCentroids	Tracking Data	Continous (float)	Metres
GoalKeeperDistanceFromTheGoal	Tracking Data	Continous (float)	Metres
GoalKeeperAngleFromTheGoal	Tracking Data	Continous (float)	Degrees
PitchControlAttacking	Tracking Data	Continous (float)	Percentage
MaxPitchControl	Tracking Data	Continous (float)	Percentage
MoreLikelyPitchControl	Tracking Data	Continous (float)	Percentage
ShooterMeanSpeed	Tracking Data	Continous (float)	Km/h
ShooterStdSpeed	Tracking Data	Continous (float)	Km/h
CounterAttack	Tracking Data	Continous (float)	-
BallTravelDistance	Tracking Data	Continous (float)	Metres

Table 4.2: Feature Engineering

4.4.1 Features extracted from Event Stream Data

We started by calculating the spatial features: distance and angle from the shot's position to the goal. Since every attacking team is playing left to right (following the strategy provided by Opta), the goal is in the fixed point (105, 34). Then we calculate the Euclidean distance in meters and the angle in degrees. We also considered the position when the play started. Instead of using the exact position, we decided to split the soccer pitch into vertical and horizontal regions, as it is shown in Figure 4.6.

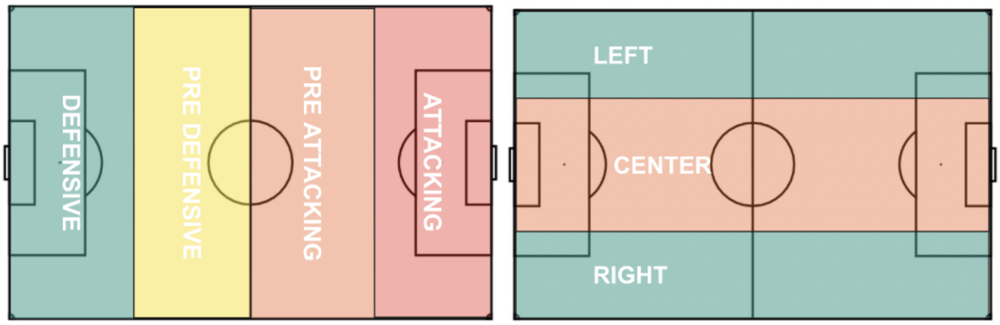


Figure 4.6: Pitch split into vertical and horizontal regions

Then we calculated the following temporal features: timestamp when the event occurred and the duration of the play, both in seconds. Instead of using the precise timestamp when the play happened, we split the game time into 5 and 15 minutes intervals. Lastly, we calculated a binary feature that indicates if the shot was taken in overtime or not.

Finally, we counted the number of events in the play and the number of passes and dribbles within those events.

4.4.2 Features extracted from Tracking Data

Concerning the features extracted from the tracking data, we split them into the following multiple groups: Speed/Distance Features, Defenders Features, Teams Features, GoalKeeper Features and Pitch Control Features.

Speed/Distance Features

The following features only take into account the speed from the ball and from the player who performed the shot during the play:

- *StrikerSpeed* - Speed from the player who executed the shot at the moment when the shot was taken;
- *BallSpeed* - Ball's speed at the moment when the shot was taken;
- *BallStrikerSpeedRatio* - This feature can also be called Control according to [76]. Here, we divided the ball's speed by the striker's speed. If the value is 1, both the striker and the ball were at the same speed when the show was taken. This can be interpreted as the striker having maximum control over the ball;

- *ShooterMeanSpeed* - Average speed from the player who executed the shot along with the play;
- *ShooterStdSpeed* - Speed's standard deviation from the player who executed the shot during the play;
- *BallTravelDistance* - Distance covered by the ball along with the play;
- *CounterAttack* - In this feature, we first calculated the distance between the position where the ball was at the beginning of the play and the end but only on the x-axis ($Xend - Xbegin$), and then we divided it by the *BallTravelDistance* feature. This way, we can understand the ratio between the distance traveled towards the opponent's goal and the whole distance covered during the play by the ball. If the value is, e.g., 0.75, it means that for every 10 meters covered, 7.5 meters were towards the opponent's goal.

Defenders Features

In the following features, we calculated the Euclidean distance between the defenders and the player who performed the shot (shown in Figure 4.7):

- *DefendersAt3MetersRadius* - Count the number of players from the defending team in a radius of 3 meters from the player who performed the shot when the shot was taken;
- *DefendersAt7MetersRadius* - Count the number of players from the defending team in a radius of 7 meters from the player who performed the shot when the shot was taken.

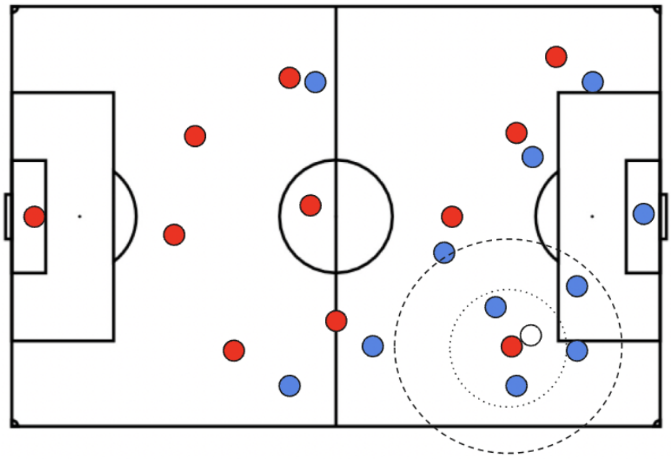


Figure 4.7: Example of a play where we analyze the number of players from the opposing team within a radius of three and seven meters from the player who performed the shot.

Team Features

The following features take the position of all players from both teams into account. Most of these features were inspired by [76]. To better understand these features we show in

Figure 4.8 the Convex Hull from a randomly selected play, in Figure 4.9 the Height and Width from both the attacking and defending team and finally in Figure 4.10 we show the centroids from both teams in the same play.

- *AttackingConvexHullArea* - Convex Hull area around the attacking team (without the goalkeeper);
- *DefendingConvexHullArea* - Convex Hull area around the defending team (without the goalkeeper);
- *HeightAttacking* - Distance between the player who is closer to the opposing team's goal and the player closer to his own goal, only considering the x-axis and without the goalkeeper (from the attacking team's point of view);
- *WidthAttacking* - Distance between the left and rightmost player in the pitch only considering the y-axis (from the attacking team's point of view);
- *HeightDefending* - Distance between the player who is closer to the opposing team's goal and the player closer to his own goal, only considering the x-axis and without the goalkeeper (from the defending team's point of view);
- *WidthDefending* - Distance between the left and rightmost player in the pitch only considering the y-axis (from the defending team's point of view);
- *AttackingTeamSpread* - Standard deviation from the distance between every player from the attacking team (without the goalkeeper) and the team's centroid. This feature is used to evaluate how dispersed the players are from the centroid;
- *DefendingTeamSpread* - Standard deviation from the distance between every player from the defending team (without the goalkeeper) and the team's centroid. This feature is used to evaluate how dispersed the players are from the centroid;
- *AttackingCentroidDistanceToGoal* - Euclidean distance between the attacking team's centroid and the opponent's goal;
- *AttackingCentroidAngleToGoal* - Angle between the attacking team's centroid and the opponent's goal;
- *DefendingCentroidDistanceToGoal* - Euclidean distance between the defending team's centroid and their own goal;
- *DefendingCentroidAngleToGoal* - Angle between the defending team's centroid and their own goal;
- *DistanceBetweenCentroids* - Distance between each team's centroid.

GoalKeeper Features

We only took into account the goalkeeper from the defending team:

- *GoalKeeperDistanceFromTheGoal* - Euclidean distance between the defending team's goalkeeper and defending team's goal;
- *GoalKeeperAngleFromTheGoal* - Angle between the defending team's goalkeeper and defending team's goal.

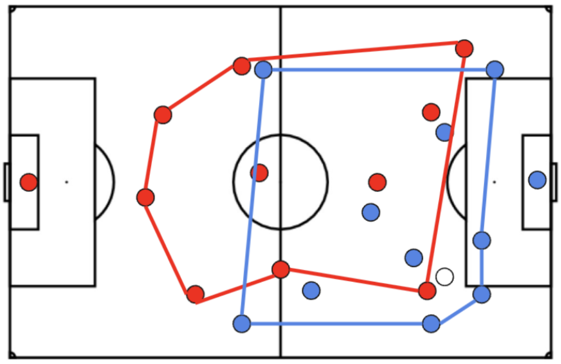


Figure 4.8: Example of play where we analyse the Convex Hull area from both teams (excluding the Goalkeepers).

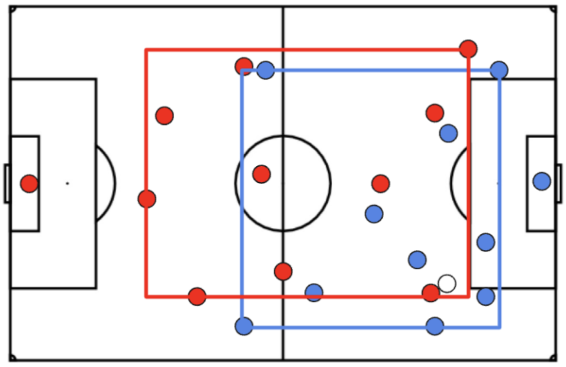


Figure 4.9: Example of play where we analyse the Height and Width from both teams (excluding the Goalkeepers).

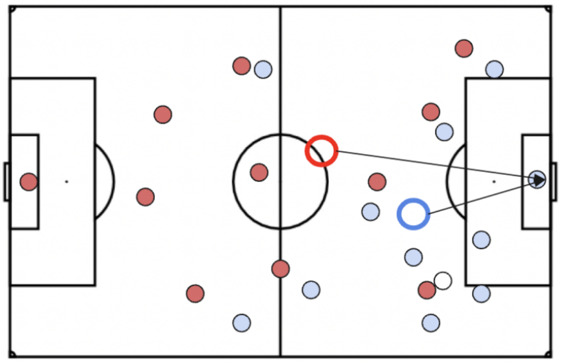


Figure 4.10: Example of play where we analyse the centroid's positions from both teams (excluding the Goalkeepers).

Pitch Control Features

The concept of pitch control was introduced by William Pearsman [83], and it is defined by the probability of a certain player getting the control of the ball, assuming it was in a certain location of the pitch. In order to calculate this model, we need to take into consideration the position, speed, and direction of every player on the pitch and the ball. In our work, we are only taking into account the attacking and pre attacking regions of the pitch (explained in Section 4.4.1) at the moment when the shot was taken.

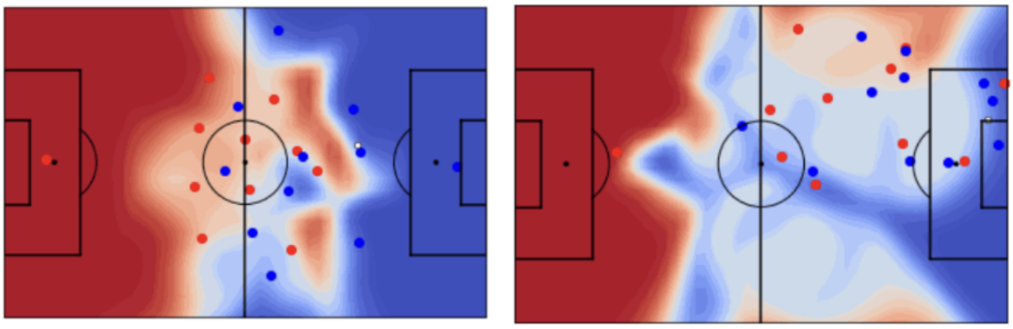


Figure 4.11: Examples of how the pitch control is distributed around the pitch.

We transformed the attacking half of the pitch into a grid of (25x30) squares to further calculate the pitch control in each grid cell. The values range from 0 (maximum pitch control for the defending team) to 1 (maximum pitch control for the attacking team) in each cell. In Figure 4.11 we show two moments when we calculated the pitch control, where the color red in the regions where the attacking team has a higher chance of gaining ball possession. The color blue in the regions where the defending team has a higher chance of gaining ball possession. Finally, the color white in the regions where the probability of both teams gaining ball possession in that region is more or less the same. With this in mind, we extracted the following features:

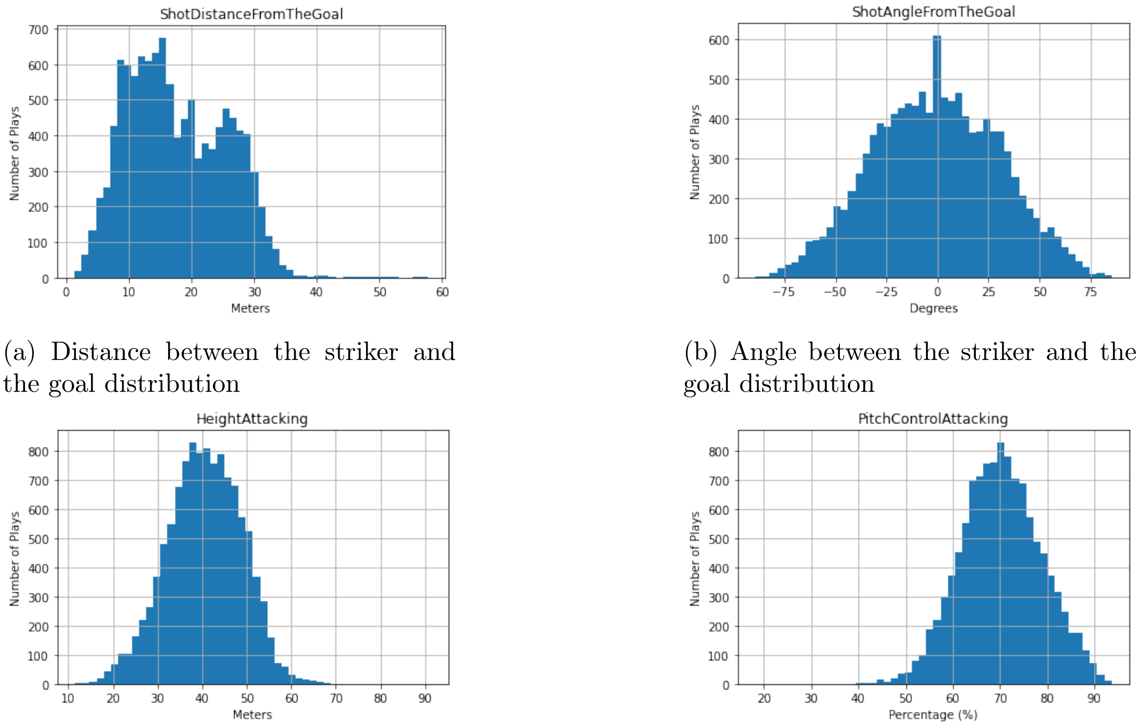
- *PitchControlAttacking* - Average pitch control. It ranges between 0 and 1 since it is an average of probabilities;
- *MaxPitchControl* - Count the number of cells with maximum pitch control for the attacking team;
- *MoreLikelyPitchControl* - Count the number of cells with pitch control above 0.51 for the attacking team.

4.5 Analysing Features

In this section, we analysed the relationship between the target value and the remaining features extracted. First, we analysed the distribution of the some features that we extracted (Figure 4.12). In Figure 4.12a we can notice an interesting pattern in shots at a distance of more or less 26 meters, and in Figure 4.12b we can conclude that there is a peak of shots made at zero degrees from the goal.

Then, we analysed how the features' distribution varies when we consider only plays that end in a goal compared to plays that end in a miss (Figure 4.13). This way, it is possible for us to determine which features have more impact on a play resulting in a goal. For instance, we can observe a notorious difference in the distributions in Figure 4.13a, which leads us to conclude that there are many more goals when the distance of the shot to the goal is shorter. Also, the Figure 4.13d leads us to conclude that there are no goals when the average pitch control is below 37%.

In addition, we also analysed how the values of the feature variables impact the probability of a goal. In Figure 4.14, we can analyse how the goal probability changes according to the values of two distinct features. This analysis lead us to conclude that the goal's probability decrease when the shot distance and angle to the goal (absolute value) increase.



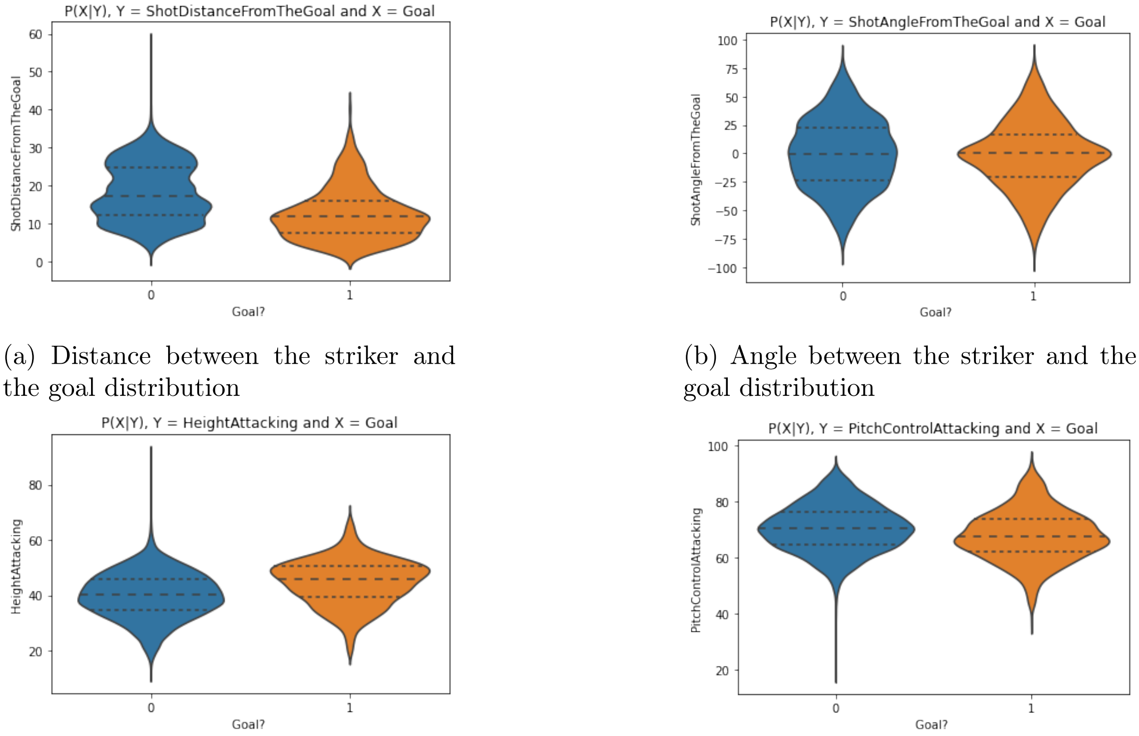
(a) Distance between the striker and the goal distribution

(b) Angle between the striker and the goal distribution

(c) Attacking team height distribution

(d) Average pitch control distribution

Figure 4.12: Example of some features' distribution



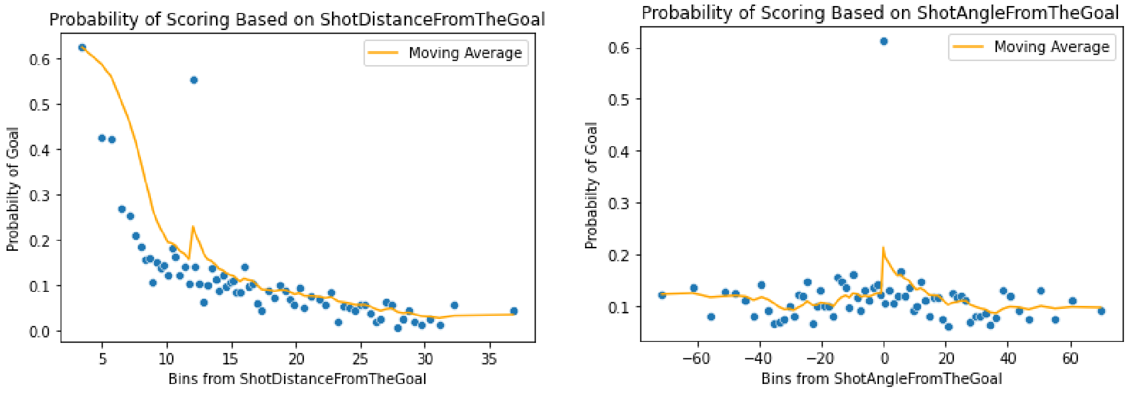
(a) Distance between the striker and the goal distribution

(b) Angle between the striker and the goal distribution

(c) Attacking team height distribution

(d) Average pitch control distribution

Figure 4.13: Example of some features' distribution when a play ends in a goal or a miss



(a) Distance between the striker and the goal probability

(b) Angle between the striker and the goal probability

Figure 4.14: Relation between the features's values and the goal's probability to score

4.6 Expected Goals' Experimentation

During the Expected Goals' experiments, we tested this approach with multiple classification models and multiple sets of features.

With regard to the models, we experimented not only with models we found in the literature (such as XGBoost and LogisticRegression) but also with classification models provided by the package Scikit-Learn [84]:

- XGBoost Classifier
- Logistic Regression Classifier
- Support Vector Machine Classifier
- K-Neighbors Classifier
- Decision Tree Classifier
- Voting Classifier with all the models from above

Concerning the features, instead of only using all 38 features extracted, we decided to experiment with the multiple following sets of features. Firstly, we evaluated the models using only the distance and angle to the goal, since they are the most used features to predict Expected Goals found in the literature. Then, we proceeded to calculate the importance of the features with the support of Logistics Regression [84]. To get the feature's importance from the Logistics Regression model, we started by fitting the whole dataset into the model to further extract the coefficient's values from each feature (shown in Figure 4.3). The idea behind this approach is the larger the coefficient (in both positive and negative values), the greater the impact it has on a prediction. Finally, we experimented using only a few of the most relevant features to train our Expected Goals model:

- Distance and Angle
- Most relevant Feature

- Top 3 relevants Features
- Top 5 relevants Features
- Top 10 relevants Features
- All 38 Features

	Feature	Importance
1	ShotDistanceFromTheGoal	0.126
2	HeightAttacking	0.071
3	StrikerSpeed	0.049
4	WidthAttacking	0.030
5	MoreLikelyPitchControl	0.016
6	DefendersAt7MetersRadius	0.015
7	AttackingCentroidAngleToGoal	0.014
8	DefendingCentroidDistanceToGoal	0.014
9	AttackingCentroidDistanceToGoal	0.014
10	DefendingCentroidAngleToGoal	0.014

Table 4.3: Top 10 most relevant features according to Logistics Regression

Then, we also employed thresholds in the values provided by the Expected Goals (xG) models, i.e., if the value is higher than the threshold (which can be related to a dangerous play or a near-miss) , we consider it to be a goal. We tested with thresholds of 0.2, 0.3, 0.5 and without threshold.

In order to evaluate the results we used multiple metrics. Some of them are well-known metrics used to rank models in the domain of Machine Learning, such as, ROC AUC, R^2 and Brier Score. Then, we also proposed a domain specific metric to evaluate xG models. This metrics aims to evaluate the separation between goals and misses taking into account the xG's average in plays that end in a goal, and subtract it to the xG's average in plays that end in a miss:

$$Diff = AVG(xG_{Goal}) - AVG(xG_{Miss})$$

Finally, we employed a grid search with all the models, features, and thresholds in order to select the best model that fits our data.

4.7 Subgroup Discovery's Experimentation

Our main objective during experiment concerning Subgroup Discovery is to extract relevant knowledge from soccer data with Subgroup Discovery techniques and address which features are used to constrain the subgroups' space. Also, we want to verify if the subgroups change when analyze each team separately. Finally, we will analyze how the subgroups change when we use the binary (Goal/Miss) and numerical (xG) as target and their interestingness within the context of soccer.

For the binary target, we used WRAcc as quality function and Beam Search as search strategy. For the numeric target, we used the Z-Score as quality function and Beam

Search as search strategy as well. Concerning the continuous features, we defined ten bins for each variable.

During all experiments regarding Subgroup Discovery, we set the depth to two, since we consider that increasing the depth of the Subgroup Discovery task will not only increase the time it take to execute but also makes it harder to interpret the outcome.

Finally, we employed Subgroup Discovery's techniques into our data we used two different packages: Pysubgroup [85] and Cortana [86].

Chapter 5

Results

This chapter will present the results of the experiments done on modeling Expected Goals and Subgroup Discovery.

5.1 Expected Goals' Results

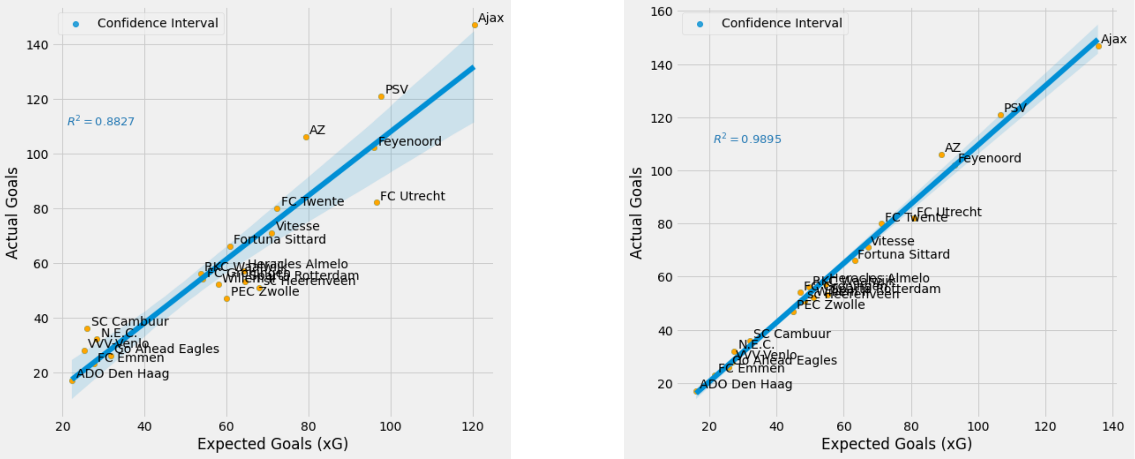
After testing all the models enumerated in Section 4.6, we concluded that most models had the best performances using all 38 features available. In the end, we concluded that the best model that fitted our data is the one produced by the XGBClassifier as it is shown in Table 5.1.

First, we want to highlight the comparison between our best model with the Logistics Regressor since it is one of the most used models when it comes to predicting xG. When comparing Figures 5.1a and 5.1b, with the blue line representing a 1st order equation ($y = \alpha x + \beta$) fitted into the data with a 95% confidence interval, we can see that the R^2 is much higher for the XGBClassifier approach (0.9895) compared to the Logistics Regressor (0.8827). According to [87], we can evaluate the xG's model with the R^2 metric. R^2 measures the percentage of the response variable variation that a linear model explains. Therefore, an R^2 of zero means the model does not explain any of the variations in the response, and an R^2 of one represents a model that perfectly describes the data. In our approach using the XGBClassifier, the R^2 is 0.9895, which leads us to conclude that the model fits the model almost perfectly.

We concluded that approaches that use Decision Trees (DecisionTreeClassifier and XGBClassifier) gave us the best results according to the R^2 metric. However, when comparing the ROC AUC plots and score from both approaches (Figures 5.2a and 5.2b), the XGBClassifier got much better results with an ROC AUC of 0.80.

Concerning the xG Diff score, it takes into account how well separated the goals and misses are according to the xG model. To present the results concerning this metric, we also created visualizations where we can see how the xG's distribution varies when we take the whole dataset into account, only the plays that end in a goal and only the plays that end in a miss (as it is shown in Figure 5.3). The highest xG Diff score found was 0.85 from the XGBClassifier model, using all features with a threshold of 0.2.

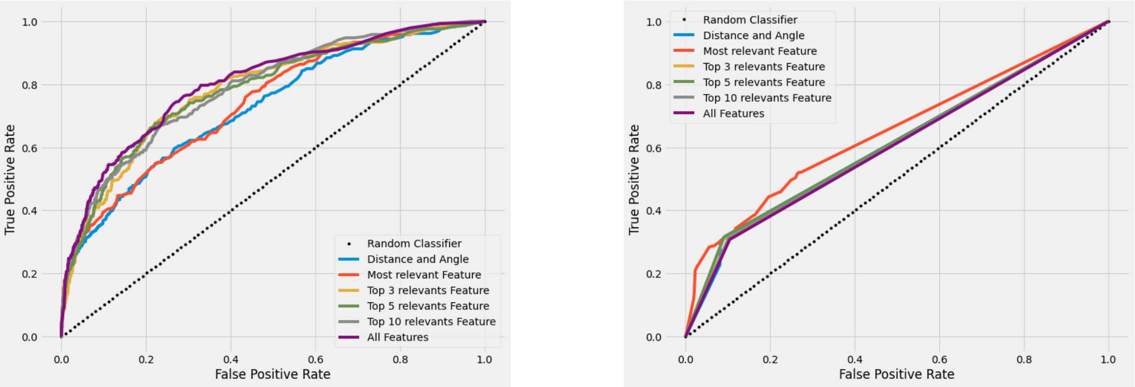
We also evaluated the models according to the Brier Score. The Brier Score was proposed by Glenn W. Brier in [88], and it aims to measure the accuracy of predictions' probability. In our scenario, since it is a unidimensional prediction, this metric is equivalent to the



(a) Logistics Regression

(b) XGBClassifier

Figure 5.1: Comparison between Logistics Regression and XGBClassifier (xG vs Actual Goals)



(a) XGBClassifier ROC AUC

(b) Decision Tree ROC AUC

Figure 5.2: Comparison between XGBClassifier and Decision Tree Classifier (ROC AUC)

mean squared error between the probabilities and the binary target Goal/Miss. Therefore, the lower the Brier Score, the better the predictions. According to the Brier Score, the best model is the XGBClassifier, using all 38 features and a threshold of 0.5, resulting in a score of 0.02.

However, if we took the thresholds into account, the results become slightly better. The XGBClassifier is still the model which provides the best results according to all the metrics, yet the best threshold changes from one metric to another. According to the ROC AUC metric, the highest area under the curve is reached when we do not use any threshold, but according to Brier Score, we get the best score when we use a threshold of 0.5. Thus, we can not conclude anything about the threshold. It relies on the metrics we use to evaluate the models, as it is shown in Table 5.1.

During the previous experiments, we used 80% of the dataset to train the model, and then we used the whole dataset to test it. However, our main goal is to overfit the model into the data. In order to do that, we proceeded to train and test the best model (XGBClassifier) with the whole dataset. This resulted in a model with a ROC AUC of 1.0 and an R^2 of 1.0. This led us to conclude that the model overfitted the data perfectly and is ready to be used in the Subgroup Discovery task. For a more detailed analysis upon the xG results

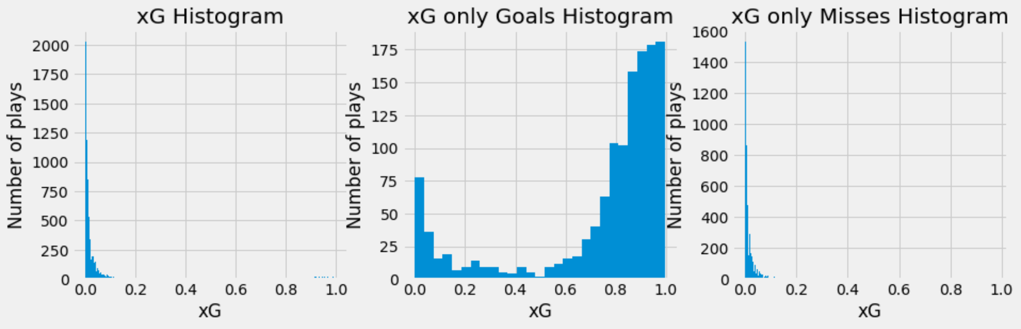


Figure 5.3: XGBClassifier xG's distributions

Metric	Best model	Best features' set	Best Threshold	Metric's Value
ROC AUC	XGBClassifier	All Features	1.0	0.80
R^2	XGBClassifier	All Features	0.3	0.99
Brier Score	XGBClassifier	All Features	0.5	0.02
xG Diff Score	XGBClassifier	All Features	0.2	0.85

Table 5.1: Best result per metric

check the appendix A.

5.1.1 Unexpected Goals

After analysing the results from xG, we came across many models that fitted the data reasonably, but they had wrong predictions when it came to goals scored with a low xG, as we can see in Figure 5.3. That led us to think about those "Unexpected Goals".

We concluded that most misses were well predicted, i.e., all missed shots had a really low xG score. However, the problem was mainly in the shots that ended in a goal since there are multiple shots with rather low xG that resulted in a goal. A real example of this scenario is Wayne Rooney's goal on the 29th of November 2017 against West Ham United, where he scored 54 meters away from the opponent's goal.

So, if we subtract the binary target Goal/Miss by the value of xG:

$$Surprisingness = goal? - xG$$

Where *goal*? can be either 0 or 1, and xG can range between 0 and 1. This formula results in a continuous spectrum between -1 and 1, which we call Surprisingness. Values near zero represent shots whose xG is according to reality, misses that have low xG, and goals that have high xG. Values close to 1 represent the Unexpected Goals, and values close to -1 represent the Unexpected Misses.

Therefore, since our overfitted model does the separation of Goals and Misses perfectly, it is not possible to analyze Unexpected Goals/Misses. For this reason, we will use the best model from the experiments (XGBoost) but trained with only 80% of the population using all 38 features of the dataset. Its results are in the Figures 5.1b and 5.2a. As we can see in the second histogram in Figure 5.3, the xG values near zero represent our Unexpected Goals, and from the third histogram, since there are almost no values near one, we can conclude that Unexpected Misses are much scarcer.

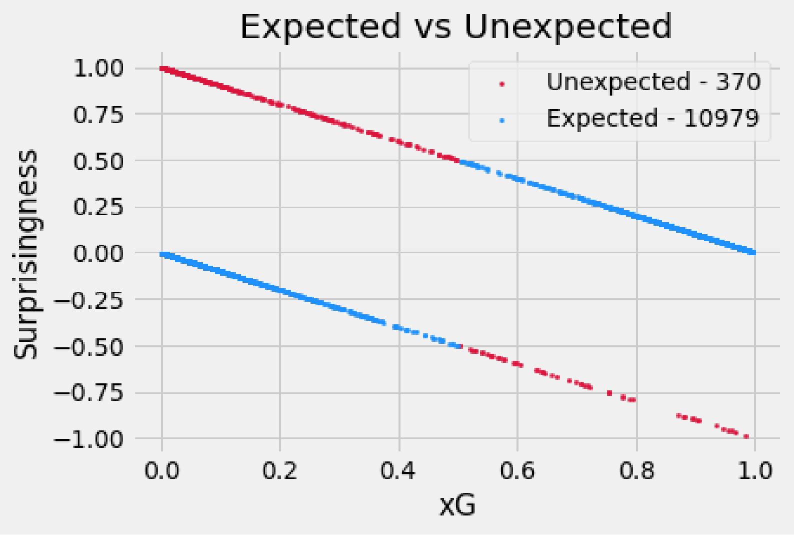


Figure 5.4: Unexpected Goals

In order to facilitate the analysis of these Unexpected Goals/Misses, Figure 5.4 illustrates how the data is distributed. We use a threshold of -0.5 and 0.5 to define the Unexpected shots. That is, values between -0.5 and 0.5 are considered Expected, and values outside that range are considered Unexpected.

Finally, we decided to implement Subgroup Discovery only in the Unexpected Goals (red dots on the left in the Figure 5.4). To do so, we used the CORTANA software with the following parameters:

- Quality Function - Z-score
- Search Strategy - Beam Search
- Depth - 2
- Target average - 0.87
- Population - 316

This experiment aims to analyze what values the features take that increase the surprisingness of unexpected goals. After analyzing the best subgroups found, we concluded that the surprisingness increased when the distance of the shot to the goal increased and when the height of the team decreased, reaching a target average of 0.97 with a coverage of 75 elements (23%).

Then, we also found other subgroups which increased the target average from 0.87 to 0.93 with a coverage of 25%. These subgroups are defined as having the attacking team's height lower than 42 meters and the actions that happened in the first half of the match.

These analyses let us know the dataset's characteristics where the model makes most of the wrong predictions. By the subgroups analysis, we can conclude that long shots (superior to 16 meters), when the height of the team is inferior to 42 meters (for example, in corners or free kicks) and also adding the factor that these shots are in the first half, leads us to conclude that the model has more difficulty in making correct predictions within this search space.

It should be noted that all this analysis was done in the soccer context, but we consider that this can be done in multiple other domains in order to understand what aspects lead a model to make wrong predictions.

5.2 Subgroup Discovery's Results

Concerning the results gathered by the Subgroup Discovery's techniques, we first had to define some preliminary parameters such as the number of bins for the numerical features, the refinement's depth, and the number of subgroups to be extracted:

- Number of bins per numerical variable: 10
- Refinement's Depth: 2
- Number of best subgroups to be extracted: 100

5.2.1 Cortana vs PySubgroup

Firstly, Subgroup Discovery was employed in both Pysubgroup and Cortana packages to verify if the quality and subgroup descriptions changed from one package to another. The whole dataset with 11375 plays was used in both approaches considering only the binary target (Goal/Miss). It is relevant to mention that both approaches are defined with the same parameters mentioned above and with WRAcc and Beam Search as Quality Function and Search Strategy, respectively.

In order to compare both methods, we used the ROC AUC metric from the subgroups and the descriptors from the best subgroups found.

The ROC AUC score was 0.711 and 0.712 for the Pysubgroup and Cortana implementations, respectively, which leads us to conclude that the subgroup's quality is the same for both approaches. Also, the descriptors from the best subgroup are basically the same for both approaches, as it is shown in Table 5.2.

Albeit, we decided to use Cortana's approach for the rest of the experiments since we consider it to be a more robust and better-documented library than Pysubgroup.

Framework	Description	WRAcc Score	Coverage	Target Share
Cortana	StrikerSpeed >= 5.42 AND ShotDistanceFromTheGoal <=16.91	0.035	4546	0.20
PySubgroup	StrikerSpeed >= 5.42 AND ShotDistanceFromTheGoal <=16.50	0.035	4366	0.21

Table 5.2: Comparison between the best subgroup found by each package

5.2.2 Binary vs Numerical target

Then, we employed Subgroup Discovery but with a numerical target (xG) instead of the binary target (Goal/Miss) to verify how the results changed from one approach to another (with Cortana).

After analyzing the results, we noticed that the descriptors from the top-5 best subgroups change from one approach to another. Regarding the experiment with the binary target, all subgroups from the top-5 contain the same features in the descriptors (distance between the striker and the opponent's goal and the striker's speed, both at the moment when the shot was taken). While the subgroups' descriptors from the numerical target approach include the same features in the descriptors as the binary approach plus another feature (number of opponents in a radius of three meters). This leads us to conclude that it is possible to extract more knowledge from the dataset when considering different target types.

We also decided to intercept the subgroups from both approaches to see how many subgroups they have in common and check if the subgroup's rank matched from one approach to another. Five subgroups were found by both approaches, as it is shown in Table 5.3. Besides one of the subgroups, they all match more or less the same rank. It is relevant to mention that we are only evaluating the top-100 subgroups by each approach. If we increase the number of subgroups found, the number of subgroups in common would also increase.

Subgroup's Description	Binary Rank	Numerical Rank
ShotDistanceFromTheGoal <= 12.39 AND DefendersAt7MetersRadius <= 4.0	53	58
ShotDistanceFromTheGoal <= 12.39 AND DefendingCentroidDistanceToGoal <= 11.67	89	94
ShotDistanceFromTheGoal <= 12.39 AND StrikerSpeed <= 4.71	7	86
ShotDistanceFromTheGoal <= 12.39 AND StrikerSpeed <= 6.37	30	37
StrikerSpeed <= 4.71 AND ShotDistanceFromTheGoal <= 12.39	43	43

Table 5.3: Rank comparison between Subgroups

Regarding the results from the best subgroups (from the binary and numerical target) in the Cortana's approach, the goal's probability went from 11.5% to 20.0% with a coverage of 4546 plays and xG's mean when from 0.12 to 0.38 with a coverage of 913 plays.

5.2.3 Whole dataset vs One team dataset

Finally, we analyzed if there are subgroups specifics for some teams. In order to accomplish that, we extracted three subsets from the whole dataset, each one of them containing only plays from a certain team.

Concerning the three teams, which we called TeamA, TeamB, and TeamC (to preserve the anonymity of the data), where each one of them has a different playing style. TeamA is one of the top-table teams, TeamB is one of the middle-table teams, and finally, TeamC is one of the bottom-table teams.

Dataset	Positives/Size	Best Subgroup Description	WRAcc Score	Coverage	Target Share
All Plays	1307/11349 (11.5%)	ShotDistanceFromTheGoal <=16.91 AND StrikerSpeed >= 5.42	0.035	40%	20%
TeamA's Plays	147/1052 (14.0%)	ShotDistanceFromTheGoal <=12.18 AND DefendingCentroidDistanceToGoal >= 10.59	0.050	32%	30%
TeamB's Plays	81/613 (13.1%)	ShotDistanceFromTheGoal <=21.88 AND DefendersAt7MetersRadius <= 2	0.044	29%	29%
TeamC's Plays	47/619 (7.6%)	ShotDistanceFromTheGoal <= 16.19 AND BallSpeed >= 21.99	0.031	32%	17%

Table 5.4: Comparison between the whole dataset and the plays from three different teams

Since our dataset is unbalanced, each subset is also expected to be unbalanced. After analysing the subsets' targets, we concluded that there is a certain positive correlation between the target's balance and the team's performance. This is logical since the target's unbalance is caused due to the lack of goals scored compared to the total attempts, so if a team performs better, it will score more goals resulting in a less unbalanced dataset. This can be verified in our scenario (shown in the second column in Table 5.4) if we take a look at the goal frequency between TeamA (14.0%) and TeamC (7.6%).

Regarding the Subgroup Discovery results presented in Table 5.4, we concluded that there is a positive correlation between the team's performance and the value from the Subgroup Discovery's quality function. The WRAcc score increases when the team's performance increases. The best subgroup for TeamA, TeamB, and TeamC has a WRAcc score of 0.050, 0.044, and 0.031, respectively.

Also, we concluded that the subgroups found reflect each team's playing style. As we can see in Table 5.4, all the teams increase the goal's probability to score when the distance from the shot to the goal decreases (which is obvious), but then each team has a different descriptor that defines its playing style. Concerning TeamA, the goal's probability increases when the opposing team is further away from the goal, according to the feature *DefendingCentroidDistanceToGoal* (e.g. counter-attacks or isolated plays). The goal's probability increases for TeamB if there are less than two players from the opposing team in a radius of seven meters from the striker at the moment when the shot was made. Which leads us to conclude that TeamB scores much more goals when the striker is isolated from the opposing team. Finally, TeamC has a higher chance of scoring if the ball speed is higher than 22.99 km/h when the shot was made (e.g. corners).

Since all subgroups included the feature ShotDistanceToTheGoal, we decided to do the same experiments but without that feature. The results are presented in appendix B.

This page is intentionally left blank.

Chapter 6

Conclusion

Subgroup Discovery has proven to be an effective and robust data mining approach to deal with soccer data. Its explainability is a huge advantage comparing to other Data Mining techniques in terms of actionability, i.e., it eases decision-making about future actions in the domain.

Overall, the best-scored subgroups from the two targets (goal/miss and xG) present the same features in the subgroup's descriptors. However, the granularity of the continuous target variable (versus the binary target) allow us to extract more knowledge from the same dataset. Also, there are subgroups that are specific for certain teams, which leads us to conclude that the team's playing style impacts the subgroups found.

Regarding the Expected Goals' prediction, we consider gradient boosted trees (XGBoost) to be the model that best fitted our data. We also consider that this approximation made by the model enables us to extract more information from a binary target, so within the positive set of the target we can perceive which samples are "more positive" than others.

We conducted two experiments designed to answer the four research questions proposed.

1. Does the use of different search strategies and quality measures lead to subgroups that represent substantially different knowledge?

2. Can we identify interesting subgroups combining information from different sources?

3. Is it possible to use a Subgroup Discovery approach to find knowledge that is specific to a particular team or league

4. Are the subgroups discovered related with the five time phases (defensive and offensive organisations, the defense-attack and attack-defense transitions, and stop balls) in the game or with the time when those occurred (first or second half)?

According to the results obtained from the preliminary results and main experiment, we drawn the following conclusion for each one of the questions:

1. In the preliminary experiments, we analysed the subgroups discovered by four different quality measures in the same dataset, and each quality function gathered different subgroups from each other. This led us to conclude that the subgroups found are highly dependent to the quality function, which make sense since each quality function has its own parameters and weights. Concerning the Search Strategy, we tested

with three different approaches (two exhaustive and a greedy one). They all provided the same subgroups found. The only thing that varied was the execution time required to discover the subgroups candidates. The greedy Search Strategy (Beam Search) took less time to compute the subgroups found;

2. In both experimentations conducted during this study we combined information from different sources. In the preliminary experiment, we combined Wyscout's event-stream data and data from FIFA. In the main experiment, we combined information from Tracab's tracking data and Opta's event-stream data. After analyzing the subgroups found by each experiment, we concluded that it is possible to identify subgroups whose descriptors contain information from the multiple sources;

3. In the experiment performed in Section 5.2.3 we concluded that there are subgroups specific to certain teams since the subgroups varied from one team no another. This concludes that Subgroup Discovery is able to catch different playing styles from different teams.

4. Since most features calculated take into account the time phases mentioned, it is logical that most subgroups found are related to the five-time phases. Also, during the Unexpected Goals, one of the most relevant subgroups found is defined by shots made in the first half, concluding that there are subgroups related to the time when the plays occurred.

6.1 Lessons Learned

In this section we highlight some of the lessons learned during this study.

- **More interesting knowledge can be obtained with Abstraction knowledge** - As we concluded when analysing the preliminary results from the three approaches tested, the approach which provided the best results was the one where we employed Abstraction Knowledge (feature engineering).

- **Visualisation tool supports the interpretation of the subgroups discovered** - After discovering the most interesting subgroups according to the metrics used, we only could understand the subgroup's nature after visualising the plays within the subgroups discovered.

- **Beam Search is a reliable and optimised approach to iterate over the search space** - After comparing the subgroups found with multiple search strategies, we concluded that the Beam Search provided the same results as the exhaustive searches in a significantly shorter execution time.

- **Increasing data granularity improves the results** - After comparing the results from the preliminary work (where we only used event-stream data) and the main experiment (where we used both event-stream and tracking data), we can conclude that this statement is true since the results got much better when we added tracking data into the experiment. In the experiment where we only used event-stream data, the best subgroup reached a WRAcc score of 0.015, while in the experiment where we increased data granularity (by adding tracking data), the best subgroup reached a WRAcc score of 0.035.

- **Soccer experts must evaluate the subgroup discovery results' actionability** - After gathering the results discovered, the specialist on soccer's domain must verify if it is possible to take advantage of the subgroups found and how they may impact the game flow.

- **Increasing the target's granularity may allow to gather even more information** - During the main experiment we converted the binary target into a continuous one by overfitting a machine learning model into the data and then extracted the goal's probability according to the model (xG). Then, we proceeded to employ Subgroup Discovery using both targets separately and we concluded that the approach which had a continuous target was able to find subgroups with a high quality function value that were not found by the approach which used a binary target.

6.2 Future work

Concerning future work, we consider that there is a lot to be done in this domain.

About the features extracted, we consider that there is always more to be done. We were able to extract 38 features in our main experiment, however we did not considered any features that took the player's individual performance into account. We believe that it would be interesting to include data from external sources in order to fill that gap.

Regarding Subgroup Discovery, we consider that it would be interesting the possibility to develop a quality function that is specific to soccer, where we can extract knowledge from its values instead of using the most used quality functions such as WRAcc and T-Score.

This page is intentionally left blank.

References

[1] All you need to know about soccer. https://www.bundesliga.com/en/faq/all-you-need-to-know-about-soccer. Accessed: 2022-01-18.

[2] Pedro Henriques Abreu, José Moura, Daniel Castro Silva, Luís Paulo Reis, and Júlio Garganta. Performance analysis in soccer: a cartesian coordinates based approach using robocup data. *Soft Computing*, 16(1):47–61, 2012.

[3] Fernando Almeida, Pedro Henriques Abreu, Nuno Lau, and Luís Paulo Reis. An automatic approach to extract goal plans from soccer simulated matches. *Soft computing*, 17(5):835–848, 2013.

[4] João Cravo, Fernando Almeida, Pedro Henriques Abreu, Luís Paulo Reis, Nuno Lau, and Luís Mota. Strategy planner: Graphical definition of soccer set-plays. *Data & Knowledge Engineering*, 94:110–131, 2014.

[5] Pedro Henriques Abreu, Daniel Castro Silva, Fernando Almeida, and Joao Mendes-Moreira. Improving a simulated soccer team's performance through a memory-based collaborative filtering approach. *Applied Soft Computing*, 23:180–193, 2014.

[6] Pedro Henriques Abreu, Daniel Castro Silva, Joao Portela, Joao Mendes-Moreira, and Luís Paulo Reis. Using model-based collaborative filtering techniques to recommend the expected best strategy to defeat a simulated soccer opponent. *Intelligent Data Analysis*, 18(5):973–991, 2014.

[7] The soccer analytics revolution. https://sites.duke.edu/socceranalyticsrevolution/history-and-background/. Accessed: 2021-12-01.

[8] Market size of the european professional soccer. https://www.statista.com/statistics/261223/european-soccer-market-total-revenue/. Accessed: 2022-01-18.

[9] David Sally. *Numbers Game-why Everything You Know about Football is Wrong*. Penguin Books Limited, 2014.

[10] Opta sports. http://www.optasports.com/. Accessed: 2021-12-01.

[11] Wyscout. http://www.wyscout.com. Accessed: 2021-12-01.

[12] Stats perform. http://www.statsperform.com. Accessed: 2021-12-01.

[13] Tracab. https://tracab.com/, 2022. Accessed: 2022-03.

[14] Football data analytics. https://metrica-sports.com/football-data-analytics/, 2022. Accessed: 2022-03.

[15] Guiliang Liu, Yudong Luo, Oliver Schulte, and Tarak Kharrat. Deep soccer analytics: learning an action-value function for evaluating soccer players. *Data Mining and Knowledge Discovery*, 34(5):1531–1559, 2020.

[16] Carolina Centeio Jorge, Martin Atzmueller, Behzad M Heravi, Jenny L Gibson, Rosaldo JF Rossetti, and Cláudio Rebelo de Sá. “” want to come play with me? outlier subgroup discovery on spatio-temporal interactions. *EXPERT SYSTEMS*, 2021.

[17] Willi Klösgen. Explora: A multipattern and multistrategy discovery assistant. In *Advances in knowledge discovery and data mining*, pages 249–271. 1996.

[18] Stefan Wrobel. An algorithm for multi-relational discovery of subgroups. In *European symposium on principles of data mining and knowledge discovery*, pages 78–87. Springer, 1997.

[19] Dennis Leman, Ad Feelders, and Arno Knobbe. Exceptional model mining. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 1–16. Springer, 2008.

[20] Florian Lemmerich. *Novel techniques for efficient and effective subgroup discovery.* Bayerische Julius-Maximilians-Universitaet Wuerzburg (Germany), 2014.

[21] Stephen Cook. The p versus np problem. *The millennium prize problems*, pages 87–104, 2006.

[22] Nada Lavrac, Branko Kavsek, Peter Flach, and Ljupco Todorovski. Subgroup discovery with cn2-sd. *J. Mach. Learn. Res.*, 5(2):153–188, 2004.

[23] Martin Atzmueller, Frank Puppe, and Hans-Peter Buscher. Exploiting background knowledge for knowledge-intensive subgroup discovery. In *IJCAI*, pages 647–652, 2005.

[24] Dragan Gamberger and Nada Lavrac. Expert-guided subgroup discovery: Methodology and application. *Journal of Artificial Intelligence Research*, 17:501–527, 2002.

[25] Liaquat M Sheikh, Basit Tanveer, and MA Hamdani. Interesting measures for mining association rules. In *8th International Multitopic Conference, 2004. Proceedings of INMIC 2004.*, pages 641–644. IEEE, 2004.

[26] Nanlin Jin, Peter Flach, Tom Wilcox, Royston Sellman, Joshua Thumim, and Arno Knobbe. Subgroup discovery in smart electricity meter data. *IEEE Transactions on Industrial Informatics*, 10(2):1327–1336, 2014.

[27] María José Del Jesus, Pedro González, Francisco Herrera, and Mikel Mesonero. Evolutionary fuzzy rule induction process for subgroup discovery: a case study in marketing. *IEEE Transactions on Fuzzy Systems*, 15(4):578–592, 2007.

[28] Tarek Abudawood and Peter Flach. Evaluation measures for multi-class subgroup discovery. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 35–50. Springer, 2009.

[29] Matthijs Van Leeuwen and Arno Knobbe. Diverse subgroup set discovery. *Data Mining and Knowledge Discovery*, 25(2):208–242, 2012.

[30] Martin Atzmüller. *Knowledge-intensive subgroup mining: techniques for automatic and interactive discovery*, volume 307. IOS Press, 2007.

[31] Barbara FI Pieters, Arno Knobbe, and Sašo Dzeroski. Subgroup discovery in ranked data, with an application to gene set enrichment. In *Proceedings preference learning workshop (PL 2010) at ECML PKDD*, volume 10, pages 1–18, 2010.

[32] Liqiang Geng and Howard J Hamilton. Interestingness measures for data mining: A survey. *ACM Computing Surveys (CSUR)*, 38(3):9–es, 2006.

[33] Ken McGarry. A survey of interestingness measures for knowledge discovery. *The knowledge engineering review*, 20(1):39–61, 2005.

[34] Agoston E Eiben and Marc Schoenauer. Evolutionary computing. *Information Processing Letters*, 82(1):1–6, 2002.

[35] Bharat Gupta and Deepak Garg. Fp-tree based algorithms analysis: Fpgrowth, cofi-tree and ct-pro. *International Journal on Computer Science and Engineering*, 3(7):2691–2699, 2011.

[36] Francisco Berlanga, María José Del Jesus, Pedro González, Francisco Herrera, and Mikel Mesonero. Multiobjective evolutionary induction of subgroup discovery fuzzy rules: a case study in marketing. In *Industrial Conference on Data Mining*, pages 337–349. Springer, 2006.

[37] Branko Kavsek and Nada Lavrac. Using subgroup discovery to analyze the uk traffic data. *Metodoloski zvezki*, 1(1):249, 2004.

[38] Martin Atzmueller and Frank Puppe. Sd-map–a fast algorithm for exhaustive subgroup discovery. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 6–17. Springer, 2006.

[39] Willi Klösgen and Michael May. Spatial subgroup mining integrated in an object-relational spatial database. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 275–286. Springer, 2002.

[40] Dragan Gamberger, Nada Lavrač, and Goran Krstačić. Active subgroup mining: a case study in coronary heart disease risk group detection. *Artificial Intelligence in Medicine*, 28(1):27–57, 2003.

[41] Nada Lavrač and Dragan Gamberger. Relevancy in constraint-based subgroup discovery. In *Constraint-based mining and inductive databases*, pages 243–266. Springer, 2006.

[42] Filip Železnỳ and Nada Lavrač. Propositionalization-based relational subgroup discovery with rsd. *Machine Learning*, 62(1):33–63, 2006.

[43] Henrik Grosskreutz. Cascaded subgroups discovery with an application to regression. In *Proc. ECML/PKDD*, volume 5211, page 33. Citeseer, 2008.

[44] Adand Knobbe Arno Leman, Dennisand Feelders. Exceptional model mining. In Bartand Morik Katharina Daelemans, Walterand Goethals, editor, *Machine Learning and Knowledge Discovery inDatabases*, pages 1–16, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[45] Martin Atzmueller and Florian Lemmerich. Fast subgroup discovery for continuous target concepts. In *International Symposium on Methodologies for Intelligent Systems*, pages 35–44. Springer, 2009.

[46] Florian Lemmerich, Martin Atzmueller, and Frank Puppe. Fast exhaustive subgroup discovery with numerical target concepts. *Data Mining and Knowledge Discovery*, 30(3):711–762, 2016.

[47] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. Spea2: Improving the strength pareto evolutionary algorithm. *TIK-report*, 103, 2001.

[48] Cristóbal José Carmona, Pedro González, María José del Jesus, and Francisco Herrera. Nmeef-sd: Non-dominated multiobjective evolutionary algorithm for extracting fuzzy rules in subgroup discovery. *IEEE Transactions on Fuzzy Systems*, 18(5):958–970, 2010.

[49] Victoria Pachón, Jacinto Mata, Juan Luis Domínguez, and Manuel J Maña. Multi-objective evolutionary approach for subgroup discovery. In *International Conference on Hybrid Artificial Intelligence Systems*, pages 271–278. Springer, 2011.

[50] Florian Lemmerich and Martin Becker. pysubgroup: Easy-to-use subgroup discovery in python. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 658–662. Springer, 2018.

[51] Martin Atzmueller and Florian Lemmerich. Vikamine–open-source subgroup discovery, pattern mining, and analytics. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 842–845. Springer, 2012.

[52] Martin Atzmueller, Maintainer Martin Atzmueller, and SystemRequirements Java. Package ‘rsubgroup’. 2021.

[53] Angel M Garcia, Francisco Charte, Pedro González, Cristóbal J Carmona, and María José del Jesus. Subgroup discovery with evolutionary fuzzy systems in r: The sdefsr package. *R J.*, 8(2):307, 2016.

[54] Marvin Meeng and Arno Knobbe. Flexible enrichment with cortana–software demo. In *Proceedings of BeneLearn*, pages 117–119, 2011.

[55] Nada Lavrač, Bojan Cestnik, Dragan Gamberger, and Peter Flach. Decision support through subgroup discovery: three case studies and the lessons learned. *Machine Learning*, 57(1):115–143, 2004.

[56] Laurentius A Meerhoff, Arie-Willem de Leeuw, Floris R Goes, and Arno Knobbe. Mining soccer data: Subgroup discovery of tactics from spatio-temporal data.

[57] Jan Van Haaren, Pieter Robberechts, Tom Decroos, Lotte Bransen, and Jesse Davis. Analyzing performance and playing style using ball event data. 2019.

[58] Jan Van Haaren, Pieter Robberechts, Tom Decroos, Lotte Bransen, and Jesse Davis. Analysing performance and playing style using ball event data. 2019.

[59] Tom Decroos, Lotte Bransen, Jan Van Haaren, and Jesse Davis. Actions speak louder than goals: Valuing player actions in soccer. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1851–1861, 2019.

[60] Shoji Hirano and Shusaku Tsumoto. Grouping of soccer game records by multiscale comparison technique and rough clustering. In *Fifth International Conference on Hybrid Intelligent Systems (HIS’05)*, pages 6–pp. IEEE, 2005.

[61] Tiago Mendes-Neves and João Mendes-Moreira. Comparing state-of-the-art neural network ensemble methods in soccer predictions. In *International Symposium on Methodologies for Intelligent Systems*, pages 139–149. Springer, 2020.

[62] Tom Decroos. Soccer analytics meets artificial intelligence: Learning value and style from soccer event stream data. 2020.

[63] Tom Decroos, Maaike Van Roy, and Jesse Davis. Soccermix: Representing soccer actions with mixture models. In *ECML/PKDD (5)*, pages 459–474, 2020.

[64] Lars Magnus Hvattum and Halvard Arntzen. Using elo ratings for match result prediction in association football. *International Journal of forecasting*, 26(3):460–470, 2010.

[65] Anthony Costa Constantinou and Norman Elliott Fenton. Determining the level of ability of football teams by dynamic ratings based on the relative discrepancies in scores between adversaries. *Journal of Quantitative Analysis in Sports*, 9(1):37–50, 2013.

[66] Sciskill index – why and how. `https://www.scisports.com/sciskill-index-why-and-how/#`. Accessed: 2021-12-20.

[67] Olav Drivenes Sæbø and Lars Magnus Hvattum. Evaluating the efficiency of the association football transfer market using regression based player ratings. In *Norsk IKT-konferanse for forskning og utdanning*, 2015.

[68] Luca Pappalardo, Paolo Cintia, Paolo Ferragina, Emanuele Massucco, Dino Pedreschi, and Fosca Giannotti. Playerank: data-driven performance evaluation and player ranking in soccer via a machine learning approach. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(5):1–27, 2019.

[69] Ian G McHale, Philip A Scarf, and David E Folker. On the development of a soccer player performance rating system for the english premier league. *Interfaces*, 42(4):339–351, 2012.

[70] Abraham Charnes, William W Cooper, and Edwardo Rhodes. Measuring the efficiency of decision making units. *European journal of operational research*, 2(6):429–444, 1978.

[71] Tom Decroos, Vladimir Dzyuba, Jan Van Haaren, and Jesse Davis. Predicting soccer highlights from spatio-temporal match event streams. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[72] Tom Decroos, Jan Van Haaren, Vladimir Dzyuba, and Jesse Davis. Starss: a spatio-temporal action rating system for soccer. In *Machine Learning and Data Mining for Sports Analytics ECML/PKDD 2017 workshop*, volume 1971, pages 11–20. Springer, 2017.

[73] Tom Decroos, Lotte Bransen, Jan Van Haaren, and Jesse Davis. Vaep: An objective approach to valuing on-the-ball actions in soccer. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4696–4700. International Joint Conferences on Artificial Intelligence Organization, 2020.

[74] William Spearman. Beyond expected goals. In *Proceedings of the 12th MIT sloan sports analytics conference*, pages 1–17, 2018.

[75] José Carlos Coutinho, João Mendes Moreira, and Cláudio Rebelo de Sá. Unfoot: Unsupervised football analytics tool. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 786–789. Springer, 2019.

[76] Laurentius Antonius Meerhoff, Floris R Goes, A Leeuw, W De, and A Knobbe. Exploring successful team tactics in soccer tracking data. In *Joint european conference on machine learning and knowledge discovery in databases*, pages 235–246. Springer, 2019.

[77] Jan Van Haaren, Vladimir Dzyuba, Siebe Hannosset, and Jesse Davis. Automatically discovering offensive patterns in soccer match data. In *International Symposium on Intelligent Data Analysis*, pages 286–297. Springer, 2015.

[78] Lars Tijssen. Analyzing offensive player-and team performance in soccer using position data.

[79] Plotly python open source graphing library. `https://plotly.com/python/`. Accessed: 2021-12-17.

[80] Socceraction documentation. `https://socceraction.readthedocs.io/en/latest/#`. Accessed: 2021-12-22.

[81] Opta data from stats perform. `https://www.statsperform.com/opta/`, 2022. Accessed: 2022-03.

[82] Improving football performance. `https://www.scisports.com/`, 2022. Accessed: 2022-03.

[83] William Spearman. Quantifying pitch control. 2016.

[84] scikit-learn machine learning in python. `https://scikit-learn.org/stable/index.html`, 2022. Accessed: 2022-03.

[85] Florian Lemmerich and Martin Becker. pysubgroup: Easy-to-use subgroup discovery in python. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 658–662. Springer, 2018.

[86] Marvin Meeng and Arno Knobbe. Flexible enrichment with cortana–software demo. In *Proceedings of BeneLearn*, pages 117–119, 2011.

[87] Ian Dragulet. Modeling expected goals. *Medium*, Jan 2021.

[88] Glenn W Brier et al. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

Appendices

This page is intentionally left blank.

Appendix A

xG Models Results

	Features	Model	Threshold	ROC AUC	R^2	xG Avg	xG Goal Avg	xG Miss Avg	xG Diff	Brier Score
0	All Features	XGBClassifier	1.0	0.802	0.99	0.106	0.739	0.024	0.715	0.022
1	All Features	XGBClassifier	0.5	0.801	0.989	0.12	0.852	0.025	0.827	0.02
2	All Features	XGBClassifier	0.3	0.799	0.991	0.125	0.867	0.028	0.839	0.023
3	All Features	XGBClassifier	0.2	0.797	0.99	0.13	0.883	0.032	0.851	0.026
4	All Features	LogisticRegression	0.5	0.796	0.882	0.116	0.258	0.098	0.16	0.086

Table A.1: Top-5 models according to ROC AUC metric

	Features	Model	Threshold	ROC AUC	R^2	xG Avg	xG Goal Avg	xG Miss Avg	xG Diff	Brier Score
0	All Features	XGBClassifier	0.3	0.799	0.991	0.125	0.867	0.028	0.839	0.023
1	All Features	XGBClassifier	1.0	0.802	0.99	0.106	0.739	0.024	0.715	0.022
2	All Features	XGBClassifier	0.2	0.797	0.99	0.13	0.883	0.032	0.851	0.026
3	All Features	XGBClassifier	0.5	0.801	0.989	0.12	0.852	0.025	0.827	0.02
4	Top 5 relevants Feature	DecisionTreeClassifier	0.5	0.603	0.986	0.114	0.847	0.019	0.828	0.034

Table A.2: Top-5 models according to R^2 metric

	Features	Model	Threshold	ROC AUC	R^2	xG Avg	xG Goal Avg	xG Miss Avg	xG Diff	Brier Score
0	All Features	XGBClassifier	0.5	0.801	0.989	0.12	0.852	0.025	0.827	0.02
1	All Features	XGBClassifier	1.0	0.802	0.99	0.106	0.739	0.024	0.715	0.022
2	All Features	XGBClassifier	0.3	0.799	0.991	0.125	0.867	0.028	0.839	0.023
3	All Features	XGBClassifier	0.2	0.797	0.99	0.13	0.883	0.032	0.851	0.026
4	Top 10 relevants Feature	XGBClassifier	0.5	0.786	0.973	0.129	0.76	0.047	0.713	0.03

Table A.3: Top-5 models according to Brier Score metric

	Features	Model	Threshold	ROC AUC	R^2	xG Avg	xG Goal Avg	xG Miss Avg	xG Diff	Brier Score
0	All Features	XGBClassifier	0.2	0.797	0.99	0.13	0.883	0.032	0.851	0.026
1	All Features	XGBClassifier	0.3	0.799	0.991	0.125	0.867	0.028	0.839	0.023
2	All Features	DecisionTreeClassifier	1.0	0.616	0.984	0.119	0.857	0.023	0.834	0.036
3	All Features	DecisionTreeClassifier	0.5	0.616	0.984	0.119	0.857	0.023	0.834	0.036
4	All Features	DecisionTreeClassifier	0.3	0.616	0.984	0.119	0.857	0.023	0.834	0.036

Table A.4: Top-5 models according to xG Diff metric

This page is intentionally left blank.

Appendix B

Subgroup Discovery Results

Nr.	Depth	Coverage	Quality	Probability	Positives	Conditions
1	2	324	0.036	0.26	83	HeightAttacking >= 40.16 AND DefendersAt3MetersRadius <= 1.0
2	2	327	0.035	0.25	83	DefendersAt3MetersRadius <= 1.0 AND HeightAttacking >= 40.06
3	2	379	0.034	0.23	89	HeightAttacking >= 40.16 AND StrikerSpeed >= 3.52
4	2	408	0.034	0.23	93	HeightAttacking >= 40.16 AND DefendersAt3MetersRadius <= 2.0
5	2	376	0.034	0.23	88	HeightAttacking >= 40.16 AND DefendersAt7MetersRadius <= 4.0
6	2	369	0.034	0.24	87	HeightAttacking >= 32.94 AND HeightAttacking >= 41.52
7	2	369	0.034	0.24	87	HeightAttacking >= 40.16 AND NumberDribbles <= 0.0
8	2	398	0.034	0.23	91	HeightAttacking >= 40.16 AND Corner = '0'
9	2	406	0.034	0.23	92	HeightAttacking >= 40.16 AND DefendersAt7MetersRadius <= 5.0
10	1	421	0.033	0.22	94	HeightAttacking >= 40.16
11	2	421	0.033	0.22	94	HeightAttacking >= 29.97 AND HeightAttacking >= 40.16
12	2	371	0.033	0.23	87	HeightAttacking >= 37.49 AND HeightAttacking >= 41.48
13	2	407	0.033	0.23	92	HeightAttacking >= 37.49 AND DefendersAt3MetersRadius <= 1.0
14	2	379	0.033	0.23	88	HeightAttacking >= 40.16 AND ShotAngleFromTheGoal <= 39.56
15	2	379	0.033	0.23	88	HeightAttacking >= 40.16 AND ShotAngleFromTheGoal >= -44.97
16	2	379	0.033	0.23	88	HeightAttacking >= 40.16 AND AttackingTeamSpread >= 5.4
17	2	379	0.033	0.23	88	HeightAttacking >= 40.16 AND MoreLikelyPitchControl <= 92.67
18	2	379	0.033	0.23	88	HeightAttacking >= 40.16 AND CounterAttack >= 0.116
19	2	337	0.033	0.24	82	HeightAttacking >= 40.16 AND ShotAngleFromTheGoal >= -30.99
20	2	337	0.033	0.24	82	HeightAttacking >= 40.16 AND StrikerSpeed >= 5.3

Table B.1: Top-20 subgroups from TeamA without the ShotDistanceToGoal feature (with a total of 1052 plays)

Nr.	Depth	Coverage	Quality	Probability	Positives	Conditions
1	2	245	0.036	0.22	54	PitchControlAttacking <= 76.35 AND DefendersAt7MetersRadius <= 2.0
2	2	266	0.035	0.21	56	DefendersAt7MetersRadius <= 2.0 AND PitchControlAttacking <= 78.74
3	2	214	0.034	0.22	49	DistanceBetweenCentroids <= 5.51 AND DefendersAt7MetersRadius <= 2.0
4	2	208	0.034	0.23	48	DefendersAt7MetersRadius <= 2.0 AND DistanceBetweenCentroids <= 5.35
5	2	155	0.034	0.26	41	WidthAttacking <= 33.17 AND DefendersAt7MetersRadius <= 2.0
6	2	178	0.034	0.25	44	DefendersAt7MetersRadius <= 2.0 AND WidthAttacking <= 34.79
7	2	180	0.033	0.24	44	WidthAttacking <= 34.83 AND DefendersAt7MetersRadius <= 2.0
8	2	266	0.033	0.21	55	DefendersAt7MetersRadius <= 2.0 AND StrikerSpeed >= 3.9
9	2	251	0.033	0.21	53	MoreLikelyPitchControl <= 91.8 AND DefendersAt7MetersRadius <= 2.0
10	2	237	0.033	0.22	51	DefendersAt7MetersRadius <= 2.0 AND MoreLikelyPitchControl <= 90.87
11	2	270	0.032	0.20	55	DefendersAt7MetersRadius <= 2.0 AND NumberPasses <= 8.0
12	2	148	0.032	0.26	39	DefendersAt7MetersRadius <= 2.0 AND WidthAttacking <= 32.79
13	2	210	0.032	0.22	47	WidthAttacking <= 36.58 AND DefendersAt7MetersRadius <= 2.0
14	2	266	0.031	0.20	54	DefendersAt7MetersRadius <= 2.0 AND WidthAttacking <= 42.26
15	2	266	0.031	0.20	54	DefendersAt7MetersRadius <= 2.0 AND MoreLikelyPitchControl <= 93.33
16	2	190	0.031	0.23	44	NumberEvents <= 6.0 AND WidthAttacking <= 31.87
17	2	236	0.031	0.21	50	DefendersAt7MetersRadius <= 2.0 AND CounterAttack >= 0.191
18	2	207	0.031	0.22	46	DefendersAt7MetersRadius <= 2.0 AND WidthAttacking <= 36.36
19	2	131	0.031	0.27	36	DefendersAt7MetersRadius <= 1.0 AND PitchControlAttacking <= 78.74
20	2	246	0.031	0.21	51	WidthAttacking <= 33.17 AND MoreLikelyPitchControl <= 91.47

Table B.2: Top-20 subgroups from TeamB without the ShotDistanceToGoal feature (with a total of 613 plays)

Nr.	Depth	Coverage	Quality	Probability	Positives	Conditions
1	2	169	0.025	0.17	28.0	ShotAngleFromTheGoal >= -41.59 AND HeightAttacking >= 44.16
2	2	131	0.023	0.18	24.0	GoalKeeperDistanceFromTheGoal >= 2.27 AND HeightAttacking >= 45.7
3	2	149	0.022	0.17	25.0	ShotAngleFromTheGoal >= -28.66 AND HeightAttacking >= 44.17
4	2	149	0.022	0.17	25.0	BallSpeed >= 21.99 AND HeightAttacking >= 44.13
5	2	189	0.022	0.15	28.0	HeightAttacking >= 38.62 AND AttackingTeamSpread >= 6.76
6	2	112	0.022	0.20	22.0	ShotAngleFromTheGoal >= -41.59 AND HeightAttacking >= 47.17
7	2	112	0.022	0.20	22.0	ShotAngleFromTheGoal >= -10.33 AND HeightAttacking >= 44.32
8	2	112	0.022	0.20	22.0	StrikerSpeed >= 8.45 AND HeightAttacking >= 44.3
9	2	100	0.022	0.21	21.0	BallSpeed >= 21.99 AND HeightAttacking >= 46.81
10	2	156	0.021	0.16	25.0	AttackingTeamSpread >= 6.5 AND HeightAttacking >= 43.85
11	2	131	0.021	0.18	23.0	StrikerSpeed >= 6.65 AND HeightAttacking >= 44.13
12	2	131	0.021	0.18	23.0	BallSpeed >= 35.38 AND HeightAttacking >= 44.13
13	2	131	0.021	0.18	23.0	BallSpeed >= 35.38 AND HeightDefending >= 40.1
14	2	224	0.021	0.13	30.0	HeightAttacking >= 38.62 AND AttackingTeamSpread >= 6.51
15	2	132	0.021	0.17	23.0	ShotAngleFromTheGoal >= -20.31 AND HeightAttacking >= 44.17
16	2	93	0.021	0.22	20.0	HeightAttacking >= 44.6 AND BallSpeed >= 89.19
17	2	199	0.021	0.14	28.0	MaxPitchControl <= 4.13 AND HeightAttacking >= 44.17
18	2	199	0.021	0.14	28.0	CounterAttack >= 0.175 AND HeightAttacking >= 43.55
19	2	199	0.021	0.14	28.0	PlayDuration <= 28.96 AND PitchControlAttacking <= 66.84
20	2	186	0.021	0.15	27.0	AttackingTeamSpread >= 6.19 AND HeightAttacking >= 43.38

Table B.3: Top-20 subgroups from TeamC without the ShotDistanceToGoal feature (with a total of 619 plays)