Machine Learning for Soccer Analytics

Thesis · September 2013

DOI: 10.13140/RG.2.1.4628.3761

1 author:

Gunjan Kumar
University College Dublin

KU LEUVEN

Machine Learning for Soccer Analytics

Gunjan Kumar

Thesis submitted for the degree of Master of Science in Artificial Intelligence, option Engineering and Computer Science

Thesis supervisor:
Prof. dr. Jesse Davis

Assessors:
Prof. dr. ir. Gerda Janssens
ir. Jan Van Haaren

Mentors:
Daniele Alfarone
ir. Jan Van Haaren

Academic year 2012 – 2013

Preface

I thank my thesis supervisor for fuelling my interest in Machine Learning and motivating me throughout this thesis. I would like to extend special thanks to my mentors for their immense support and constant motivation to analyse deeply and do better research work for this thesis. I would also like to thank my friends and family for their encouragement and support.

Gunjan Kumar

Contents

Abstract

Sports analytics has been successfully applied in sports like baseball and basketball. However, its application in soccer has been limited. Research in soccer analytics with Machine Learning techniques is limited and is mostly employed only for predictions. There is a need to find out if the application of Machine Learning can bring better and more insightful results in soccer analytics. In this thesis, we perform descriptive as well as predictive analysis of soccer matches and player performances.

In soccer, it is popular to rely on ratings by experts to assess a player's performance. However, the experts do not unravel the criteria they use for their rating. We attempt to identify the most important attributes of player's performance which determine the expert ratings. In this way we find the latent knowledge which the experts use to assign ratings to players. We performed a series of classifications with three different pruning strategies and an array of Machine Learning algorithms. The best results for predicting ratings using performance metrics had mean absolute error of 0.17. We obtained a list of most important performance metrics for each of the four playing positions which approximates the attributes considered by the experts for assigning ratings. Then we find the most influential performance metrics of the players for determining the match outcome and we examine the extent to which the outcome is characterised by the performance attributes of the players. We found 34 performance attributes using which we can predict the match outcome with an accuracy of 63.4%. Next, we devise a method to aggregate individual player ratings to produce a set of team ratings and investigate how closely these team ratings can determine the match outcome. We create 27 different team rating attributes, and using 8 of these attributes aggregated over the player ratings for the current match we obtained a prediction accuracy of 90%. This indicates that the expert ratings are influenced by the match outcome. Then we use a weighted average method to aggregate team ratings over past match performances. We find that match outcomes are best characterised by the ratings for the current match. Next, we investigate how well the expert ratings for the past performances of players predict the next match outcome. We find that we could only manage a prediction accuracy of 53.4% because of the unpredictability of 'draw'. We conclude that the outcome of a match is correlated to the ratings for the current match only.

List of Figures and Tables

List of Figures

List of Tables

List of Abbreviations

Abbreviations

EPL	English Premier League 2011-12
Home-team	Team for which the match venue is home.
Away-team	Team for which the match venue is not home.
AH	Average of home-team players' statistics
AA	Average of away-team players' statistics
SH	Sum of home-team players' statistics
SA	Sum of away-team players' statistics
CC	Correlation Coefficient
MAE	Mean Absolute Error
RMSE	Root Mean Square Error
MSE	Mean Square error
RelAE	Relative Absolute Error
BTime	Model Building Time in seconds
TTime	Model Testing Time in seconds
ROC	Weighted (by class size) average receiver operating characteristic area
Kappa	Value of kappa statistic
F-Measure	Weighted (by class size) average F-Measure
AdditiveReg(DS)	Additive Regression with Decision Stump
RegByDiscret(J48)	Regression by Discretisation with J48
SimpleLinearReg	Simple Linear Regression
SMOreg	Support vector machine for regression
ClasViaReg	Classification Via Regression
LeastMedSq	Least median squared linear regression
M5P	Base routines for generating M5 Model trees and rules
LWL	Locally weighted learning
Ada	AdaBoostM1 Algorithm
IBk	K-nearest neighbours classifier
RBF	Radial basis function
FURIA	Fuzzy Unordered Rule Induction Algorithm
SMO	Sequential Minimal Optimization Algorithm
FT	Functional trees
JRip	A propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction (RIPPER),
SVM	Support Vector Machines
num	Number
SD	Standard Deviation
Thresh	Threshold

Chapter 1

Introduction

Sports analytics is the investigation and modelling of professional sports performance and contests using scientific techniques. This discipline frequently employs principles and techniques from statistics, data mining, game theory, biomechanics, kinesiology, etc. Sports analytics has been successfully applied to baseball and basketball. However, there has not been much application of sports analytics in soccer, even though soccer is one of the most popular sports in the world. The major reason being the lack of publicly available data in soccer. Last year, in order to foster performance analytics in soccer, Manchester City Football Club in cooperation with data provider OPTA made an extensive dataset for the 2011-2012 English Premier League season publicly available, which contains detailed statistics for all players involved in each match of the Premier League. This provides us a unique opportunity to explore this resource with the use of suitable techniques to gain insight into the evaluation of player and team performances.

Soccer is one of the best examples of team work. Even though the success of a team is ordinarily defined in terms of its wins and losses, these wins and losses are the results of complex and intriguing interactions between the performances of individual players of the two teams and also the uncertainties pertinent to the situation. One might even say that these interactions and uncertainties in the nature of this game make soccer a beautiful game. Because of this complex nature of interaction, it is difficult to decide upon the suitable set of attributes which can be used to evaluate an individual player's performance. Currently it is popular to rely on soccer experts that typically assign ratings to players and teams, to assess and compare their performances. These ratings are published in newspapers and sport websites. These player ratings are so popular amongst soccer fans that they gave birth to a game called Fantasy Football [Wikipedia, 2013a], in which participants assemble an imaginary team of real life footballers and score points based on those players' actual statistical performance. However, these performance ratings are assigned by soccer pundits based on their experience and understanding of the game. It is noteworthy that this rating assignment is "black-box" in the sense that the experts cannot precisely elucidate the criteria they used for their rating. We can say that there is a latent knowledge used by experts for these ratings which might be difficult to define. One of our goals in this thesis will be to shed light on this hidden knowledge.

Sports performance analysis enables the coach, players and the managers to objectively assess and thereby improve their sporting performance. The use of computational and statistical approaches in sports analytics is gaining popularity especially in relation to performance comparison, visualisation and prediction of match outcome. In this thesis, we will apply techniques from machine learning to the player statistic data to analyse player performances and observe how the player performances affect the match outcomes.

1.1 Objectives of the Thesis

In this thesis, we use machine learning techniques on player performance data to achieve the following objectives:

1. To identify the most important attributes of player's performance which determine their ratings given by experts. In this way we are finding out the latent knowledge which the experts use to assign ratings to players.

2. To find out which performance attributes of the players of the two competing teams affect the match outcome and to what extend the match outcome is characterised by the performance attributes of players.

3. Find out good ways in which individual player ratings can be aggregated resulting in a set of team ratings and investigate how closely these team ratings can determine the match outcome. This will indicate the correlation between expert ratings and match outcomes. Hence, it will show the influence of match outcomes on the expert ratings.

4. To investigate how well the expert ratings given to the players of a team in the past performances of the team predict the next match outcome.

1.2 Literature Review

Sports Analytics has been historically most active in baseball. The reason being the accessibility of baseball data and a growing analytics community behind it. Sabermetrics is the specialist analysis of baseball through objective evidence. It mostly uses baseball statistics which measure the in-game activity [Wikipedia, 2013c]. Sabermetrics has been around since 1960 and it gained popularity with the success of the methods of Bill James. The methods of Bill James were highlighted in the 2003 book Moneyball by Michael Lewis [Lewis, 2003]. Sabermetrics is more about finding and using new and unfamiliar statistics. And these novel statistics are formulated by the experts based on their understanding of the game. For example, "Runs Created", one of the most famous statistics used to evaluate offense. Bill James working through his own logic, formulated "Runs Created" which is intended to provide a formal way of estimating how a batting line translates into runs.

Formal games are classified into four major categories, namely invasion, net/wall, striking/fielding, and target games based on game structure [Ellis, 1983, Hooper, 1998, Werner and Almond]. Soccer is an invasion game whereas baseball is a striking/fielding game. As opposed to other game forms, invasion games are characterised by the use of a goal or similar target for scoring, common tactical features of invading territory to make space in attack, and the containment of space in defence [Bunker and Thorpe]. The continuous opposition and dynamic structure make invasion games like soccer, more complex than other game forms, and thus more difficult to analyse adequately. Any performance analysis in invasion games should therefore be structured by the help of a notational analysis system - either simple hand notation or sophisticated computerised notational systems [Tenga, 2010]. Pollard et al. [1988] reported that the objective analysis of match performance in soccer has long been hindered by the continuous and fast-moving nature of the game. Baseball is a less fluid sports than soccer. It is easier to break down a baseball game into discrete events to be analysed. The same cannot be said about soccer, hence it is tougher to generate data for soccer matches. Unlike baseball, performance analysis in soccer suffered from the lack of rich data. In addition, the introduction of analytics to practice was resisted by those who held to the traditional view that experienced coaches were able to observe freely and report accurately the key aspects of match performance [McGarry and Franks, 2003]. However, due to the increasing interest in soccer analytics and better availability of technology, the situation has recently started changing. Companies like OPTA, Prozone and Amisco are now able to collect rich soccer data and provide it to soccer clubs to analyse their performance. On 17 August 2012, a rich dataset of player performance statistics was made publicly accessible for the first time. This dataset released by OPTA, contains 210 attributes of players for the entire English Premier League season 2011-12.

If we look at the performance analytics literature related to soccer, we realise that most of the research is being done with few performance variables and is dependent on understanding the structure of the game. One of the most comprehensive studies about performance analytics was published as a book in 2007, “The Essentials of Performance Analysis” by Hughes and Franks. This book identifies the different types of performance analysis, namely notational and bio-mechanical. Notational analysis is an objective way of recording performance, such that critical events in that performance can be quantified in a consistent and reliable manner. This enables quantitative and qualitative feedback that is accurate and objective [Hughes and Franks, 2007]. In this thesis, we limit our scope to notational analysis. Notational analysis can be traced back to as early as 1966 and the Washington Redskins, American football team were among the first to use them in 1968 [Hughes and Franks, 2004].

Researchers have investigated questions like whether “possession play” or “direct play” is more effective in goal scoring in soccer [Bate, 1988, Hughes and Franks, 2005, Hughes et al., 1988, Olsen and Larsen, 1988, Reep and Benjamin, 1968]. Studies have shown evidence on both sides, supporting “possession play” [Hughes et al., 1988, Hughes and Churchill, 2004, Hughes and Franks, 2005, Hughes and Snook, 2006] or “direct play” [Bate, 1988, Hughes, 1990, Olsen and Larsen, 1988, Reep and

Benjamin, 1968] as a more effective playing style. The original work of Reep and Benjamin [1968] is considered to be a landmark in match performance analysis in soccer [Hughes and Franks, 1997]. They worked on data collected from 3213 matches played between 1953 and 1968. These data on goal scoring and the length of passing sequences were analysed statistically and appeared to follow a probability structure. The two main findings from this research are that approximately 80% of goals result from a sequence of three passes or less, and that a goal is scored in every 10 shots [Tenga, 2010]. These findings have been reconfirmed by several different studies [Bate, 1988, Franks, 1988, Hughes, 1990]. In short, Reep and his colleague showed that a successful style of play can be built by maximising the “chance” elements of the game [Reep and Benjamin, 1968, Tenga, 2010]. Furthermore, team possessions originating from the final third of the playing field were found to be effective in goal scoring [Bate, 1988, Hughes, 1990, Hughes and Snook, 2006]. Also, mixed results were obtained in Collet [2012] where ball possession was found to be correlated with match outcome only in the cases of elite teams. One interesting effect of the study by Reep and Benjamin [1968] is that these findings may have shaped the tactics of British football. Most coaches have been affected, to a greater or lesser extent, by the tactics referred to as the “long-ball game” or “direct play”, which was a tactic employed as a consequence of this research [Hughes and Franks, 2004]. In the same study [Hughes and Franks, 2004] strongly recommend that more consideration should be given to normalising frequency data with respect to the relevant total datasets [Hughes and Bartlett, 2002].

It has been suggested that a multivariate statistical approach should be used instead of univariate analysis. Most studies have used a univariate approach to compare a single tactical factor between successful and unsuccessful groups of performances. Because of the complexity of soccer match performance analysis and possible interaction of multiple tactical factors, a multivariate statistical approach needs to be used to study potential tactical factors and their interaction [Tenga, 2010]. The study also points to the fact that conclusions drawn from several related studies can be questioned on the grounds of inadequate sample size. Also a key question in performance analysis is deciding which attributes should be considered key performance indicators. Most of the times these attributes are constructed from the expert understanding of the game. In Hughes et al. (2012) several performance analysts gathered together to decide upon the key performance indicators for each player position in soccer. The performance indicators can be broadly classified to three types namely, Biomechanical performance indicators like “Ball release velocity”, Technical performance indicators like “Tackles won and lost”, and Tactical performance indicators like “Length of passes” [Hughes and Bartlett, 2002].

Another aspect of sports analytics is modelling the game. Early studies in this regard includes [Mosteller, 1979], he developed a predictive model which also provided several guidelines to reduce the effect of outliers. These guidelines are still used by the researchers in this domain. According to Hughes and Franks [2007] modelling the performance in sport can be put under the following generic headings: Empirical models, Dynamic systems, Statistical techniques and, Artificial Intelligence (Expert Systems, Artificial Neural Networks). It would not be incorrect to say that most of the research in modelling in this domain has been done using statistical techniques. The popular methods in baseball and basketball predictions are based upon statistical techniques. In soccer analytics, the statistical approaches have gained more attention in recent years [Karlis and I., 1998]. Constantinou et al. [2012] agree that a common approach in soccer analytics is to use Poisson distribution for goal-based data analysis where match results are generated by the attack and defence parameters of the two competing teams [Maher, 1982, Lee, 1997, Dixon and Coles, 1997, Karlis and Ntzoufras, 2003]. Multinomial logistic regression models have also been tried for this modelling [Liu and Zhang, 2008]. While the Poisson models predict the number of goals conceded and scored, the other statistical models restrict their prediction to match result. However, when goal driven models and match results driven model were compared in Goddard [2005], it was found that both approaches yield almost similar prediction performance. Recently, artificial intelligence approaches to the modelling problem have been attempted as well. Constantinou et al. [2012] used a Bayesian network model, while Rotshtein et al. used fuzzy model with genetic and neural tuning to predict the match results. Rue and Øyvind Slavensen [1998] used a Bayesian approach combined with Markovian chains and the Monte-Carlo method. These models are complex, use many assumptions, require large statistical samplings, and may not always be easily interpreted [Rotshtein et al.]. Neural networks have been used to make predictions in several sports including American football [Purucker, 1996]. Neural network approximators lack interpretability and hence cannot be used for performance analysis or feedback but only for prediction.

We observe a general issue here that most of the work in soccer analytics has been done using data which has a very limited number of performance attributes. Also we note that only few studies in soccer analytics have been done using machine learning algorithms, and these studies were mostly limited to the prediction task and restricted to a few algorithms. There is a need to find out if the application of machine learning algorithms can bring better, more insightful results in soccer analytics. Therefore, we have set the four objectives of this thesis, and perform experiments to investigate the utility of machine learning techniques to achieve these objectives.

1.3 Scope

In this thesis, we will use the dataset obtained from OPTA as well as data collected from the popular soccer statistic website WhoScored.com[1]. We will limit ourselves to a notational analysis of soccer matches because of the intrinsic nature of the available data which contains a mix of technical and tactical performance indicators only. Since we know the outcome of all the concerned soccer matches, this machine learning task will be supervised. We will do extensive preprocessing to modify the prepared dataset for the suitability of each experiment. We will use the machine learning algorithms available in the WEKA Data Mining Software package [Hall

[1] http://www.whoscored.com/

et al., 2009] for our experiments and then compare and analyse the performances and results of the different algorithms.

1.4 Overview

The rest of this thesis is organised as follows. The second chapter contains the experiments for answering the first research question, to identify the most important attributes of player's performance which determine their ratings given by experts. Then in the third chapter we work upon our next objective namely to find out which performance attributes of the players of the two competing teams affect the match outcome. In the fourth chapter we work on third and fourth objectives. Here we attempt to figure out good ways to aggregate individual player ratings so that it results in a set of team ratings such that these team ratings determine the match outcome as closely as possible. In the same chapter we address the fourth objective of our thesis, hence, we investigate how well the past performances of the players predict the outcome of future matches. In the fifth chapter we draw our conclusions from all the experiments performed and summarise the results.

Chapter 2

Finding Performance Metrics behind Expert Ratings

In this chapter we provide a brief background of the domain and then begin our experiments for the first objective of the thesis. This objective is *to identify the most important attributes of player's performance which determine the ratings given by the experts.* In this way we will find the latent knowledge which the experts use to assign ratings to players. We prepare appropriate datasets for our experiments and then perform a series of classification tasks. During these classifications, we employ different pruning strategies to eliminate the least influential performance attributes. Thus, we will obtain lists of most influential attributes through different algorithms and strategies. We will select the optimal lists of attributes with respect to classification performance and merge them to obtain our final list of most influential player's performance attributes. This final list of attributes will be representative of the performance parameters considered by the experts while assigning the ratings.

2.1 Background

In this section we give a concise description of how soccer is played. Then we introduce the English Premier League and provide an overview of the ratings assigned by soccer experts to player performances.

2.1.1 Soccer

Soccer, also known as association football, is a sport played between two teams of eleven players each. It is played with a spherical ball on a rectangular field with one goal at each end. The objective of the competing teams is to put the soccer ball inside the goal of the opposite team, which is called a score. Out of the eleven players on each team, one of them is the *goalkeeper* and the rest of the ten players can play the roles of *attacker*, *defender* or *midfielder*. The goalkeepers are the only players allowed to touch the ball with their hands or arms while it is in play and then only in their penalty area. The other ten players of the team can only use their feet, head or torso to strike or pass the ball. The team that scores the higher number of goals is declared the winner. If the score of both teams are equal at the end of the game then either a draw is declared or the game goes into extra time and/or a penalty shootout depending on the format of the competition.

2.1.2 English Premier League 2011-12

The English Premier League is ali professional league for men's association soccer clubs. It is England's primary soccer competition. There are 20 clubs in the Premier League. During the course of a season each soccer club plays the others twice in a double round-robin system, once at their home stadium and once at that of their opponents, for a total of 38 games. Teams receive three points for a win and one point for a draw. No points are awarded for a loss. Teams are ranked by total points, then goal difference, and then goals scored. At the end of each season, the club with the most points is declared champion. If the points are equal, the goal difference and then goals scored determine the winner [Wikipedia, 2013b]. The 2011-12 season of the English Premier League began on 13 August 2011 and ended on 13 May 2012. A total of 380 matches were played and Manchester City won the title.

2.1.3 Expert Ratings

Expert ratings are the ratings or score assigned to the performance of the player or team by soccer experts. There are several popular expert ratings available like Capello Index, Castrol Index and ratings by soccer websites like WhoScored.com. Some of these ratings are cumulative, hence all the past performances of the player add up and he has a current standing rating. Some of them are more detailed containing a rating for a player in each match individually. Most of the times these ratings are not transparent and their complete method of ratings' assignment is not open to the public. These ratings are assigned using the opinion of soccer experts, i.e., using their expert knowledge. This expert knowledge used for the ratings' assignment is usually not explicit and not completely known. In this thesis we will attempt to find out how these expert ratings relate to performance metrics of players. We are interested in ratings assigned to each player in the 2011-12 English Premier League for their performance in each individual match. The expert ratings available from WhoScored.com are used for this purpose of this thesis.

2.2 Preparing the data

The dataset of player performances for EPL released by OPTA[1] contained 210 attributes and 10369 instances. Each instance represents the performance of a player in a match. Out of the 210 attributes of players in this dataset, 198 attributes were performance statistic while the others were identifiers for the player for that match. Next, we need the expert ratings for each performance of the player in each match.

[1] http://www.optasports.com/

For the expert ratings we consult WhoScored.com. We extract the ratings for each player for each match of the EPL from the website WhoScored.com[2]. Note that WhoScored.com also uses statistics collected by OPTA to compute its player ratings. If this weren't the case, the task of revealing the expert knowledge would most likely be much more difficult. We found some inconsistencies with the names of the players across the OPTA data and WhoScored.com data. We used the Levenshtein distance between player names to perform record linkage in our data. Thus, the expert ratings dataset was created. Next, the dataset from OPTA and the expert ratings dataset are joined together. Now each instance of our dataset contains a player performance metric from OPTA and an expert rating from WhoScored.com and is representative of a player's performance for a particular match of EPL. We recognise that a soccer player can play in either of the four positions, *attacker, defender, midfielder* or *goalkeeper*. Since these four roles are different, there should be different criteria to assign expert ratings to players belonging to different positions. Hence, we break our dataset in four parts: an attackers dataset, a defenders dataset, a midfielders dataset, and a goalkeepers dataset. However, it should be noted that dividing the players into only four positions is a simplification. It is possible to further split the player positions, for example an attacker can be a centre forward, a second striker or a winger, similarly a midfielder can be a centre midfield, a defensive midfield, an attacking midfield or a wide midfield, and a defender can be a centreback, a sweeper, a fullback or a wingback. For this thesis we restrict ourselves to only four player positions since our dataset has identifiers for only these positions. We remove those attributes from our dataset which are not representative of the player's performance. Thus we arrive at 195 attributes in each of our datasets.

2.3 Methodology

We have four datasets for four player positions. We perform classification for each of the datasets using the player performance attributes and the target class as "expert rating". We use the algorithms from the Java package of WEKA for classification. WEKA is a collection of machine learning algorithms for data mining tasks [Hall et al., 2009]. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. The algorithms from WEKA can either be applied directly to a dataset or can be used from their Java package. (For a brief introduction to the algorithms used, see Appendix: B). We perform some trials on our dataset using different classification algorithms and with different parameters. We use 10 fold stratified cross validation, to mitigate uneven representation in training and test sets and any biases caused by a particular sample of training data. Another motivation for this validation scheme is the lack of sufficient training data to split into a training, validation and test sets. Then, after observing the results of these classification tasks we choose the following set of classification algorithms for our task: *Linear Regression, SMOreg, Gaussian Processes, LeastMedSq, M5P, Bagging with REP Tree, Additive Regression with Decision Stump, REP Tree, Regression*

[2] http://www.whoscored.com/Statistics

by Discretization using J48, Decision Table, Multilayer perceptron, Simple Linear Regression, Locally Weighted Learning, IBk, KStar and RBF Network. Except for the ensemble based algorithms, we used the default parameters for the algorithms. We apply the set of classification algorithms on the four datasets and observe the results. We also subsequently prune the number of attributes in the datasets using three different strategies and observe and compare the classification performance.

In the first pruning strategy, which we call *Global Ranked Pruning*, we prune the attributes by first ranking the attributes in order of decreasing weights assigned by the best performing algorithms, and then progressively pruning the attributes from the bottom of the list in step size of 5. In the second pruning strategy, which we call *Iterative Local Pruning*, we prune the attributes for the next iteration of an algorithm execution based on the model generated by that algorithm in its previous execution. So the pruning is "local" to the characteristic of each algorithm's immediate execution. This pruning strategy is applicable only for algorithms whose generated model can be interpreted to derive a ranked list of attributes. This means that we cannot apply *Iterative Local Pruning* for algorithms like *Multilayer Perceptron*. The third pruning strategy is *Threshold Pruning* which is similar to *Iterative Local Pruning* except that we use a minimum threshold weight to select the attributes. Hence, this strategy only selects the attributes included in the model learnt in the previous execution of the algorithm and prunes the attributes whose weight is below a certain threshold value. The threshold is set in accordance to the standard deviation and mean of the weights assigned to the attributes included in the model learnt in the previous execution of the algorithm.

2.4 Results and Discussions

On all the four datasets, we find that the *Linear Regression* and *SMOreg* algorithms perform best. The results for the attackers dataset is shown in Table 2.1. In this table, 'CC' represents the correlation coefficient, 'MAE' represents mean absolute error, 'RMSE' represents 'root mean square error', 'RelAE' represents relative absolute error, while 'BTime' and 'TTime' represents model building and model training times. The results for other three positions can be found in Appendix: C. We were working with 195 attributes and next we will employ different strategies to prune the attributes.

2.4.1 Global Ranked Pruning

For each player dataset we do the following. We look up the model generated by the *Linear Regression* algorithm and we see a list of weighted attributes. We sort the attributes in this list in decreasing order of absolute value of weights. The sorted list now contains attributes in decreasing order of importance considered by the *Linear Regression* algorithm. We use this list for pruning. For example for attackers this list contains 116 attributes, the top 20 attributes are shown in Table 2.2. Similarly for defenders this list contains 94 attributes, the top 20 attributes are shown in Table 2.3. For midfielders this list contains 126 attributes, the top 20 attributes are shown in

TABLE 2.1: Algorithms results with Attackers dataset (195 attributes)

Algorithm	CC	MAE	RMSE	RelAE	BTime(s)	TTime(s)
LinearRegression	0.9646	0.1679	0.2379	24.1348	3.9160	31.3670
SMOreg	0.9646	0.1659	0.2386	23.8477	72.0060	520.9650
GaussianProcesses	0.9649	0.1682	0.2370	24.1756	25.3690	280.7680
LeastMedSq	0.9639	0.1724	0.2489	24.7807	11.3030	101.0300
M5P	0.9582	0.1806	0.2582	25.9657	4.5400	37.7600
Bagging	0.9145	0.2589	0.3659	37.2189	3.1050	24.8870
AdditiveReg(DS)	0.8998	0.2965	0.4000	42.6187	0.5250	4.2680
REPTree	0.8921	0.2934	0.4080	42.1705	0.3190	2.4310
RegByDiscret(J48)	0.8679	0.3235	0.4548	46.5085	1.1880	13.0030
DecisionTable	0.8489	0.3381	0.4770	48.6047	2.9630	25.1450
MultilayerPerceptron	0.7218	0.4975	0.7638	71.5081	461.7650	4167.5960
SimpleLinearReg	0.7674	0.4295	0.5784	61.7347	0.0310	0.1300
LWL	0.7459	0.4365	0.6032	62.7474	0.0010	99.0370
IBk	0.6912	0.4698	0.6936	67.5311	0.0000	3.9460
KStar	0.6422	0.4844	0.7368	69.6353	0.0000	129.9960
RBFNetwork	0.3394	0.6209	0.8485	89.2492	0.4740	5.4930

Table 2.4 and for goalkeepers this list contains 62 attributes, the top 20 attributes are shown in Table 2.5.

Now for each of the player dataset, using this ranked list of attributes we perform pruning of attributes. For attackers, we create a dataset containing only the 116 attributes considered important by Linear Regression, then we create next attacker dataset containing only top 111 attributes from the 116 attribute list, then the next dataset with only top 106 attributes from the list, and so on we go on pruning 5 attributes from the bottom of the ranked list and keep on creating new attacker datasets until we reach near only top 5 attributes. Similarly we do for defenders, midfielders and goalkeepers and hence obtain sets of pruned datasets. We call this pruning strategy as *Global Ranked Pruning*. Next, we execute our set of machine learning algorithms on all of these datasets. We record all the result summaries especially *mean absolute error, correlation coefficient, root mean square error, relative absolute error, model building time and model testing time*. We analyse all the models generated. We can see the trend of mean absolute error against the number of attributes for attackers for all the algorithms in our set in the figure 2.1. Next, we observe the trend for *mean absolute error* for different classes of algorithms namely, function based, tree based, rule based, lazy and meta algorithms separately on attackers dataset more clearly we can look at figures 2.2(functions), 2.3(tree), 2.5(rule and lazy) and 2.4(meta) respectively . (For results for other three player positions, see Appendix: C). We also take note of the model building time and model training time for the different algorithms and how it varies across decreasing number of attributes. The plot of model building time for attackers is shown in Figure 2.6 and the model training time is shown in Figure 2.7.

Table 2.2: Top 20 Attacker's attributes by Linear Regression using 195 attributes

Red Cards
Error leading to Goal
Last Man Tackle
Assists
Penalties Not Scored
Goals from Throws
Penalties Conceded
Successful Crosses Corners in the air
Goals from Set Play
Own Goal
Error leading to Attempt
Foul Won Penalty
Successful crosses in the air
Clearances Off the Line
Goals Open Play
Goals from Corners
Key Throw In
Goals from Outside Box
Fouls Conceded excluding handballs penalties
Successful Crosses Left

Table 2.3: Top 20 Defender's attributes by Linear Regression using 195 attributes

Red Cards
Error leading to Goal
Own Goal
Penalties Conceded
Assists
Shots Cleared off Line Outside
Goals from Direct Free Kick
Clearances Off the Line
Left Foot Goals
Unsuccessful Corners Left
Goals Open Play
Error leading to Attempt
Penalty Goals
Goals from penalties
Unsuccessful Corners Right
Goals from Outside Box
Goals from Set Play
Goals
Foul Won Penalty
Last Man Tackle

TABLE 2.4: Top 20 Midfielder's attributes by Linear Regression using 195 attributes

Goals from Throws
Goals from Set Play
Goals Open Play
Goals from Corners
Red Cards
Error leading to Goal
Penalties Conceded
Own Goal
Headed Goals
Other Goals
Right Foot Goals
Goals from Outside Box
Goals from Inside Box
Left Foot Goals
Goals
Direct Free-kick Goals
Goals from Direct Free Kick
Assists
Foul Won Penalty
Clearances Off the Line

TABLE 2.5: Top 20 GoalKeeper's attributes by Linear Regression using 195 attributes

Red Cards
Own Goal
Saves from Penalty
Error leading to Goal
Handballs Conceded
Fouls Won in Danger Area including penalties
Foul Won Penalty
First Goal
Attempts Open Play on target
Error leading to Attempt
Fouls Conceded excluding handballs penalties
Dispossessed
Total Unsuccessful Passes All
Penalties Conceded
Last Man Tackle
Goals Conceded Outside Box
Saves Made from Inside Box
Total Fouls Conceded
Substitute On
Interceptions

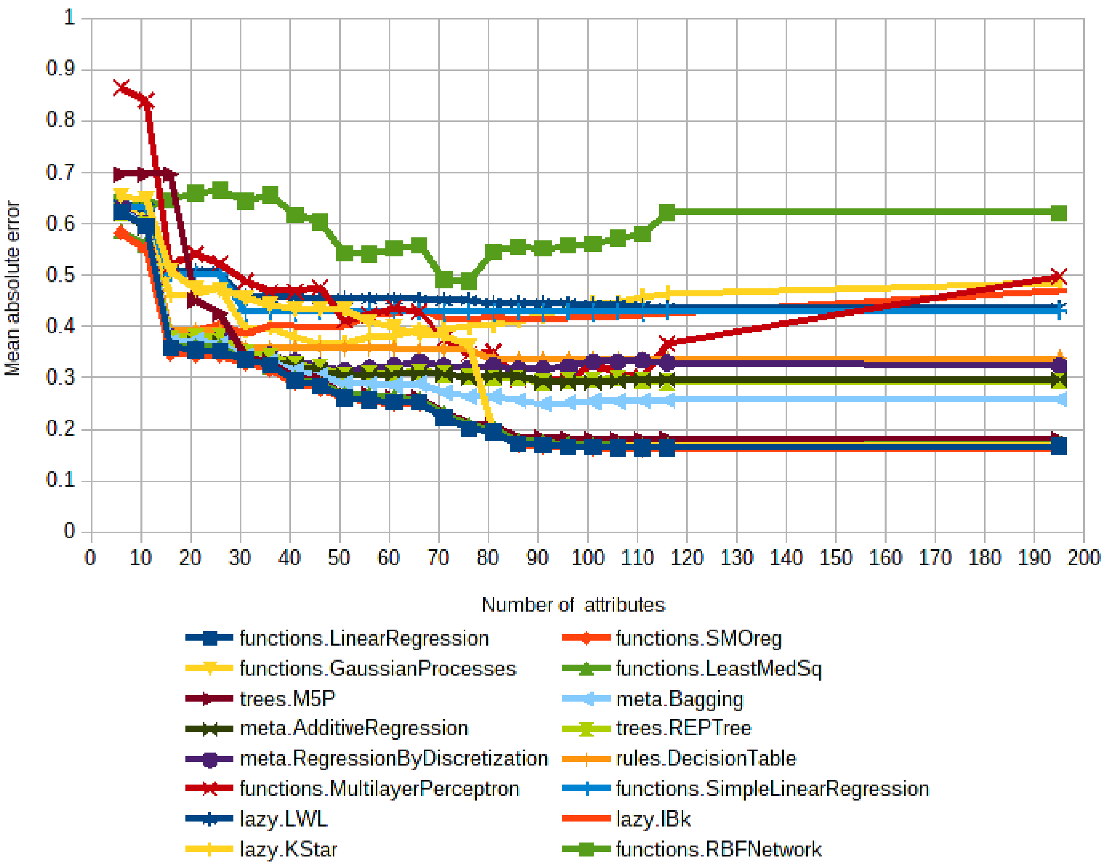

Figure 2.1: Global Ranked Pruning (Attackers): Mean Absolute Error Vs Number of Attributes

2.4.2 Discussions on Results from Global Ranked Pruning

We observe that the best classification performance for attacker is obtained using *SMOreg* algorithm with a mean absolute error of 0.1619 at 111 attributes, *Linear Regression*, *Gaussian Processes* and *LeastMedSq* algorithms closely follow the lead with mean absolute error around 0.165 at 111 attributes. The best performing algorithms belong to the function based family of algorithms (Figure 2.2), however, *Multilayer Perceptron* and *Simple Linear Regression* algorithms are exceptions as they do not perform well. The reason being that *Simple Linear Regression* oversimplifies the model and uses only the "goals" attribute to build model, however, it is reasonable to say that the expert ratings do not depend on just the number of goals scored by the player. As for the *Multilayer Perceptron*, the mean absolute error decreases as we go on decreasing the number of attributes until there are 86 attributes, at this point *Multilayer Perceptron* gives it best result with mean absolute error at 0.29. Further decreasing the number of attributes increases the error. When we look across all the algorithms in our set (Figure 2.1), we observe that the point around 86 attributes

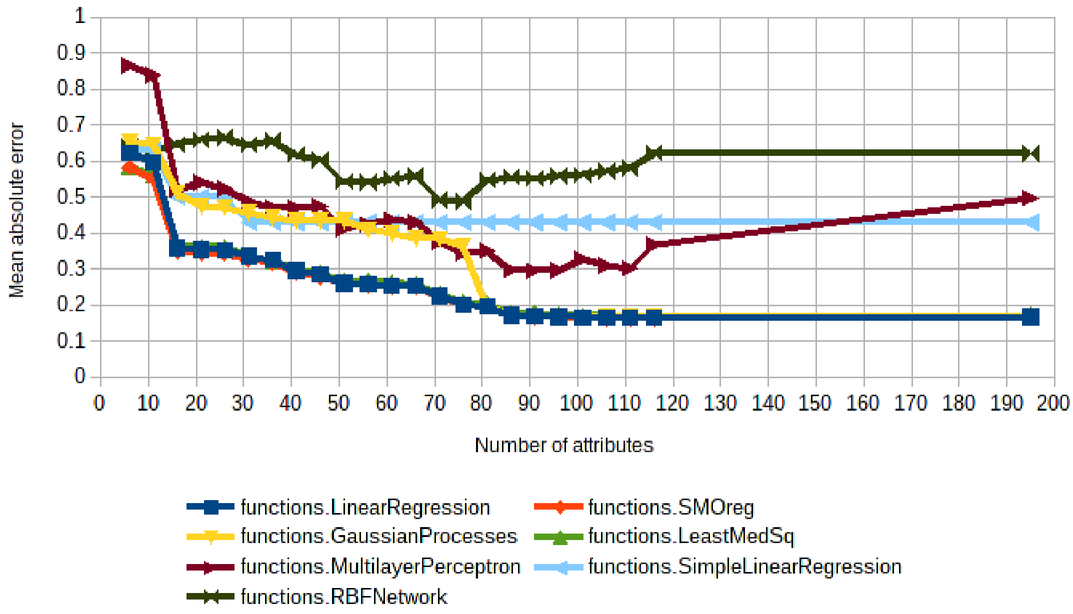

FIGURE 2.2: Global Ranked Pruning (Attackers): Function based algorithms -Mean Absolute Error Vs Number of Attributes

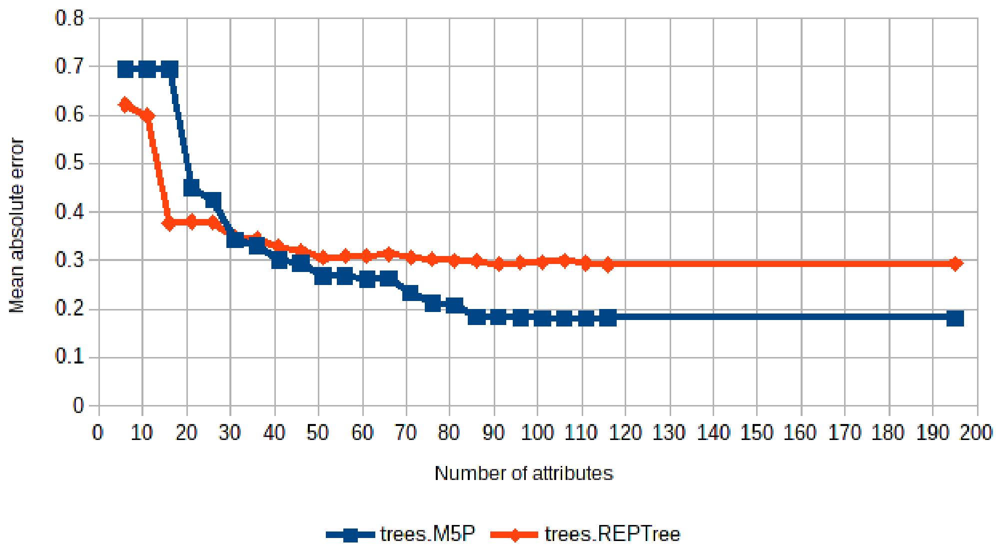

FIGURE 2.3: Global Ranked Pruning (Attackers): Tree based algorithms -Mean Absolute Error Vs Number of Attributes

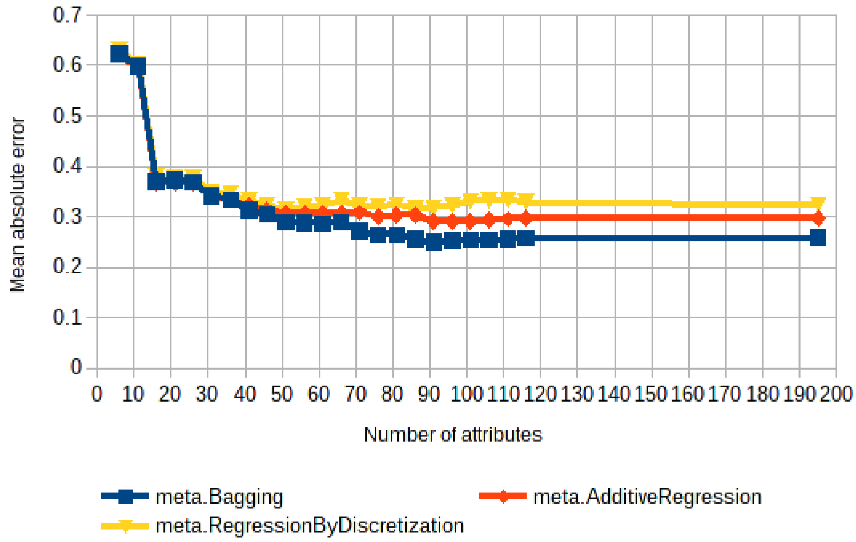

Figure 2.4: Global Ranked Pruning (Attackers): Meta algorithms -Mean Absolute Error Vs Number of Attributes

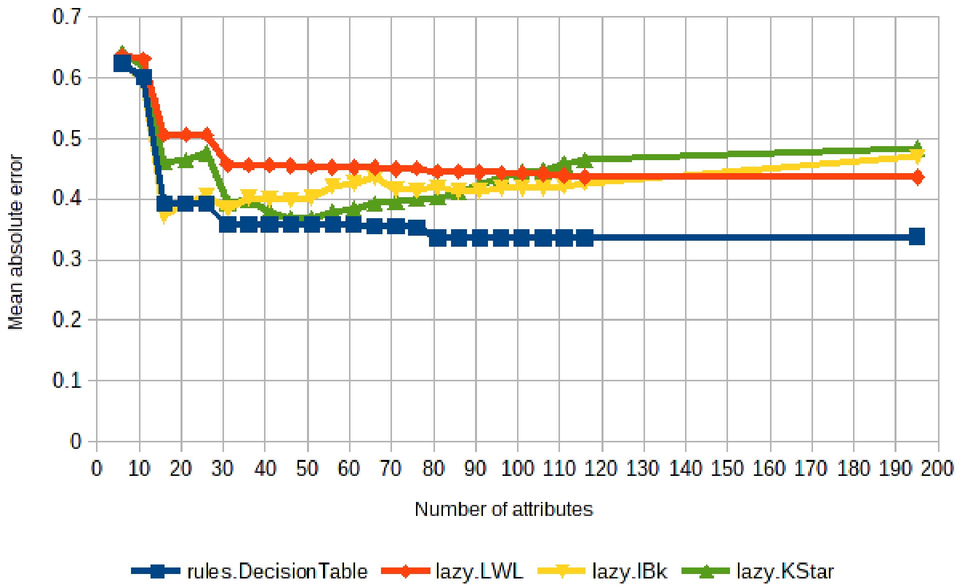

Figure 2.5: Global Ranked Pruning (Attackers): Lazy and Rule based algorithms -Mean Absolute Error Vs Number of Attributes

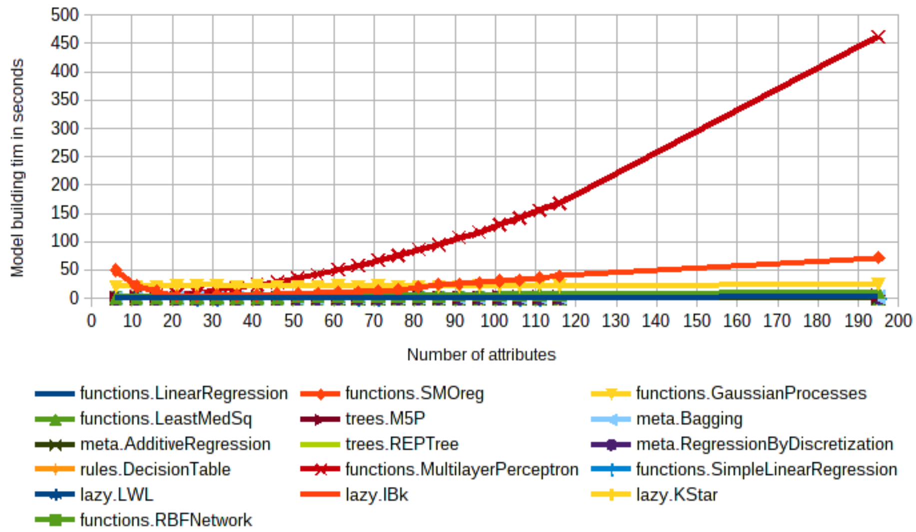

Figure 2.6: Global Ranked Pruning (Attackers): Model Building Time(sec) Vs Number of Attributes

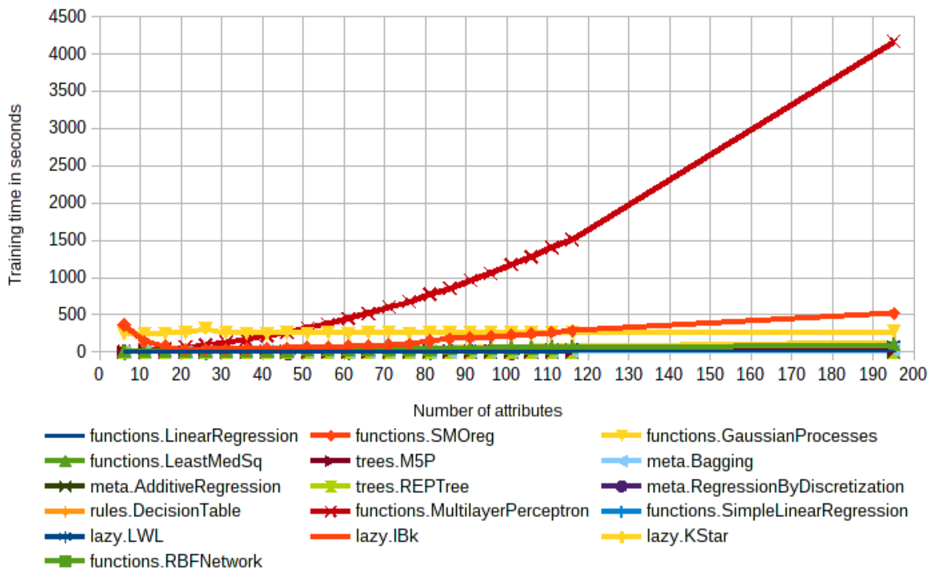

Figure 2.7: Global Ranked Pruning (Attackers): Model Training Time(sec) Vs Number of Attributes

sees a general dip in error for almost all attributes. The *SMOreg* algorithm at 86 attributes has mean absolute error of 0.1706, which is sufficiently good even when compared to the best performance(SMOreg at 111 attributes with MAE of 0.1619). We also observe that the lazy (instance based) algorithms have higher mean absolute error compared to other algorithms (Figure 2.5). *IBk*, a nearest neighbour classifier, *KStar* which uses similarity function and *Locally Weighted Learning* are all instance based algorithms. The higher error using these algorithms can be attributed to the nature of this classification task. If two players have good expert ratings, this does not necessarily imply that both of the player have similar valued performance metrics, one of them can be good in some attribute while the other can be good in some other. There can be several different attributes considered for higher rating. Hence instance based algorithms perform poorly. When we look at the mean absolute error of these three instance based algorithms we observe that *LWL* has no change, *IBk* has a little change and with *KStar* the mean absolute error goes down to 0.37 at 46 attributes. The reason for this different behaviour for KStar can be associated to the distance function it uses which is entropy based unlike other instance based algorithms. It makes sense that the *RBF Network* also gives poor result because it uses k-means to create the basis functions. Also the rule based algorithms like *Decision Table* do not give good results (Figure 2.5). If we have look at the tree based algorithms (Figure 2.3) we observe that both *REP Tree* and *M5P* algorithm cannot improve upon their classification performance with attribute pruning. However, *M5P* performs significantly better than *REP Tree* for more than 80 attributes. *M5P* algorithm constructs model trees which have linear regression function at their leaf nodes, while *REP Tree* is a fast decision tree learner using information gain/variance. Another major difference is their pruning strategy, *M5P* prunes by replacing subtrees with linear regression functions [Frank et al., July 1998] while *REP Tree* prunes using using reduced-error pruning (with backfitting) [Hall et al., 2009]. So the better performance of *M5P* can be because of the regression functions at the leaves, we have already observed that regression based algorithms generally perform better for current classification task which has all the attributes and the target class as numeric type. Next, we observe the meta algorithms (Figure 2.4) where we used ensemble method with tree based algorithms. We used *Bagging with REP Tree, Additive Regression with Decision Stump* and *Regression by Discretization with J48*. It is found that the performance of the tree based algorithms improve when they are combined with ensemble methods. For example there is reduction of around 0.04 in the mean absolute error of *REP Tree* when used with bagging. The meta algorithms do not exhibit any performance improvement with pruning of attributes.

Similarly, when we look at the results with defenders, goalkeepers and midfielders dataset, we observe same behaviour of the algorithms as seen in case of attackers. However, there are differences between the player datasets with respect to the optimal set of attributes for classification. This is an expected observation since all the four types of players have different roles to play and they ought to be evaluated on different sets of metrics. With attacker dataset, we got reasonably good results with a set of 86 attributes. In case of defenders we observe that the least mean absolute error(=0.2673) is with *SMOreg* algorithm using 94 attributes and we have mean absolute error of 0.2856 with 69 attributes using the same algorithm. Also, *Multilayer Perceptron* which performs poorly with higher number of attributes, gives its best performance at 69 attributes. We will consider this set of 69 attributes as optimal for defenders using current pruning strategy. Similarly for midfielders we see a general dip in mean absolute error across almost all algorithms including *Multilayer Perceptron* at 91 attributes, this is the optimal set for midfielders. For goalkeepers this optimal set consists of 37 attributes. The best results for all the four player types are indicated in Table 2.6 and the results at optimal number of attributes are shown in Table 2.7). We see that the increase in mean absolute error from "best results irrespective of number of attributes' to "best results with optimal number of attributes' is minimal.

When we consider the model building time (Table 2.6) and training time for the models (Figure 2.7) using different algorithms, we see that *Multilayer Perceptron* take relatively very large amount of time with higher number of attributes. For example for defenders with 195 attributes it takes 16 minutes for building model and 2.5 hours for training the model. But as we go on decreasing the number of attributes the building and training time both decreases linearly with a slope of around 36 degrees. The other algorithms taking significant building and training times are *SMOreg* and *Gaussian Processes*. With subsequent attribute pruning the time taken by *SMOreg* decreases while it remains constant in case of *Gaussian Processes*. The instance based algorithms are the quickest in model building and training. Similar trend in model building and training time can be seen across all the four player datasets.

From the Tables 2.6 and 2.7 we can also observe that the least mean square error for defenders is highest amongst the four player types. This means that it is most difficult to predict the expert rating for defenders using the given set of performance attributes. We can also say that it is therefore more difficult to say which of the attributes in our dataset determines a better defender according to the expert ratings. Next, we also see that the optimal number of attributes for midfielder are the highest (91), hence we can say that the expert ratings for midfielder take into account more performance metrics than other player types. So a better midfielder is better in more dimensions than other players, say goalkeeper who needs to take care of only 37 performance metrics to get a good expert rating. However, it is apparent since midfielders can have different roles, e.g., centre midfielder, defensive midfielder, attacking midfielder, wide midfielder, etc., whereas all goalkeepers have the same role.

TABLE 2.6: Best Results irrespective of number of attributes with Global Ranked Pruning

Best Performance	**Attackers**	**Defenders**	**Gks**	**Midfielders**
Mean Abs Error	0.1619	0.2674	0.2070	0.1708
Algorithm	SMOreg	SMOreg	SMOreg	SMOreg
Num of attributes	111	94	62	126

Table 2.7: Results with optimal number of attributes with Global Ranked Pruning

Best Performance	Attackers	Defenders	Gks	Midfielders
Mean Abs Error	0.1706	0.2856	0.2160	0.1763
Algorithm	SMOreg	SMOreg	SMOreg	SMOreg
Num of attributes	86	69	37	91

2.4.3 Iterative Local Pruning

Now we apply a different strategy for pruning the attributes. For each of player datasets with 195 attributes initially, we execute the *Linear Regression, SMOreg, Gaussian Processes, LeastMedSq, M5P, Bagging with REP Tree, Additive Regression with Decision Stump, REP Tree, Regression by Discretization using J48, Decision Table, Multilayer perceptron, Simple Linear Regression, Locally Weighted Learning, IBk, KStar and RBF Network* algorithms. Then we select those algorithms, which generates such classification models from which we can extract a list of attributes. This is usually possible only in case of interpretable models. We select four algorithms for Iterative Local Pruning, *Linear Regression, LeastMedSq, M5P and REP Tree.* The first two algorithms are function based while the other two are tree based. When we had executed all the different algorithms in our set we got results as show for attackers in Table 2.1. From this table, we noted that function based algorithms like *Linear Regression* and *LeastMedSq* and tree based algorithm *M5P* gives good classification results with our datasets. Also it is possible to obtain a ranked list of attributes from the models generated by *Linear Regression*, *LeastMedSq*, *M5P* and *REP Tree*. Note that for *REP Tree* the model generates a tree from which we extract attributes at the nodes and the leaves of the tree and remove the duplicate attributes to obtain the required list of attributes. In this pruning strategy, for all the four player datasets, and for each of the four algorithms, we execute the algorithm on the dataset using the initial 195 attributes, we extract the list of attributes (say List A) used in model creation by the algorithm in current iteration, then we create a new dataset by selecting only those attributes present in the extracted list (List A). Then we execute the same algorithm on the new dataset, extract the list of attributes used in model creation and use these attributes to create the next dataset and so on. Here the dataset used for an execution of the algorithm depends on the attributes used by the algorithm in model building in previous iteration. We stop the pruning at that iteration of the algorithm where the set of attributes used in model creation is same as the set of attributes used in model creation in previous iteration, at this point the pruning saturates. This pruning strategy is called “Iterative Local Pruning” because here the pruning is done iteratively and is dependent on the previous iteration’s model only (local) and is done for each algorithm separately, in contrast to the “Global Ranked Pruning” where for a particular player dataset, the pruning was done based on the ranked list of attributes generated by the best performing algorithm in the first execution of the algorithm.

For attackers dataset, the result summary for the *Linear Regression* algorithm is shown in Table 2.8, result summary for *LeastMedSq* algorithm is shown in Table 2.9, for *M5P* algorithm is shown in Table 2.10 and for *REP Tree* algorithm is shown in Table 2.11. (The results for other three datasets can be found in Appendix: C.)

TABLE 2.8: Iterative Local Pruning (Attackers): Linear Regression

NumOfAttrib	CC	MAE	RMSE	RelAE	BTime(s)	TTime(s)
195	0.9646	0.1679	0.2379	24.1348	7.4490	60.7150
117	0.9664	0.1644	0.2320	23.6302	0.1610	4.3590

TABLE 2.9: Iterative Local Pruning (Attackers): LeastMedSq

NumOfAttrib	CC	MAE	RMSE	RelAE	BTime(s)	TTime(s)
195	0.9639	0.1724	0.2489	24.7807	18.6720	159.8560
175	0.9639	0.1724	0.2489	24.7807	16.2680	147.0290

TABLE 2.10: Iterative Local Pruning (Attackers): M5P

NumOfAttrib	CC	MAE	RMSE	RelAE	BTime(s)	TTime(s)
195	0.9582	0.1806	0.2582	25.9657	7.7110	49.8840
83	0.9566	0.1826	0.2630	26.2458	2.7460	24.6970
76	0.9577	0.1804	0.2596	25.9324	2.3780	21.7070
64	0.9525	0.1872	0.2747	26.9047	1.7050	16.7360

TABLE 2.11: Iterative Local Pruning (Attackers): REP Tree

NumOfAttrib	CC	MAE	RMSE	RelAE	BTime(s)	TTime(s)
195	0.8921	0.2934	0.4080	42.1705	1.0360	3.9550
43	0.8911	0.2928	0.4097	42.0959	0.0940	0.7550

2.4.4 Discussions on Results from Iterative Local Pruning

We observe that the pruning saturates very quickly. In case of *LeastMedSq* and *REP Tree* the pruning saturates in after two iterations, also for *Linear Regression* the pruning saturates after two or three iterations. *M5P* algorithm takes three or four iterations to saturate. The mean absolute errors at the saturation iteration of the algorithms are shown in Table 2.12 with the corresponding number of attributes for the four player types. When we compare these results to the result obtained with Global Ranked Pruning (Figure 2.2), we find that *Linear Regression* algorithm with Iterative Local Pruning strategy, saturates at that iteration where the mean absolute error is minimum. The set of attributes at this saturation point gives us the set of attributes which would give best possible performance with *Linear Regression*, for example, *Linear Regression* on attackers dataset will give its best performance with 116 attributes (further pruning with Global Ranked Pruning increases the error).

Another aspect to be noted here is that, even though this saturation point is the best performance point for *Linear Regression*, the optimal set of attributes obtained with Global Ranked Pruning is still good enough. For example on attackers-dataset with Global Ranked Pruning, the optimal point is 86 attributes with mean absolute error of 0.1706 while the Iterative Local Pruning with *Linear Regression* saturates at 116 attributes with mean absolute error of 0.1644. The reduction in the number of attributes is of greater significance than the increase in error. Similar arguments can be made for defenders, goalkeepers and midfielders datasets (compare Table 2.12 and Table 2.7).

When *M5P* is used with Iterative Local Pruning, the error might increase or decrease with pruning (unlike *Linear Regression* where Iterative Local Pruning always results in reduction of error). When *M5P* is applied to goalkeepers dataset it saturates at 34 attributes. It is interesting to note here that Global Ranked Pruning with 35 attributes attributes had higher mean absolute error (Figure C.13. For *LeastMedSq* we observe no change in mean absolute error with Iterative Ranked Pruning, and it saturates with highest number of attributes in its set. For *REP Tree* algorithm, the mean absolute error decreased in cases of attackers, midfielders and goalkeepers, however, the error increased in case of defenders. This could again point to the fact that it is harder to characterise good performance metrics for expert ratings for defenders in comparison to other three player positions. This was also indicated in Global Ranked Pruning when we find that the least mean absolute error for defenders are highest amongst the four player positions (Table 2.7). Another interesting observation with *REP Tree* here is that for attackers and defenders dataset, the error at the saturation point in case of Iterative Local Pruning is less than the error with the same number of attributes with Global Ranked Pruning. So even though the performance of *REP Tree* at saturation is not optimal, its selection of 42 attributes for attackers and defenders is better than the selection using Global Ranked Pruning with *REP Tree* of 42 attributes of respective datasets. (Figure 2.3, Figure C.3).

2.4.5 Threshold Pruning

This pruning strategy is similar to *Iterative Local Pruning* except that we use a minimum threshold weight to select the attributes. We implement this pruning strategy with *Linear Regression* algorithm. At first, we execute the *Linear Regression* algorithm with 195 attributes, then we extract the list of attributes used in the generation of the model with their weights. Out of this list we discard all those attributes which are below our threshold weight. We use the remaining set of attributes for the next iteration of the algorithm. We keep on repeating the process until we reach a saturation point when the number of selected attributes becomes constant after the current iteration of the algorithm. We perform this activity for all the four player datasets. We tried several different threshold value settings for our experiments and realised that best results were obtained with threshold as (0.3 X standard deviation of attribute weights) or threshold as (mean of attribute weights - standard deviation of attribute weights).

TABLE 2.12: Iterative Local Pruning: Saturation of Pruning

	Attackers	Defenders	GKs	Midfielders
Linear Regression **NumOfAttrib** **MAE**	 116 0.1644	 94 0.2700	 61 0.2071	 126 0.1722
M5P **NumOfAttrib** **MAE**	 63 0.1872	 60 0.3086	 34 0.2231	 95 0.1900
LeastMedSq **NumOfAttrib** **MAE**	 174 0.1724	 173 0.2766	 108 0.2164	 177 0.1764
REP Tree **NumOfAttrib** **MAE**	 42 0.2928	 42 0.3939	 25 0.3730	 74 0.3232

2.4.6 Discussions on Results from Threshold Pruning

The results for Threshold Pruning with *Linear Regression* using the two different kinds of threshold "difference of *mean of attribute weights* and the *standard deviation of attributes weights*" and "0.3 times the standard deviation of attribute weights" are show in Tables 2.13, 2.14 for attackers. (Results for other three positions can be found in Appendix: C).

TABLE 2.13: Threshold Pruning (Attackers): Linear Regression (Threshold = mean -SD)

NumOfAttrib	CC	MAE	RMSE	RelAE	BTime(s)	TTime(s)	Thresh
195	0.9646	0.1679	0.2379	24.1348	7.5490	63.0340	0.0587
39	0.9113	0.2689	0.3715	38.6553	0.0650	0.7590	0.0156
29	0.9103	0.2695	0.3735	38.7333	0.0220	0.2880	0.0381
27	0.9104	0.2693	0.3732	38.7098	0.0180	0.2340	0.0674
25	0.9106	0.2691	0.3728	38.6872	0.0130	0.1690	0.0871
25	0.9106	0.2691	0.3728	38.6872	0.0330	0.1610	0.0871

The Table 2.15 represents the best saturation of threshold pruning using *Linear Regression* over all the four player types. For attackers, both the threshold values reached same saturation value, which means more agreement and hence good set of attributes. For defenders and midfielder, the threshold (difference of mean and SD) results in saturation point with very low number attributes with comparatively worse performance than threshold at 0.3 times SD, hence we take the values obtained using 0.3 times SD as threshold as optimal. In case of goalkeepers, threshold at *difference*

Table 2.14: Threshold Pruning (Attackers): Linear Regression (Threshold = 0.3 x SD)

NumOfAttrib	CC	MAE	RMSE	RelAE	BTime(s)	TTime(s)	Thresh
195	0.9646	0.1679	0.2379	24.1348	7.3920	58.6910	0.0642
37	0.9113	0.2688	0.3715	38.6448	0.0490	0.6220	0.0778
26	0.9106	0.2691	0.3728	38.6872	0.0190	0.2760	0.0775
25	0.9106	0.2691	0.3728	38.6872	0.0120	0.1680	0.0775
25	0.9106	0.2691	0.3728	38.6872	0.0140	0.2030	0.0775

of mean and SD, reaches a lower error at saturation and hence is considered to be better than the other threshold. The results are summarised in Table 2.15 .

Table 2.15: Threshold Pruning Optimal Results

Linear Regression	Attackers	Defenders	GKs	Midfielders
NumOfAttrib	25	14	6	13
MAE	0.2691	0.4427	0.5718	0.4240

2.5 Conclusion

We have used three different pruning strategies, Global Ranked Pruning, Iterative Local Pruning and Threshold Pruning. We obtained optimal lists of attributes for each player position from each of the three different strategies. We compare the lists to find out the similarity between the different optimal attribute sets for each of the four player positions. For measuring the similarity between two ranked lists of attributes, we calculate the percentage similarity between the lists at top 5 attributes, then the percentage similarity at top 10 attributes, and so on until the shorter list of attributes is exhausted. We calculate the mean of all these similarities to obtain a final measure of similarity. Using this approach, the similarity at higher ranks in the lists are given more importance than similarity at the lower end of the list. The similarity between the different optimal attribute sets for attackers is shown in Figure 2.8. In this graph each of the selected attribute lists is compared with the other lists. To avoid comparison of the same two lists twice, each list on the axis is compared to only the lists with whom it has not already been compared. Thus the increasing number of bars along the X-axis. (For the other similarity results for the lists for other three player positions, see Appendix: C.) We observe that the similarity between the lists generated using function based algorithms are the highest.

Now to find the final list of attributes for each player position which best characterises the expert ratings, we combine the lists obtained from three different strategies. This is a rank aggregation problem. We solve this problem in the following way. For each of the player type, we create a list of all attributes presents in all the lists mentioned in Table 2.16. Next, we create a reward and penalty strategy for ranking these attributes. For each attribute we note its occurrence in all the optimal lists, for each occurrence in a list at position n from the top, we assign we weight n to the attribute, this is reward. Hence the reward is the sum of its rank from the optimal lists in which that attribute is present. Then we calculate the penalty for the attribute, the penalty for an attribute is the product of the "average size of optimal attribute lists for that player type" and the "percentage of attributes lists for that player type in which that attribute is absent". In this way we obtain the final optimal list of attributes for each player type. The top 25 attributes for attacker-list is shown in Table 2.17, for defenders is shown in Table 2.18, for goalkeepers is shown in Table 2.19 and for midfielders is shown in Table 2.20.

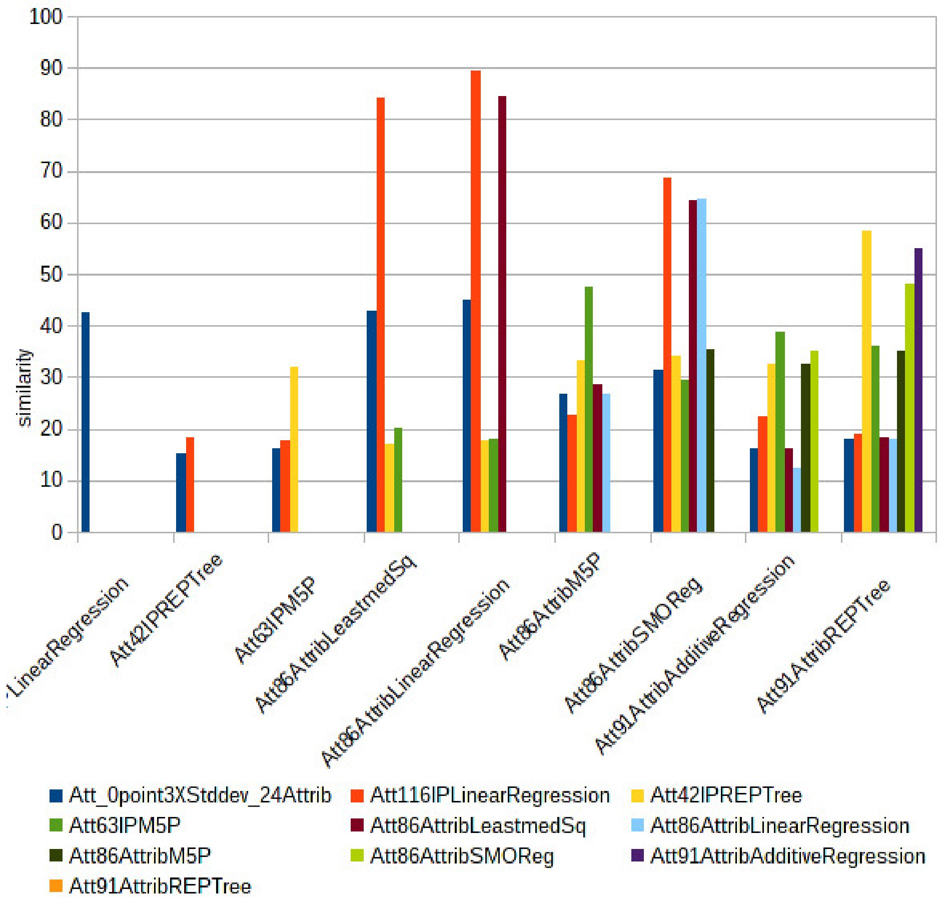

FIGURE 2.8: Similarity between optimal lists for attackers (Legend for list names 2.16)

Table 2.16: Legend for list of optimal attackers attribute lists (2.8)

Name of Attacker List	**Algorithm used**	**NumOfAtt**	**Pruning**
Att_0point3XStddev_24Attrib	Linear Regression	24	Threshold
Att116IPLinearRegression	Linear Regression	116	Iterative Local
Att42IPREPTree	REPTree	42	Iterative Local
Att63IPM5P	M5P	63	Iterative Local
Att86AttribLeastmedSq	LeastmedSq	86	Global Ranked
Att86AttribLinearRegression	Linear Regression	86	Global Ranked
Att86AttribM5P	M5P	86	Global Ranked
Att86AttribSMOReg	SMOreg	86	Global Ranked
Att91AttribAdditiveRegression	AdditiveRegression	91	Global Ranked
Att91AttribREPTree	REPTree	91	Global Ranked

Table 2.17: Top 25 attributes from final Optimal List of attributes for attackers

Assists
Last Man Tackle
Red Cards
Challenge Lost
Goals from Throws
Left Foot Blocked Shots
Attempts from Corners on target
Team Formation
Tackles Lost
Own Goal
Substitute Off
Goals from Corners
Interceptions
Blocked Shots from Inside Box
Other Clearances
Successful Crosses Corners in
Unsuccessful Dribbles
Unsuccessful Lay-Offs
Foul Won Penalty
Total Fouls Won
Unsuccessful Flick-Ons
Penalties Conceded
Touches open play opp six yard
Goals from Outside Box
Goals Open Play

TABLE 2.18: Top 25 attributes from final Optimal List of attributes for defenders

Error leading to Goal
Assists
Touches
Shots Cleared off Line Outside
Headed Shots On Target
Successful Ball Touch
Pass Left
Pass Right
Unsuccessful Long Passes
Attempts Open Play on target
Blocked Shots
Unsuccessful Passes Defensive
Unsuccessful Passes Middle thirf
Unsuccessful Passes Opposition
Clearances Off the Line
Goals
Shots Off Target inc woodwork
Tackles Lost
Left Foot Goals
Own Goal
Successful Passes Own Half
Unsuccessful Short Passes
Pass Backward
Penalty Goals
Goals from Outside Box

Table 2.19: Top 25 attributes from final Optimal List of attributes for goalkeepers

Red Cards
Own Goal
Successful Passes Middle third
Crosses not Claimed
Successful Short Passes
Fouls Won in Danger Area including penalties
Foul Won Penalty
Team Formation
Total Successful Passes All
Unsuccessful Passes Middle third
Unsuccessful Passes Final third
Saves from Penalty
Fouls Won not in danger area
Duels lost
Ground Duels won
Aerial Duels won
Fouls Conceded excluding handballs penalties
Saves Made from Outside Box
Dispossessed
Successful Long Passes
Successful Passes Own Half
Shots On Conceded
Shots On Conceded Outside Box
Last Man Tackle
Position in Formation

TABLE 2.20: Top 25 attributes from final Optimal List of attributes for midfielders

Error leading to Goal
Goals
Red Cards
Fouls Won not in danger area
Goal Assist Corner
Assists
Right Foot Blocked Shots
Left Foot Blocked Shots
Total Fouls Won
Goals from Throws
First Goal
Goals from Inside Box
Goals from Outside Box
Successful Lay-Offs
Goals from Set Play
Goals Open Play
Left Foot Goals
Goals from Corners
Headed Goals
Blocked Shots
Other Goals
Penalties Conceded
Clearances Off the Line
Key Set Pieces
Team Formation

Chapter 3

Characterising Match Outcome with Team Performance Metrics

In this chapter, our objective *to identify the most important performance attributes of the players which affect the match outcome.* We will further *investigate the extent to which these attributes of the players of the two competing teams can characterise the match outcome.* We will prepare the datasets in suitable formats and perform several classification tasks. We will use attribute selection to identify the most important performance attributes for characterising match outcome.

3.1 Preparing the data

The dataset of player performances for EPL released by OPTA contained 210 attributes and 10369 instances. Out of the 210 attributes of players, 198 attributes were performance statistic while the others were identifiers for the player for that match. The dataset released by OPTA did not contain match-outcomes. The dataset did not contain *Own Goals*, the goals scored by a player against his own team. The data for *Own Goals* scored by each player in each match of the tournament was fetched from WhoScored.com [1] and this attribute was added to the dataset. Next, we needed to obtain data for each match, hence the dataset for player performances was modified and condensed to generate data for each match played by every team. Then the dataset was appropriately modified to contain only 380 instances, one for each of the match. As we generated the match dataset from player dataset we had aggregate the performance metrics of the individual player of the team for that match. Some of the performance metrics were averaged while most of them were summed based on the understanding of the influence of these attributes on the team performance. In the newly generated match dataset, each instance represented a match and contained the aggregate of performance metrics of home team as well as the away team for that match. Hence the number of attributes in the dataset doubled.

[1] http://www.whoscored.com/

Next, we analysed the data for dimensionality reduction. The attributes which were to found to be redundant or constant for all players were removed from the datasets. Then some match specific attributes were added to the dataset and we arrived at 368 attributes for each match instance. Then we create another data table which contains the summary of the performances of the teams over the complete season. Hence this dataset has 20 instances, one for each team. We compare the number of matches won, matches lost and total goals scored for each team as shown in this dataset to the values available on the sports website. In this way we corroborate the correctness and reliability of our data. Now we have obtained appropriate and correct datasets at player level and at match level which we will use for our experiments.

3.2 Methodology

In this supervised classification task, we have to predict the target class which is *match outcome* using the performance attributes of the two teams in that match. A match outcome can be 'Home Team Win', 'Home Team Loss' or 'Draw'. We use the prepared match dataset. We remove all the attributes which directly influence the match outcome, usually the attributes giving direct information about the goals scored like sum of goals scored by home/away team from corners, sum of goals scored from throws, sum of goals scored from penalties, sum of goals assist corners, sum of goal assist free kicks, sum of goal assist throw-ins, sum of clean sheets of home/away team,etc. This is done to avoid generating trivial rules for characterising the match-outcome. The goals scored and conceded can on its own determine the match outcome, but we remove the attributes correlated to these so that we find out which other performance attributes of teams affect the match-outcome. This also means that the number of attributes of the dataset decreases further and we have 336 attributes now. Next, we will execute several machine learning algorithms on this matchdataset, and do further attribute selection to see how it affect the target class, i.e., match outcome prediction

3.3 Results and Discussions

We execute a set of machine learning algorithms like *MultiLayer Perceptron, Functional Trees, Sequential Minimal Optimization, Naive Bayes, Random Forest, Decision Table, Fuzzy Unordered Rule Induction Algorithm, J48Graft, J48, JRip, REP Tree, LibSVM, Kstar, AdaBoostM1 with Functional Tree.* The WEKA toolkit was used for these implementations. We executed the algorithms with different parameter values and it was observed that the default parameter values of the WEKA toolkit were giving the best performance in all cases except the ensemble based algorithms like *Bagging* and *AdaBoostM1*. In the case of ensemble based algorithms the best performance was observed when used with best performing non-ensemble algorithm. Hence bagging and boosting algorithms were not advantageous over other algorithms. For evaluation of performance of algorithms we used 10-fold cross-validation. The output is summarised in (figure 3.1). We observe that *MultiLayer Perceptron, Functional Trees, Sequential Minimal Optimization* are the top performing algorithms according to percentage of correctly classified instances, ROC areas, F-Measure and Kappa statistic. We observe that we get almost 69.5% correctly classified instances with *Multilayer Perceptron* with learning rate = 0.3. This indicates that the data we are using for the classification task is useful and informative about the match outcome.

Then we use attribute selection based on gain ratio to decrease the number of attributes. A minimum threshold is set and attributes with gain ratio below that threshold are not used in the classification task. The initial threshold is set to be 0.0277 which includes all attributes having positive gain ratio. We get 59 attributes in the set. Then we change the threshold to 0.05 resulting in 48 attributes. Next, the threshold is 0.07 and includes only 34 attributes. We observe that increasing the threshold more than 0.07 decreases the performance of algorithms drastically. The results are for threshold value of 0.07 is shown in Table C.25 (for the other two threshold results, see Appendix: C) We observe that instance based algorithms like *KStar* are worst performers, tree based algorithms using information gain also perform poorly while tree based algorithm having logistic regression function at nodes/leaves performs comparatively better. Also as we go on decreasing the number of attributes by setting gain-ratio threshold the performance of some algorithms change (see fig 3.2, 3.3, 3.4 and 3.5). We observe that the performance of *Multilayer perceptron* decreases as we go on decreasing the number of attributes, but *Sequential Minimal Optimization* algorithm remains maintains its good prediction. The performance of *KStar* increases as we go on decreasing the number of attributes. So it means that the instance based algorithms are worst affected by the curse of dimensionality. *KStar* algorithm works better with fewer and more useful attributes.

In Figure 3.1 we can see the percentage of correctly classified instances for all algorithms for all the threshold levels. *Functional Tree and Sequential Minimal Optimization* algorithm give us reasonably good prediction even with threshold value of 0.07, using only 34 attributes. It can be concluded from this that the first top 34 attributes ranked by gain-ratio have reasonably sufficient information to characterise the match outcome. These top 34 attributes are shown in Table 3.3.

3.4 Conclusion

The objective of this experiment was to find out which performance attributes of the players of the two competing teams affect the match outcome and to what extent is the match outcome characterised by the performance attributes of players. In response to that we can say that the attributes listed in Table 3.3 affect the match outcome most. For a more comprehensive list we can say that the 59 attributes obtained with positive gain ratio are the ones affecting match outcome. The rest 308 attributes have almost negligible information to characterise match outcome. For the second part of this objective, we needed to investigate the extent to which these attributes characterise match outcome. We have the best performance with 34 attributes as

63.4% correctly classified instances with *Sequential Minimal Optimization* algorithm. Also observing the performances of all algorithms we can conclude that the top 34 attributes listed in 3.3 characterise the match outcome to a satisfactory extent.

Table 3.1: Algorithms performance on matchdata (number of attributes=336).

Algorithm	**% Correctly classified**	**ROC-Area**	**F-Measure**	**Kappa**
MultiLayer Perceptron	69.4737	0.836	0.693	0.5235
FT	67.8947	0.801	0.683	0.5045
AdaboostM1 with FT	67.6316	0.827	0.672	0.4939
SMO	65.2632	0.754	0.648	0.4529
NaiveBayes	56.5789	0.746	0.571	0.3384
RandomForest	56.8421	0.74	0.552	0.3094
DecisionTable	57.6316	0.724	0.554	0.3196
FURIA	56.8421	0.715	0.551	0.3162
J48Graft	53.9474	0.637	0.527	0.275
J48	52.6316	0.631	0.522	0.2619
Jrip	52.1053	0.622	0.48	0.2174
REP Tree	47.1053	0.609	0.438	0.1342
LibSVM	45	0.5	0.279	0
Kstar	45	0.5	0.279	0

Table 3.2: Algorithms performance with selected attributes (threshold=0.07, number of attributes=34) on matchdata

Algorithm	**% Correctly classified**	**ROC-Area**	**F-Measure**	**Kappa**
MultiLayer Perceptron	59.4737	0.783	0.593	0.3694
FT	62.1053	0.736	0.619	0.4098
SMO	63.4211	0.74	0.621	0.4206
NaiveBayes	54.7368	0.761	0.555	0.3154
RandomForest	56.3158	0.743	0.549	0.3064
DecisionTable	58.6842	0.747	0.566	0.3385
FURIA	56.8421	0.501	0.501	0.287
J48Graft	52.6316	0.635	0.522	0.2591
J48	52.8947	0.644	0.527	0.2659
Jrip	54.2105	0.636	0.493	0.2399
REP Tree	55	0.668	0.513	0.2708
LibSVM	45	0.5	0.279	0.3667
KStar	51.5789	0.7	0.512	0.2438

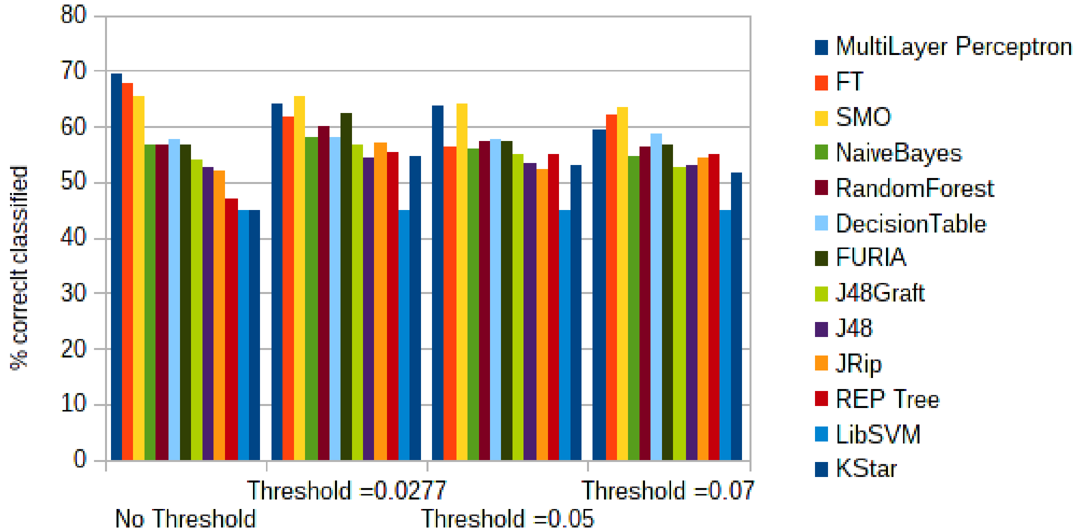

FIGURE 3.1: Algorithms comparison: Percentage of correctly classified instances Vs gain-ratio threshold

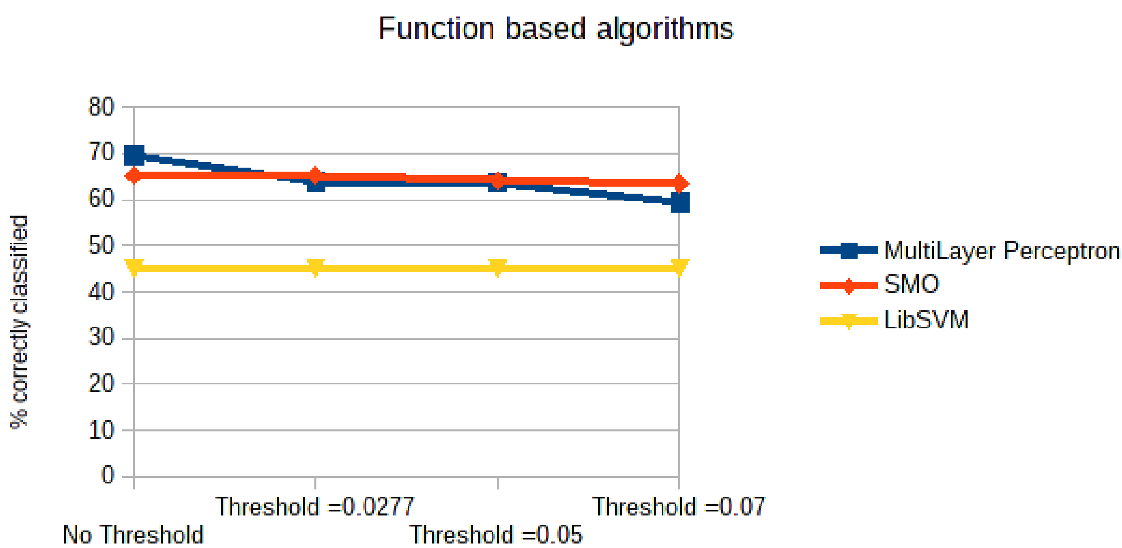

FIGURE 3.2: Function based algorithms: Percentage of correctly classified instances Vs gain-ratio threshold

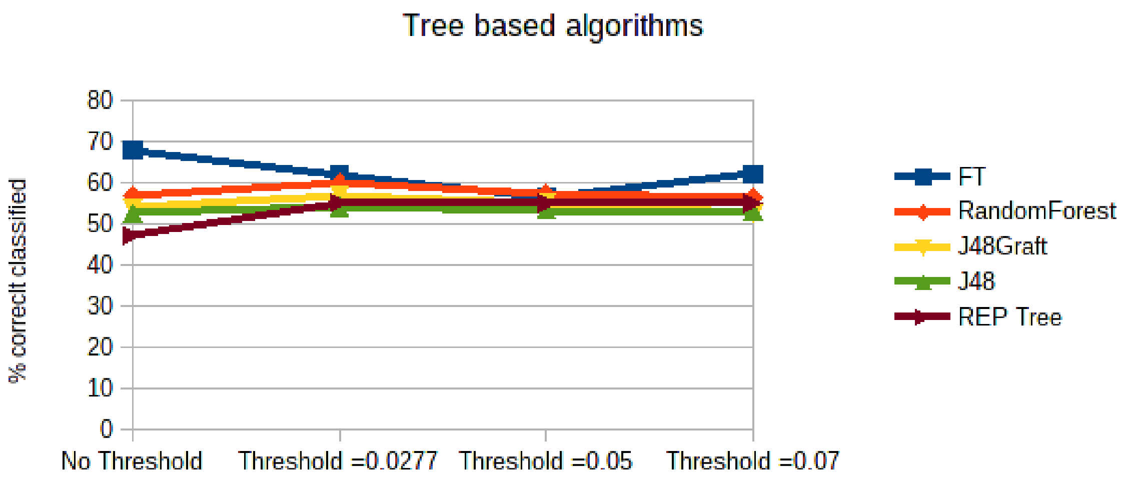

Figure 3.3: Tree based algorithms: Percentage of correctly classified instances Vs gain-ratio threshold

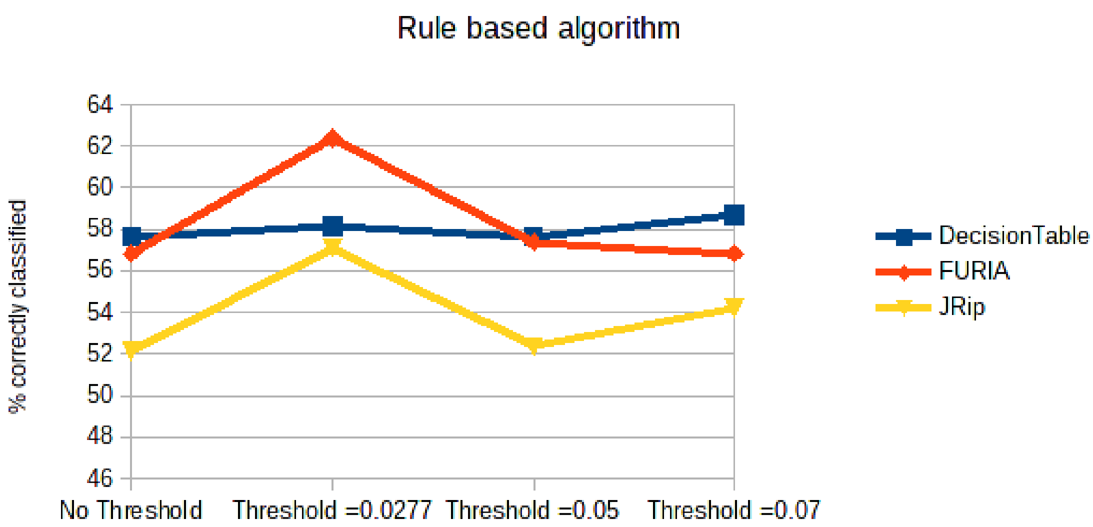

Figure 3.4: Rule based algorithms: Percentage of correctly classified instances Vs gain-ratio threshold

TABLE 3.3: Top 34 highest gain- ratio team attributes

Gain-Ratio	**Attribute**
0.2423	SA Through Ball
0.2252	SH Successful Crosses Corners in the air
0.1483	AvgH Time Played
0.1448	SH Successful Crosses Left
0.1282	SH Shots On Conceded Inside Box
0.1173	SA Shots On from Inside Box
0.1099	SH Successful Short Passes
0.1048	SH Big Chances Faced
0.1048	SA Big Chances
0.1048	SA Shots On Target inc goals
0.1043	SH Total Successful Passes Excluding Crosses Corners
0.1019	SH Touches
0.0981	SH Total Successful Passes All
0.095	SA Shots On Conceded
0.0947	SH Successful Passes Own Half
0.0927	SA Shots On Conceded Inside Box
0.0922	SH Successful Passes Middle third
0.0914	SH Successful Passes Opposition Half
0.0891	SH Right Foot Shots On Target
0.0888	SH Shots On Target inc goals
0.0879	SH Shots On from Inside Box
0.0879	SH Big Chances
0.0879	SA Big Chances Faced
0.087	SH Pass Backward
0.0868	SH Shots On Conceded
0.0839	SH Pass Left
0.0811	SA Touches open play opp box
0.0807	SH Total Unsuccessful Passes All
0.0802	SH Red Cards
0.0796	SH Unsuccessful Passes Final third
0.0769	SH Left Foot Shots On Target
0.0736	SA Total Clearances
0.0723	SA Headed Clearances
0.0719	SH Successful Passes Defensive third

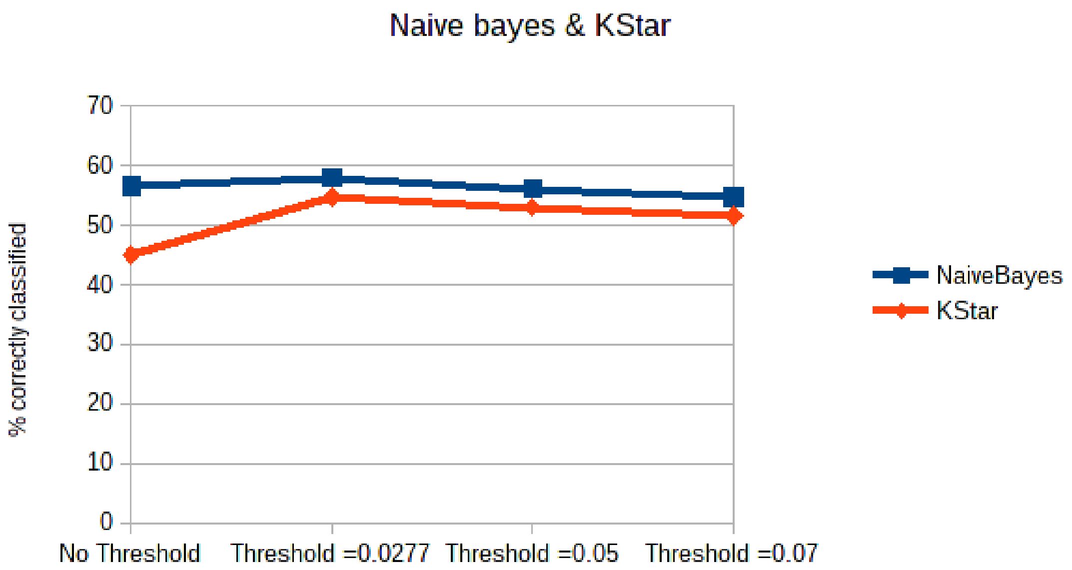

Figure 3.5: Naive bayes and KStar algorithms: Percentage of correctly classified instances Vs gain-ratio threshold

Chapter 4

Combining Player Ratings

In this chapter we will study how player expert ratings are correlated to match outcomes, and how predictive they are for future matches. For this we need to perform following tasks. Firstly, we have to find out methods to combine individual player ratings to form team ratings which are good representation of team performance. Next, we analyse how good is our ratings system for characterising the current match outcome. This can be done using either only the ratings for the current match or it could also include all the past performances of the players involved for classification. Then we use our ratings system to predict the next match outcome where we use only the past performance ratings of the players involved.

4.1 Objectives

The purpose of this experiment is to achieve the third and fourth objectives of our thesis (1.1). Therefore the two objectives for the current experiment are:

1. To find out aggregation methods by which individual player ratings can be aggregated resulting in a set of team ratings such that the team ratings are a good representation of the team performance. To evaluate the aggregation method, we use the team ratings generated as input for the classification task of predicting the match outcome.

   Additionally, if the team ratings are good at predicting the match outcome, it could mean either of the two hypotheses. First, it could mean that *the best performing team wins*, for this we have to assume that the ratings assigned by the experts are only dependent on performance and not on match outcome. This is shown in Figure 4.1 when the dependency line A is absent. However, we know that in soccer, it is not always true that the best performing team wins. Hence this hypothesis is weaker. The second hypothesis is that *the ratings by experts are influenced by match outcome*, shown in Figure 4.1 when the dependency line A is present. This case is highly likely. Hence, using this classification result we evaluate the quality of our aggregation method (shown as “F” in Figure 4.1) as well as the results indicate how the ratings assignment by the experts are influenced by the match outcome.

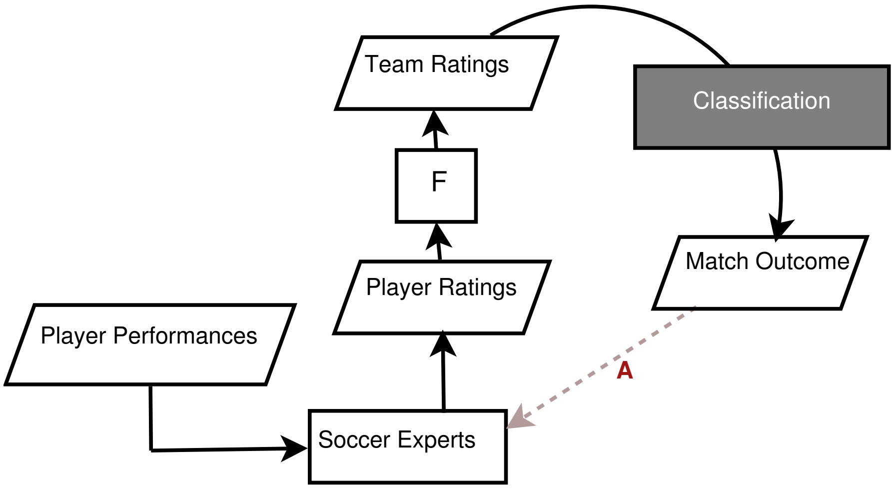

Figure 4.1: Predicting match outcome using ratings of player performances for current match

We should also note that this classification can be performed in two ways as shown in Figure 4.2 and 4.3. Our objective is to find a good “F”, which represents the process which aggregates the individual player ratings to give team ratings. In the first classification case, shown in Figure 4.2, we use only the ratings of player performances of the players involved in the *current match* to predict the match outcome. In the second case, as shown in Figure 4.3, we use the ratings of player performances for the *current match as well as the past matches* involving that player to predict the match outcome.

2. To investigate how well do team ratings generated from the aggregation method formulated during the previous objective can predict the next match outcome. In this classification task, shown in Figure 4.4, we use only the ratings for the past performances of the players.

4.2 Preparing the data

For this experiment we take the player dataset created for the first objective (Section 2.2) and then modify it. The dataset contains 10369 instances, one instance for every player’s performance in each match of the EPL. It has 195 attributes, all of which are player performance metrics including the expert rating assigned to the players performance in that match. For this experiment we also need to use information at match level like match date, team, opposition, venue which help us build a list of matches played. This list of matches played can then be appropriately linked to the respective match instances from the player dataset. Also, identifier attributes for

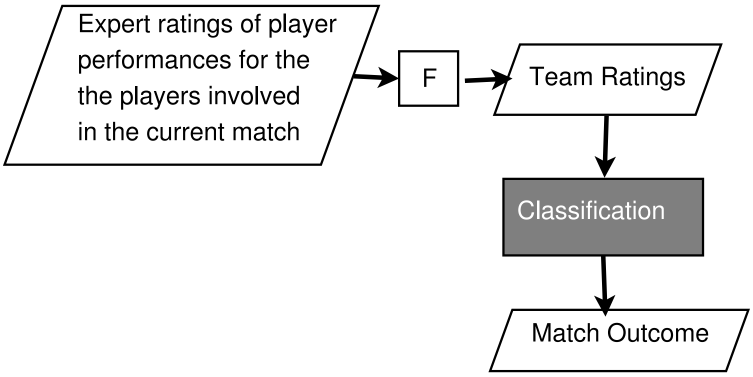

Figure 4.2: Predicting match outcome using ratings of player performances for current match

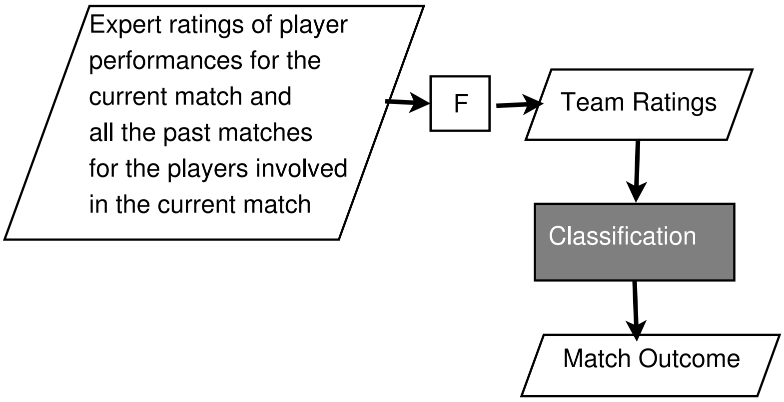

Figure 4.3: Predicting match outcome using ratings of player performances for current match and the past matches

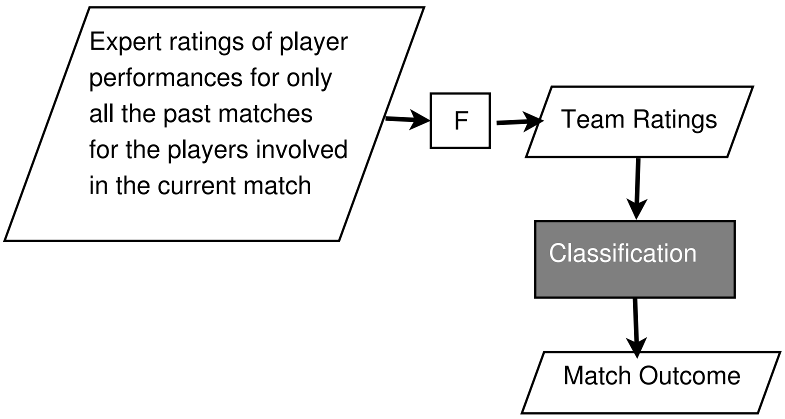

Figure 4.4: Predicting match outcome using ratings of player performances for current match using only the past match

players like their first name, last name, playing position are added. The resulting dataset which we use for our current experiment has 202 attributes.

4.3 Methodology: Computing Team Ratings

First our task is to find a good aggregation method to aggregate the player ratings for a team. Using the prepared dataset from previous section, for each match of the EPL, we record the performances of the players of the home and of the away team. The performances of the players is composed of following attributes: the expert rating assigned to them, goals scored by them, own goals scored by them and the position in which they played. Note that the attributes 'goals scored' and 'own goals' are needed to determine the match outcome for the teams involved while the 'position' attribute will be needed to generate position specific aggregated ratings. Using these attributes of player performances, we create a match dataset. This dataset contains the following attributes for each match: for the home team and away team we have *average* of all players' ratings, averages of ratings at each player position (attackers, defenders, goalkeepers and midfielders), *minimum* ratings at each player position, *maximum* ratings at each player position and the match outcome. Hence we have 27 attributes for matches in our dataset. The number of instances is 380 which equals the number of matches played.

However, these 27 attributes were created based on intuition and understanding of soccer. We should further check whether we can add additional attributes or modify our attributes for achieving better classification results. For this purpose, we execute different classification algorithms on the dataset while keeping the match outcome as the target class. We observe the models generated from these classifications. Additionally, we perform attribute selection using different criteria like gain ratio to see which attributes are considered more important by the algorithms. Based on these observations, we create 5 new team ratings attributes and add them to our dataset. These new attributes are, the difference of the team average ratings between the two competing teams and the differences of the position specific average ratings between the two competing teams.

Now, we will try to prune the least important attributes, i.e., those aggregated ratings attributes which have lower information content to characterise the match outcome. Hence, we select different subsets of attributes from the new list of team attributes, and execute classification algorithms on them. We observe these results and note that subset of attributes which gives best results. This selected subset of aggregated ratings attributes contains only 8 attributes and is shown in list.4.2. This set of attributes is better than other subsets of aggregated ratings attributes for characterising the current match outcome. Hence, we create a new match dataset which has only these 8 attributes for team ratings. Now we have two different match datasets created from aggregated player ratings, one of them contains 27 attributes while the other one contains the 8 selected attributes. The process of creating this subset of attributes is our "ratings aggregation method", F in Figure 4.2. This selected subset of attributes is the output of the "ratings aggregation method" and we will refer to this subset of attributes as *best aggregated ratings attributes*. Note that for aggregating the ratings we used only the ratings of player performances for the current match.

4.4 Results and Discussions: Computing Team Ratings

We execute the classification algorithms on the match dataset prepared in section 4.3 which has 27 attributes created as aggregations from the ratings of players involved in the match. The target class is match outcome. We observe that *Sequential Minimal Optimization, LibSVM, Bagging with Functional Trees and AdaBoost with Functional Trees* algorithms gives us good results in terms of percentage of correctly classified instances. These results are shown in Table 4.1. Then we try to estimate

TABLE 4.1: Predicting outcome with 27 rating attributes

	%Correctly classified	**ROC Area**	**F-Measure**	**Kappa**
SMO	89.7368	0.934	0.897	0.8401
LibSVM	88.6842	0.913	0.886	0.8235
BaggingFT	87.3684	0.971	0.872	0.8032
AdaFT	87.1053	0.956	0.87	0.7994
RBFNetwork	86.5789	0.953	0.865	0.7917

which attributes are most important for determining match outcome. We rank the attributes according to the different attributes selection criteria provided by the WEKA toolkit, notably, gain ratio based and correlation based feature subset selection. As a result, we observe that the team average rating of home and away teams are ranked highest in importance, followed by the averages for player positions. Next, we add the following attributes to our dataset, difference of team averages between home and away team, difference of average rating of attackers between home and away team, difference of average rating of defenders between home and away team, difference of average rating of goalkeepers between home and away team and differences of average rating of midfielders between home and away team. Now we apply the classification algorithms on the new dataset. We experiment with different subsets of attributes, for example one set of attributes comprised of the difference between the average ratings for both teams and the differences between the average ratings at each player position for both teams, this subset of attributes gives 88.68% correctly classified instances with *SMO* algorithm. When we use only the maximum and minimum ratings at each player position for both teams as our attributes, we get 87.37% correctly classified instances with *SMO* algorithm. When we use the subset of attributes containing difference between the team averages for both teams, differences between the average ratings at each player position for both teams, the minimum ratings for goalkeepers and defenders from both teams, and maximum ratings for midfielder and attacker for both teams, we get 89.21% correctly classified instances with *SMO* algorithm. In similar manner we try different subsets of attributes for classification and compare the results.

After several executions with different subsets of algorithms we conclude that it is better to use average rating of home team and average rating of away team as two separate attributes instead of using the difference between the team average ratings. However, for the position specific average ratings like average of the attacker's rating for home or away team, it is better to use the difference between the position specific average rating of the team team rather than to use the two attributes individually. From the result of these classifications we can say that team average ratings are more important than position average ratings which are in turn more important than the position maximum ratings. Moreover the maximum rating at player positions for the teams are more important than the minimum ratings. Also, the maximum ratings of the attackers in the team is more important than the maximum rating of midfielders. Based upon these insights we realise that the best set of attributes for our classification task (4.2) is: The best results obtained with this subset of attributes

Table 4.2: Best subset of ratings attributes for classifying match outcome

1. Team average rating for home team
2. Team average rating of away team
3. Difference of average of attacker's rating between home and away teams
4. Difference of average of defender's rating between home and away teams
5. Difference of average of goalkeeper's rating between home and away teams
6. Difference of average of midfielder's rating between home and away teams
7. Maximum ratings for attacker for home team and
8. Maximum ratings for attacker for away team.

is shown in Table 4.3. With *SMO* and *RBF Network* algorithms we obtain 90% correctly classified instances. With this result we can say that the list of attributes shown in Table 4.2 contains sufficient information to characterise the match outcome reasonably well. This validates the usefulness of our ratings aggregations method. Therefore the method to create the ratings attributes shown in Table 4.2 is our 'ratings aggregation method" (F in Figure 4.2).

Table 4.3: Predicting outcome with 8 selected(4.2) ratings attributes

	%Correctly classified	**ROC Area**	**F-Measure**	**Kappa**
SMO	90	0.946	0.901	0.8451
RBFNetwork	90	0.97	0.9	0.845
FT	89.2105	0.966	0.891	0.8316
LibSVM	87.6316	0.906	0.875	0.807
NaiveBayes	85.7895	0.971	0.865	0.7841

4.5 Methodology: Correlating Team Ratings with Match Outcomes

Next, using this *best aggregated ratings attributes* we perform classification task shown in Figure 4.3. The target class is the match outcome. The *best aggregated ratings attributes* generated are to be used for this classification. However, the aggregated ratings attributes can be aggregation of player ratings for the current match only or it can be aggregation of player ratings for the current match and as well as the past matches played by that player. More over, the number of past matches to include in ratings aggregation can be controlled through a window. We should also note that the performances of the player change over the course of time. Sometimes a player is in good form and sometimes not. For this reason we will employ a weighting scheme when we aggregate the player ratings inclusive of past matches. This scheme will assign more weight to the more recent match performances. Through this scheme we assign a weight equal to the window size to the most recent match, and keep on decreasing the weight by one as we include more past matches. Hence, if the window size is three, the current match gets a weight of three, the previous match gets a weight of two and the match before that gets a weight of one. And the resultant aggregated ratings for a match is the weighted average of the ratings of the current match and all the past matches included in the window. However, in cases when the window size is larger than the number of matches played by that player until that match, we include only the ratings only for the matches available and calculate the weighted average. This current weighted average scheme ensures that in all cases the more recent match has more contribution to resultant aggregated ratings.

We do a series of classification experiments. Initially we use a window size of two, including only the ratings from current match and the previous match and predicting the match outcome of current match. [1] Next, we increase the window size in steps of one until we reach a window size of ten where we are including the ratings from the current match and the past nine matches played by the players. Once we perform this series of classification tasks using all the 27 attributes and then we classify using only the 8 attributes mentioned in the list 4.2. We observe the trend of performance of the qualification over the increasing window size.

4.6 Results and Discussions: Correlating Team Ratings with Match Outcomes

The results for predicting the match outcome using the player ratings from the current match and the past matches are shown in Table 4.4 and the plot is shown in Figure 4.5.The window size indicates the number of matches past matches (including the current match) used for aggregating the ratings. Also, note that more importance has been given to more recent match using a weighted average aggregation scheme.

[1] We have already performed classification using window size of one in 4.3 when we were trying to find the best subset of aggregated ratings attributes evaluated on the basis on the basis of how well they could characterise match outcome.

Table 4.4: Percentage of Correctly Classified Instances: Classifying Match Outcome with Aggregated Ratings, using *SMO* Algorithm

Window Size	With 27 Attributes	With Best 8 Attributes
1	89.7368	90.0000
2	78.4211	79.2105
3	74.7368	74.7368
4	69.2105	71.0526
5	67.8947	68.1579
6	65.5263	65.2632
7	64.7368	64.7368
8	61.8421	62.6316
9	61.8421	61.8421
10	61.0526	61.5789

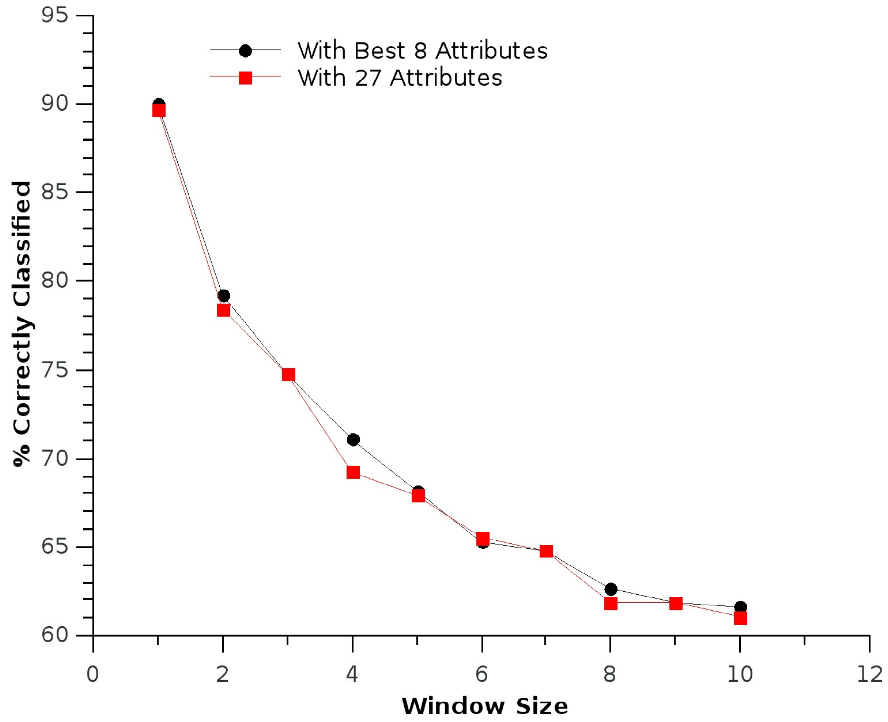

Figure 4.5: Classifying Match Outcome with Aggregated Ratings, using *SMO* Algorithm

From the plot of this data in Figure 4.5, we can see that the classification accuracy for match outcome decreases rapidly with increasing window size. This trend shows that the outcome of a soccer match is "most" dependent on the current match performance only. And the further we consider the past performances, the worse our prediction becomes. This trend is seen even though we used a weighted scheme to take more contribution from the recent matches and lesser contribution from past matches. However, even this less contribution from past matches is enough to significantly drop the prediction performance. So the decrease in prediction performance using an unweighted aggregation scheme would have been steeper than shown in Figure 4.5. Hence the past performances of the player have very less influence over the outcome of the current match. Nevertheless, this is an empirical observation and does not rule out the psychological dependency of the past performances on the current performance of the individual players. We should also note that the best 8 attributes (mentioned in 4.2) are better at predicting match outcome compared to using any other subset of these attributes or using all the attributes. This reiterates the importance of these 8 attributes over others.

4.7 Methodology: Predicting Match Outcomes

This task is intended to satisfy the last objective of the thesis. As shown in Figure 4.4, this classification task is for *prediction* unlike the previous classification which were descriptive in nature, meant for characterising the match outcome with different kinds of attributes. The primary difference being that, in this classification task we use the player ratings only for the past matches and do not include the ratings for the current match. For aggregating the player ratings over a match, we use the same aggregation scheme, "F", as described in section 4.4. Moreover we can either use all the 27 aggregated attributes or we can use the 8 the attributes mentioned in the list. 4.2. Similar to the previous section 4.5, we aggregate the player ratings over a window of past matches. However, this window will not include the current match. Hence a window size of three means that we include the ratings for the three matches previous to the current match. And similarly we use a weighted average scheme for aggregating the ratings over the matches. For example, with a window size of three, the previous match gets a weight of three, the next one gets a weight on two and the last match in the window gets a weight of one. In case the number of past matches available is less than the window size, we use only the matches available and compute the weighted average of the aggregated ratings over all matches available in the window. Trivially, the first match of each team is not included in the dataset. The intent of this scheme is to have more contribution from recent matches in the resultant aggregated ratings and the contribution decreasing linearly as we go more in the past.

4.8 Results and Discussions: Predicting Match Outcomes

We execute different classification algorithms and find that *Sequential Minimal Optimization Algorithm* gives one of the best classification results. The results for these classifications are shown in Table 4.5 and plotted in Figure 4.6 We observe

Table 4.5: Percentage of Correctly classified instances: Predicting Next Match Outcome with Aggregated Ratings of Past Matches, using *SMO* Algorithm

Window Size	**With 27 Attributes**	**With Best 8 Attributes**
1	44.9864	45.5285
2	47.9675	45.5285
3	48.5095	46.3415
4	49.5935	49.0515
5	49.8645	49.8645
6	52.8455	51.2195
7	53.3875	51.2195
8	52.8455	51.4905
9	51.7615	53.1160
10	52.0325	52.8455

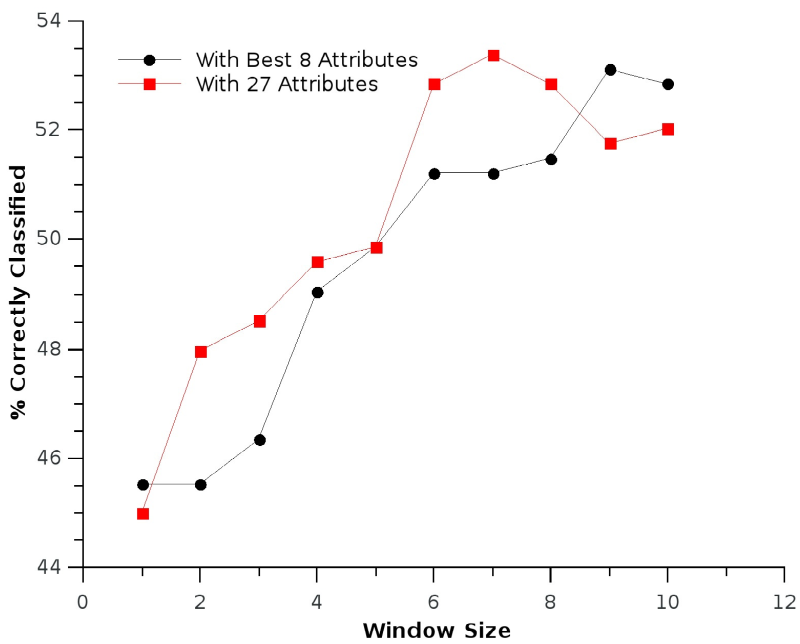

Figure 4.6: Predicting Next Match Outcome with Aggregated Ratings of Past Matches, using *SMO* Algorithm

that the prediction acccuracy increases with increasing window size until a critical point. When we are using all the 27 attributes this critical point is reached at a window size of 8 and when using only the 8 attributes mentioned in the list 4.2, we reach this critical point at a window size of 9. The prediction accuracy decrease after this critical point in both cases. The best performance we managed to achieve is 53.39 % correctly classified instances with a ROC area of 0.587. The kappa statisitic is 0.2018, hence this prediction is better than a random guess. We observe that the prediction performances are poor in contrast to the descriptive classification task shown in Figure 4.5. Another point of contrast between results of this predictive and previous descriptive task (Figure 4.5) is that for prediction the classification accuracy increase with increasing window size up until the critical point while for descriptive classification the accuracy always decreased with increasing window size. This points to the fact the outcome of the current is *best* characterised by the ratings of player performances for the current match "only". Hence it is extremely difficult to predict the outcome of future matches.

For our match dataset, the class distribution is , 'Home Team Wins' in 45% instances, 'Away Team Wins' in 30.8% instances while the match is draw in 24.2% instances. For our best prediction results (Figure 4.6), with accuracy of 53.39%, the F-measure for classifying 'Home Team Wins' is 0.665, for 'Away Team Wins' is 0.484 and for 'Draws' is 0. The overall weighted averaged F-measure being 0.453. Hence we conclude that it is most difficult to predict a 'Draw'. When we remove all the instances whose outcome is a draw. Then our prediction accuracy jumps to 67%. This proves that an important reason for the low accuracy of our prediction for all target classes is because of the difficulty to predict a draw in our classification task. This means that the whether a match outcome will be a 'Draw' or not cannot be predicted using the ratings player performances. A 'Draw' almost negligibly depends on the player performances but depends on other factors, which could just be sheer luck or biases of referees or some other factor of the system which is not captured by player performance ratings.

4.9 Conclusion

We performed sets of classification tasks to achieve our objectives. We were able to find a good ratings aggregation method in section 4.4. Using this aggregation strategy we find the set of attributes which can best characterise the match performance. This set of attributes is shown in list 4.2. Next, in section 4.6 we tried to characterise the match outcome using the attributes generated from aggregated player ratings. We find that the attributes from the current match are the most useful for this task. And the subsequent past matches have decreasing influence on the current match outcome. Our best result for characterising match outcome was an accuracy of 90% using the best 8 attributes only. This shows a positive correlation between match outcome and the ratings given by the soccer experts. The experts are influenced by match results while assigning ratings to the player performances, hence the line A in the Figure4.1 is present. In the last part of this chapter, section 4.8, we predicted the outcome of next match using the attributes created from the aggregated ratings of player performances over the past matches. The best accuracy we obtained was 53.39% which is better than a random guess. We observed that this low accuracy was partly due to the extreme difficulty of predicting a ‘Draw’ for a match. This low accuracy also means that the outcome of a match depends only the players’ performances in the current match and not the performances of the past matches.

Chapter 5

Conclusion

"Every match is a contradiction, being at once both highly predictable and highly unpredictable. The simplicity of the rules and the familiarity of the tactical moves make every moment of play immediately understandable to the watching eyes. But despite this, nobody can ever be sure what will happen next." Morris [1981]

In this thesis we used Machine Learning techniques on soccer match data to accomplish four objectives. Our first objective was to identify the most important attributes of player's performance which determine the ratings assigned by soccer experts. For this, we performed a series of classification experiments with suitable attribute selection. We split our dataset for the four player positions i.e., attackers, defenders, goalkeepers and midfielders. For finding the optimal set of performance attribute for each position, we employed three different pruning strategies in our classification experiments, namely, *Global Ranked Pruning*, *Iterative Local Pruning* and *Threshold pruning*. After observing the results from the different classification setups, we selected a set of lists of optimal attributes for each player position. The best classification results for each player position with optimal attributes was as follows: for attackers, mean absolute error of 0.17; for defenders, mean absolute error of 0.28; for goalkeepers, mean absolute error of 0.22 and for midfielders, mean absolute error of 0.18. Then for each player position, all the optimal lists of attributes were rank-aggregated using a 'reward and penalty' method to generate a single list of performance attributes. These performance attributes best determine the ratings assigned by the experts. Thus, we identified a close approximation of the latent performance metrics which the soccer experts at WhoScored.com must have considered to assign ratings to player performances.

The second objective was to find out which performance attributes of the players of the two competing teams affect the match outcome and to what extend the match outcome is characterised by the performance attributes of players. We prepared datasets at player level, match level and team level. Next, we performed a set of classification tasks with suitable attribute selection. From our results we conclude that the match outcome is best determined by a set of 34 player performance attributes. We had 308 performance attributes in our datasets which had almost negligible information to characterise the match outcome. We also observed that using the selected 34 performance attributes we can predict the match outcome with an accuracy of 63.4%.

The third objective required to find a good aggregation method to generate team ratings from the player ratings. We presented an aggregation strategy using which we produced 27 attributes for the team. Next, we selected a list of 8 attributes aggregated from the team ratings which could predict the match outcome with an accuracy as high as 90%. The second part of the third objective was to investigate how closely these team ratings can determine the match outcome. Towards this end, we performed a series of classification tasks where the teams ratings were an aggregation of the ratings from the current match as well as the past matches. We used a weighted average strategy to assign more importance to ratings from the recent matches. The number of past matches to be included in our aggregation was varied through a window size. The prediction performance rapidly deteriorated with increasing window size. This indicated that the outcome of a soccer match is highly correlated with the "current match performance ratings only" and the past performance ratings of the player have very less influence. This in turn asserts the hypothesis that the experts are influenced by match outcomes when they assign ratings to player performances. It was also observed that the best 8 aggregated attributes, which we selected are better at predicting match outcome compared to any other set of aggregated attributes. Thus, using our aggregation strategy we generated a set of aggregated attributes which can closely determine the current match outcome, if the ratings from the current match are used.

Our fourth objective was to investigate the prediction of future matches using the ratings from the past matches. For this task we generated the team ratings attributes from the player ratings using a similar aggregation method as the one used for the third objective. However, to generate the team ratings attributes we used only the ratings from the previous matches. The number of previous matches to include was regulated using a window. We performed the prediction using two different sets of aggregated team attributes, one contained 27 attributes while the other contained 8 attributes. We observe that the prediction of future matches is difficult. We obtain our best prediction at an accuracy of 53.4% when we use the ratings from the past 7 matches. We conclude that the reasons for this low prediction result are the difficulty to predict a 'draw' using our dataset and that the outcome of a match is dependent mostly on the performance ratings of the current match and not of the past matches.

This thesis has explored the applications of Machine Learning techniques in soccer analytics. This work can be extended with availability of more detailed datasets. For example, in this thesis we were restricted to only four player positions which could be easily extended to more specific player positions if appropriate identifiers are available in the dataset. Furthermore, for the aggregation of ratings over the past matches, we used a weighted averaging method, however, better aggregation schemes can be explored. In this thesis, we limited ourselves to notational analysis of soccer and using only technical and tactical performance indicators. Future works can extend the applications of Machine Learning to include biomechanical analysis of soccer using different performance indicators. With the availability of better match recording tools, intelligent video analysis technologies and growing interest for soccer in the analytics community, it is expected that richer datasets will be made available in the future and novel learning algorithms might be developed. In this thesis, we have validated the usefulness of Machine Learning techniques in soccer analytics. The opportunities for applications of Machine Learning techniques in soccer analytics are bound to grow.

Appendix A

Soccer Event Definitions

Some of these event definitions are for the new attributes added to the dataset while others are for the attributes from the OPTA dataset.[1].

Own Goals

Goals scored by a player against his own team.

Shot on target

Any goal attempt that goes into the net, would have gone into the net but for being stopped by a goalkeeper's save or would have gone into the net but for being stopped by a defender who is the last man.

Shot on target

Any goal attempt that goes into the net, would have gone into the net but for being stopped by a goalkeeper's save or would have gone into the net but for being stopped by a defender who is the last man.

Shot off target

Any goal attempt where the ball is going wide of the target, misses the goal or hits the woodwork.

Blocked Shot

Any goal attempt heading roughly on target toward goal which is blocked by a defender, where there are other defenders or a goalkeeper behind the blocker.

Goal Assist

The final pass or pass-cum-shot leading to the recipient of the ball scoring a goal.

Total Goals Scored

Sum of goals scored by the team and the own goals scored by the opposition.

Total Goals Conceeded

Sum of goals scored by the opponent team and the own goals done by the team

Second Assist/Key Pass

A pass/cross that is instrumental in creating a goal-scoring opportunity, for example a corner or free-kick to a player who then assists an attempt, a chance-creating through ball or cross into a dangerous position.

Through Ball

A pass splitting the defence for a team-mate to run on to. Each pass is logged with

[1]OPTA Event Definitions: www.optasports.com

X and Y co-ordinates for its point of origin and destination.
Dribbles/Take-ons
This is an attempt by a player to beat an opponent in possession of the ball. A successful dribble means the player beats the defender while retaining possession, unsuccessful ones are where the dribbler is tackled.
Tackles
A tackle is defined as where a player connects with the ball in ground challenge where he successfully takes the ball away from the man in possession. A *Tackle Won* is deemed to be where the tackler or one of his team-mates regains possession as a result of the challenge, or that the ball goes out of play and is "safe". A *Tackle Lost* is where a tackle is made but the ball goes to an opposition player.
Clearance
This is a defensive action where a player kicks the ball away from his own goal with no intended recipient of the ball.
Block
This is where a player blocks a shot from an opposing player. Interception This is where a player intentionally intercepts a pass by moving into the line of the intended ball.
Recovery
This is where a player wins back the ball when it has gone loose or where the ball has been played directly to him.
Shield ball out of play
Where a player shields the ball from an opponent and is successful in letting it run out of play.
Foul conceded
Any infringement that is penalised as foul play by a referee.
Foul won
Where a player is fouled by an opponent. There is no foul won for a handball or a dive where a free kick is conceded.
Offside
Awarded to the player deemed to be in an offside position where a free kick is awarded.
Duels
A duel is an 50-50 contest between two players of opposing sides in the match. For every Duel Won there is a corresponding Duel Lost depending on the outcome of the Duel.
Aerial Challenge won - Aerial Challenge lost
This is where two players challenge in the air against each other. The player that wins the ball is deemed to have won the duel. When more than two players are involved the player closest to the duel winner is given an Aerial Duel lost.
Successful Take-on/Dribble - Challenge lost
The player who has been beaten is given a Challenge lost if they do not win the ball.
Tackle - Unsuccessful Take-on/Dispossessed
A tackle is awarded if a player wins the ball from another player who is in possession. If he is attempting to beat the tackler, the other player will get an unsuccessful

Take-on. If he is in possession but not attempting to "beat" his man, then he will get a dispossessed.

Foul won-Foul conceded

The player winning the foul is deemed to have won the duel and the player committing the foul having lost the duel.

Save

A goalkeeper preventing the ball from entering the goal with any part of his body.

Clean Sheet

A player or team who does not concede a goal for the full match.

Catch

A high ball that is caught by the goalkeeper

Punch

A high ball that is punched clear by the goalkeeper.

Drop

A high ball where the goalkeeper gets hands on the ball but drops it from his grasp.

Appendix B

A Brief Introduction to Terms and Algorithms from Machine Learning

This appendix is created using information obtained from WEKA documentation [Hall et al., 2009] *and Wikipedia* [Wikipedia, 2013d].

Linear Regression
This algorithm uses linear regression for prediction. Additionally the WEKA implementation uses the Akaike criterion for model selection and is able to deal with weighted instances

SMOreg
This algorithm implements sequential minimal optimization algorithm for training a support vector regression model.

Gaussian Processes
This algorithm implements the 'Gaussian Processes' for regression. Gaussian process is a stochastic process popular in statistics and probability

LeastMedSq
This generates a least median squared linear regression. It utilises the existing WEKA LinearRegression class to form predictions. Least squared regression functions are generated from random subsamples of the data. The least squared regression with the lowest median squared error is chosen as the final model.

M5P
This algorithm creates base routines for generating M5 Model trees and rules. Model trees are binary decision trees with linear regression functions at the leaf nodes.

Bagging with REP Tree
Bootstrap aggregating (bagging) is a machine learning ensemble meta-algorithm designed to improve the stability and accuracy of machine learning algorithms used in statistical classification and regression. It also reduces variance and helps to avoid overfitting. REP Trees are fast decision tree learner which builds a decision/regression tree using information gain/variance and prunes it using reduced-error pruning (with

backfitting).

Additive Regression with Decision Stump

This is a meta classifier that enhances the performance of a regression base classifier. Each iteration fits a model to the residuals left by the classifier on the previous iteration. Prediction is accomplished by adding the predictions of each classifier. Reducing the shrinkage (learning rate) parameter helps prevent overfitting and has a smoothing effect but increases the learning time. A decision stump is a machine learning model which consists of a single level decision tree.

REP Tree

This is a fast decision tree learner which builds a decision/regression tree using information gain/variance and prunes it using reduced-error pruning (with backfitting).

Regression by Discretization using J48

This is a regression method which usees any classifier on a copy of the data that has the class attribute discretized. The predicted value is the expected value of the mean class value for each discretized interval (based on the predicted probabilities for each interval).

Decision Table

This algorithm builds uses a simple decision table majority classifier. Decision tables are a compact way to model complicated logic.

Multilayer perceptron

A multilayer perceptron (MLP) is a feedforward artificial neural network model that maps sets of input data onto a set of appropriate outputs. This classifier uses backpropagation to classify instances. The nodes in this network are all sigmoid (except for when the class is numeric in which case the the output nodes become unthresholded linear units).

Simple Linear Regression

This classifier learns a simple linear regression model. It does not allow missing values and can only deal with numeric attributes. It select the attribute that gives the lowest squared error.

Locally Weighted Learning

This is an instance-based algorithm to assign instance weights which are then used by a specified WeightedInstancesHandler. It does classification (e.g., using naive Bayes) or regression (e.g., using linear regression).

IBk

This is where a player blocks a shot from an opposing player. Interception This is where a player intentionally intercepts a pass by moving into the line of the intended ball.

KStar

This is a K-nearest neighbours classifier. This instance based classifier can select appropriate value of K based on cross-validation. It can also do distance weighting.

RBF Network

This classifier models a normalized Gaussian radial basis function network. It uses the k-means clustering to provide the basis functions and learns either a logistic regression (discrete class problems) or linear regression (numeric class problems) on top of that.

Functional Trees
This classifier builds 'Functional trees', which are classification trees having logistic regression functions at the inner nodes and/or leaves.
Naive Bayes
This is a simple probabilistic classifier based on applying Bayes' theorem with strong independence assumptions.
Random Forest
This is an ensemble learning method for classification that operates by constructing a set of decision tree while training and then outputs the class which is the mode of the classes output by individual trees.
Fuzzy Unordered Rule Induction Algorithm
This is a fuzzy rule based classification method which is an extension of the the RIPPER algorithm.
J48Graft
This classifier generates grafted(pruned or unpruned) C4.5 decision tree.
J48
This classifier generates a pruned or unpruned C4.5 decision tree.
JRip
This classifier implements the propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction (RIPPER).
LibSVM
This classifier implements kernelized support vector machines (SVMs) for classification and regression.
AdaBoostM1
This ensemble method is used for boosting a nominal class classifier using the Adaboost M1 method. Only nominal class problems can be tackled. Often it dramatically improves performance, but sometimes it might overfit.
Mean Absolute Error
This is an average the magnitude of the individual errors without taking account of their sign.
Mean Square error
This is the mean of the squares of the individual error.
Relative Absolute Error
This is computed by dividing the absolute error by the absolute error obtained by just predicting the mean of target values (and then multiplying by 100). Therefore, smaller values are better and values larger than 100% indicates a scheme is doing worse than just predicting the mean.
Model Building Time in seconds
This is the time taken in second by the classifier to build a model.
Model Testing Time in seconds
This is the time taken to test the model through the selected testing scheme. In this thesis we use 10-fold stratified cross validation.
ROC
The Receiver Operating Characteristic is a plot of true positive rate against false positive rate. The best possible prediction method would yield a point in the upper

left corner or coordinate (0,1) of the ROC space, representing 100% sensitivity and 100% specificity. The (0,1) point is also called a perfect classification. A completely random guess would give a point along a diagonal line from the left bottom to the top right.

Kappa

This statistic is used to measure the agreement between predicted and observed categorizations of a dataset, while correcting for agreement that occurs by chance.

F-Measure

It is a measure of a test's accuracy. It is calculated as the harmonic mean of precision and recall.

Appendix C

Tables and Figures

C.1 Results from Chapter 2

Table C.1: Algorithms results with Defenders dataset (195 attributes)

Algorithm	CC	MAE	RMSE	RelAE	BTime(s)	TTime(s)
LinearRegression	0.8530	0.2728	0.3477	52.5298	7.4000	61.5210
SMOreg	0.8515	0.2695	0.3508	51.9098	133.5630	1074.4590
GaussianProcesses	0.8537	0.2714	0.3468	52.2765	102.3460	1163.7580
LeastMedSq	0.8484	0.2766	0.3530	53.2780	17.0750	150.9360
M5P	0.8094	0.2989	0.3912	57.5652	7.3880	58.8420
Bagging	0.7483	0.3452	0.4461	66.4898	6.0330	50.7270
AdditiveReg(DS)	0.7021	0.3717	0.4770	71.5811	0.8370	7.0470
REPTree	0.6456	0.3935	0.5141	75.7797	0.5350	5.0970
RegByDiscret(J48)	0.6007	0.4376	0.5708	84.2812	3.1870	35.8870
DecisionTable	0.6579	0.3904	0.5019	75.1959	8.3880	65.6380
MultilayerPerceptron	0.6195	0.4596	0.5930	88.5213	738.8830	6661.6850
SimpleLinearReg	0.4198	0.4610	0.6044	88.7868	0.0410	0.2160
LWL	0.3672	0.4735	0.6197	91.1902	0.0010	262.6720
IBk	0.4933	0.4871	0.6353	93.8054	0.0010	15.2430
KStar	0.3468	0.5219	0.6898	100.5211	0.0000	315.2390
RBFNetwork	0.2072	0.4976	0.6515	95.8396	2.3190	16.2310

Table C.2: Algorithms results with Midfielders dataset (195 attributes)

Algorithm	CC	MAE	RMSE	RelAE	BTime(s)	TTime(s)
LinearRegression	0.9504	0.1729	0.2356	29.2375	8.7120	73.4430
SMOreg	0.9505	0.1711	0.2354	28.9420	615.9230	4815.6860
GaussianProcesses	0.9505	0.1736	0.2352	29.3556	237.5800	2837.2170
LeastMedSq	0.9493	0.1764	0.2401	29.8332	22.5860	199.3370
M5P	0.9310	0.1962	0.2764	33.1745	12.0670	102.7850
Bagging	0.8695	0.2766	0.3754	46.7726	7.6450	64.3360
AdditiveReg(DS)	0.8327	0.3220	0.4221	54.4522	1.1280	9.5650
REPTree	0.8194	0.3232	0.4358	54.6480	0.7320	6.3060
RegByDiscret(J48)	0.7643	0.3762	0.5003	63.6172	3.1570	42.1640
DecisionTable	0.7748	0.3488	0.4786	58.9893	15.3530	109.7970
MultilayerPerceptron	0.3074	0.6129	0.7962	103.6440	980.8430	8829.5320
SimpleLinearReg	0.5386	0.5057	0.6377	85.5205	0.0480	0.3000
LWL	0.5783	0.4802	0.6190	81.2000	0.0000	469.4600
IBk	0.6619	0.4385	0.6032	74.1500	0.0000	17.6990
KStar	0.5367	0.4902	0.6869	82.8922	0.0000	559.4990
RBFNetwork	0.3182	0.5405	0.7176	91.4083	2.8190	13.0720

Table C.3: Algorithms results with Goalkeepers dataset (195 attributes)

Algorithm	CC	MAE	RMSE	RelAE	BTime(s)	TTime(s)
LinearRegression	0.9505	0.2126	0.2808	30.0825	0.6290	4.4690
SMOreg	0.9478	0.2175	0.2880	30.7785	2.0410	15.5590
GaussianProcesses	0.9517	0.2142	0.2778	30.3161	0.8860	10.6840
LeastMedSq	0.9484	0.2164	0.2861	30.6154	4.0690	35.8810
M5P	0.9317	0.2330	0.3278	32.9767	0.8060	5.6410
Bagging	0.8721	0.3195	0.4458	45.2112	1.0040	6.7320
AdditiveReg(DS)	0.8818	0.3329	0.4353	47.1052	0.1970	1.3950
REPTree	0.8134	0.3829	0.5275	54.1887	0.0750	0.6670
RegByDiscret(J48)	0.8134	0.4016	0.5325	56.8258	0.4240	2.8130
DecisionTable	0.7966	0.3804	0.5475	53.8284	1.0620	8.6860
MultilayerPerceptron	0.8163	0.4231	0.5585	59.8716	173.1670	1564.8510
SimpleLinearReg	0.5044	0.6019	0.7795	85.1652	0.0260	0.0480
LWL	0.5296	0.5811	0.7673	82.2322	0.0000	12.6100
IBk	0.5773	0.6166	0.7862	87.2469	0.0010	1.4610
KStar	0.4380	0.6503	0.8709	92.0200	0.0000	18.3420
RBFNetwork	0.0796	0.7017	0.8999	99.2932	0.2360	3.1510

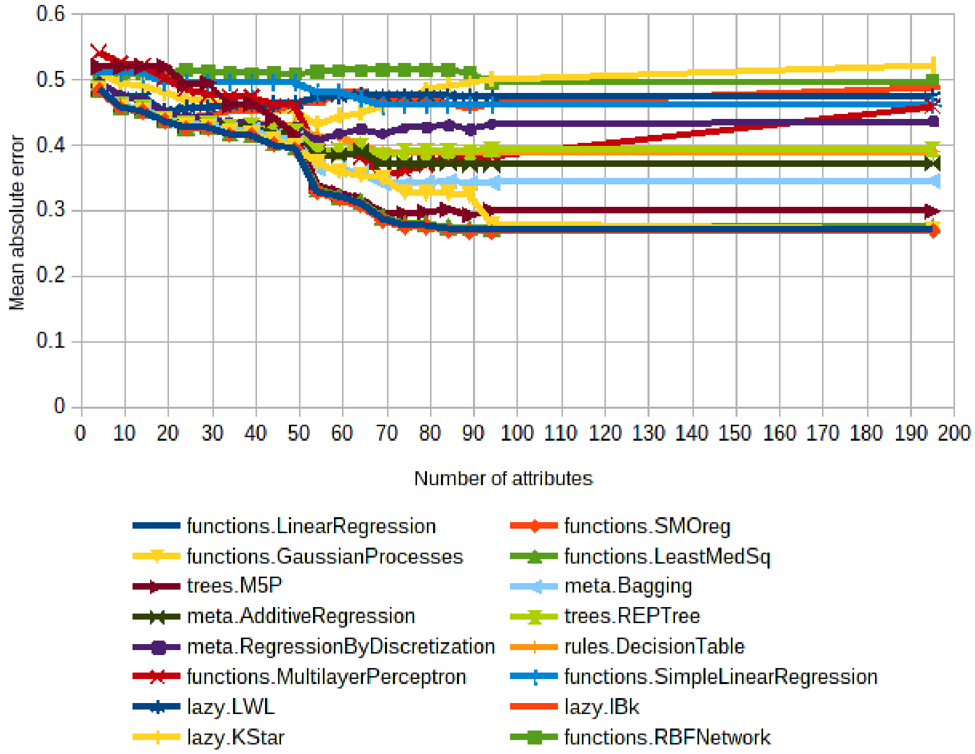

Figure C.1: Global Ranked Pruning (Defenders): Mean Absolute Error Vs Number of Attributes

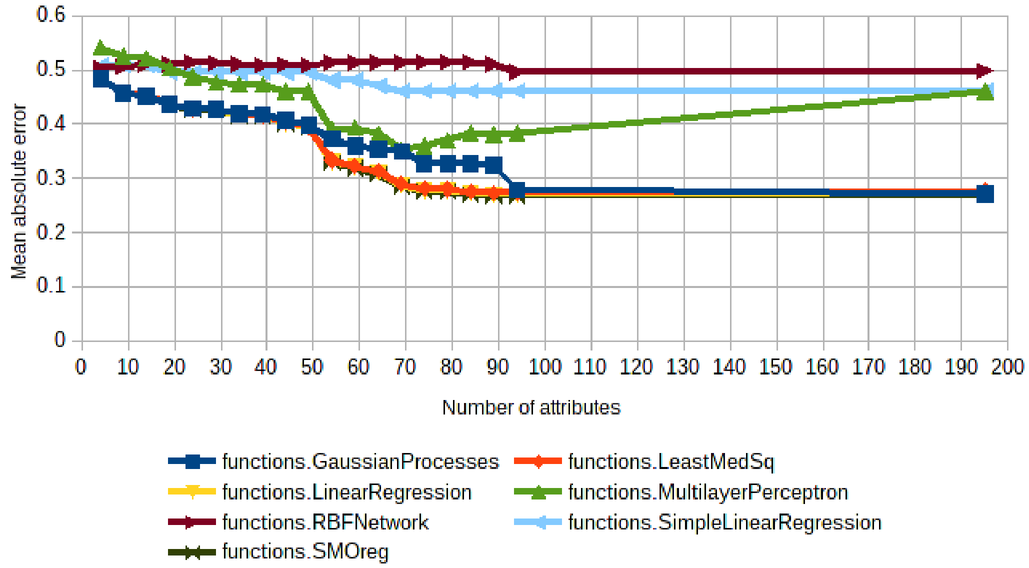

Figure C.2: Global Ranked Pruning (Defenders): Function based algorithms -Mean Absolute Error Vs Number of Attributes

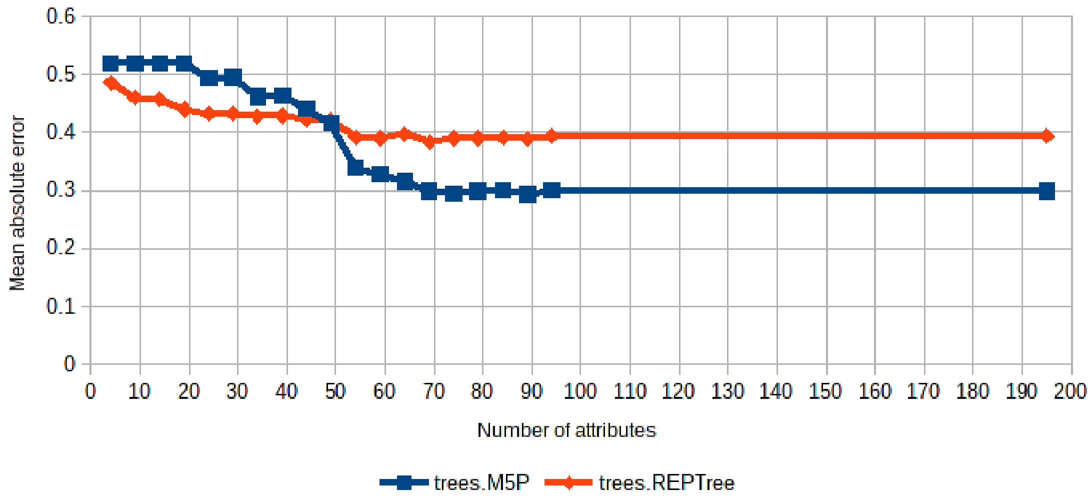

Figure C.3: Global Ranked Pruning (Defenders): Tree based algorithms -Mean Absolute Error Vs Number of Attributes

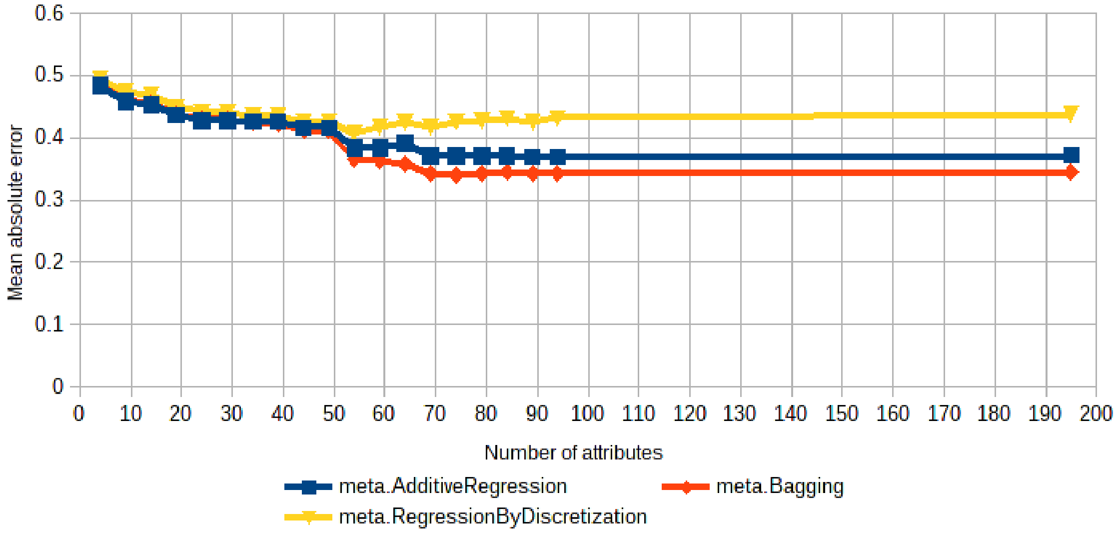

FIGURE C.4: Global Ranked Pruning (Defenders): Meta algorithms -Mean Absolute Error Vs Number of Attributes

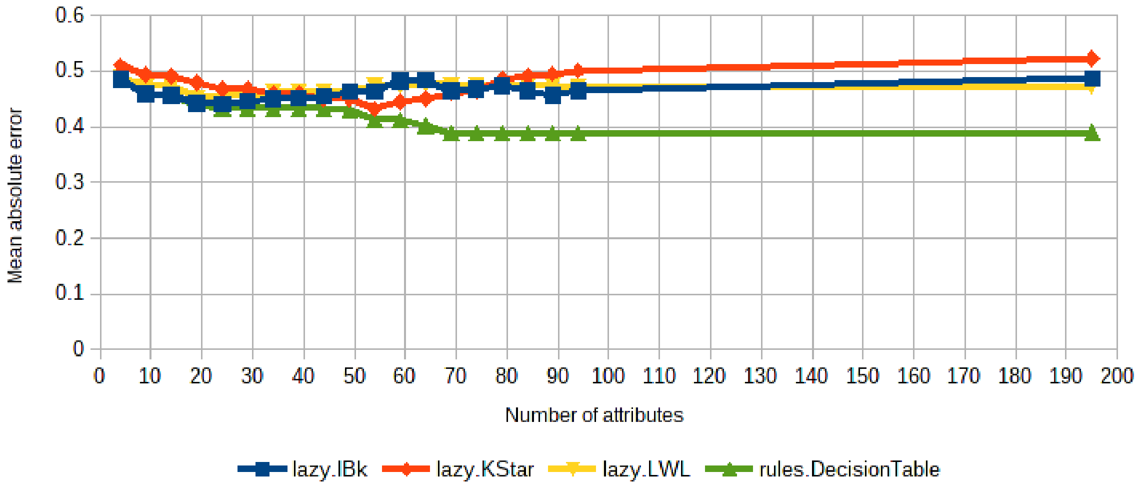

FIGURE C.5: Global Ranked Pruning (Defenders): Lazy and Rule based algorithms -Mean Absolute Error Vs Number of Attributes

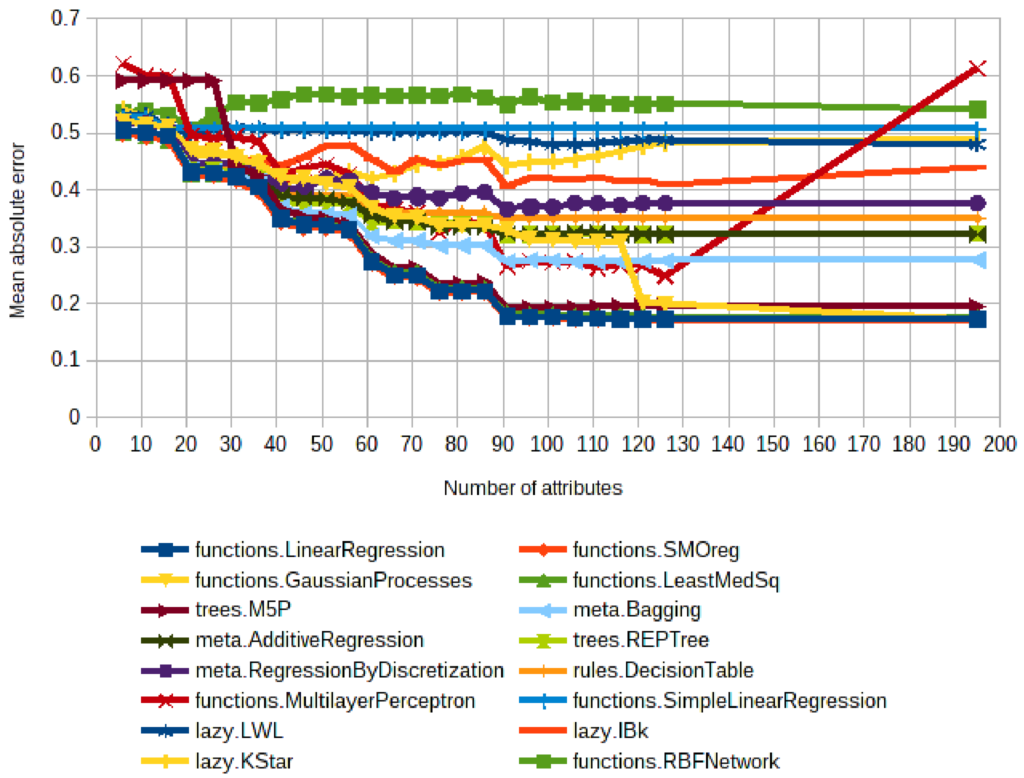

Figure C.6: Global Ranked Pruning (Midfielders): Mean Absolute Error Vs Number of Attributes

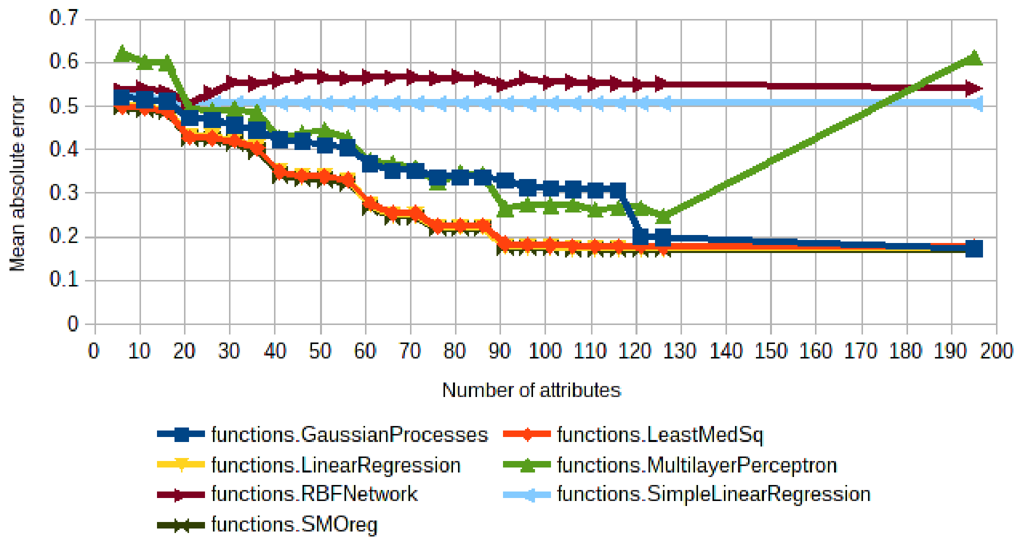

Figure C.7: Global Ranked Pruning (Midfielders): Function based algorithms -Mean Absolute Error Vs Number of Attributes

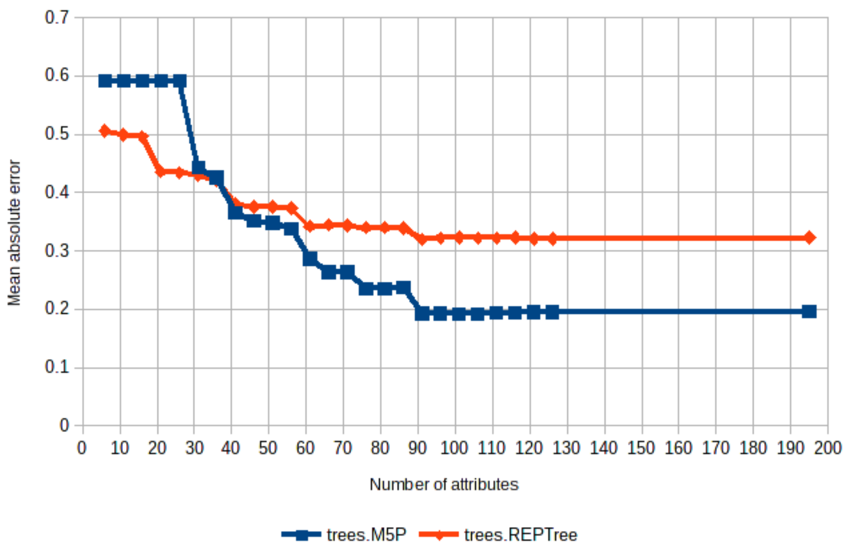

FIGURE C.8: Global Ranked Pruning (Midfielders): Tree based algorithms -Mean Absolute Error Vs Number of Attributes

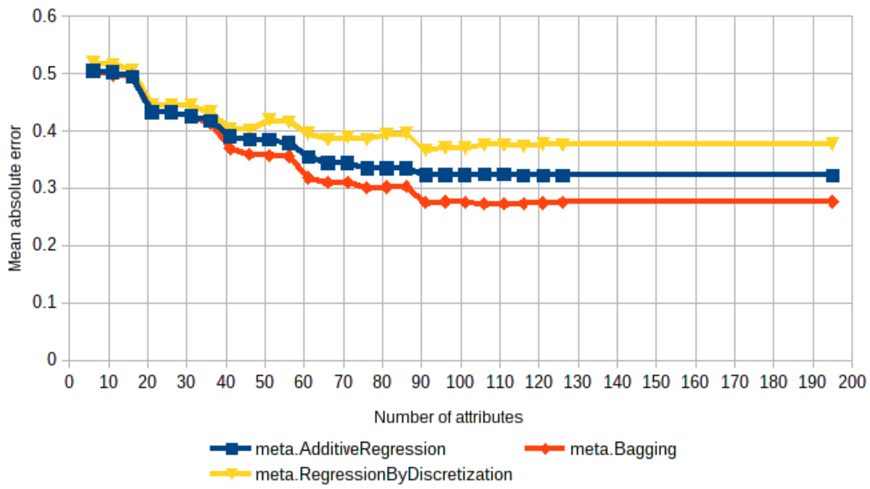

FIGURE C.9: Global Ranked Pruning (Midfielders): Meta algorithms -Mean Absolute Error Vs Number of Attributes

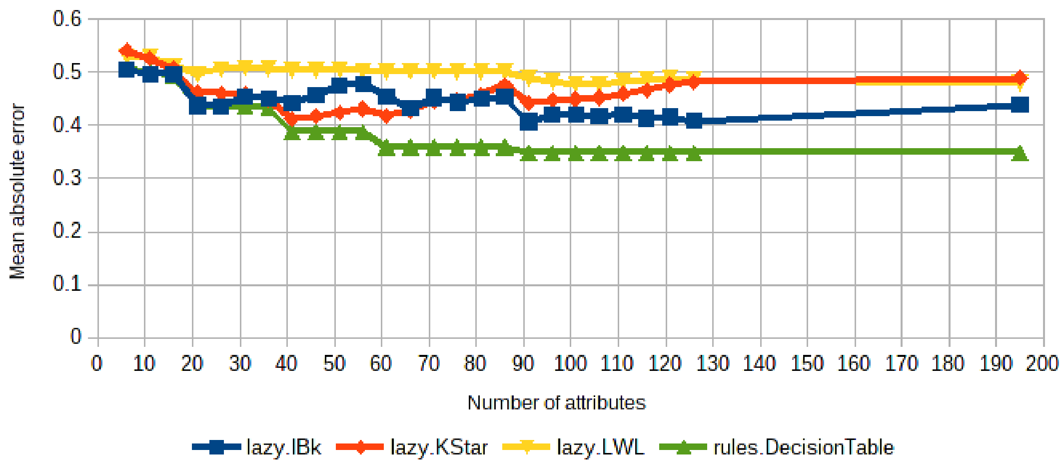

Figure C.10: Global Ranked Pruning (Midfielders): Lazy and Rule based algorithms -Mean Absolute Error Vs Number of Attributes

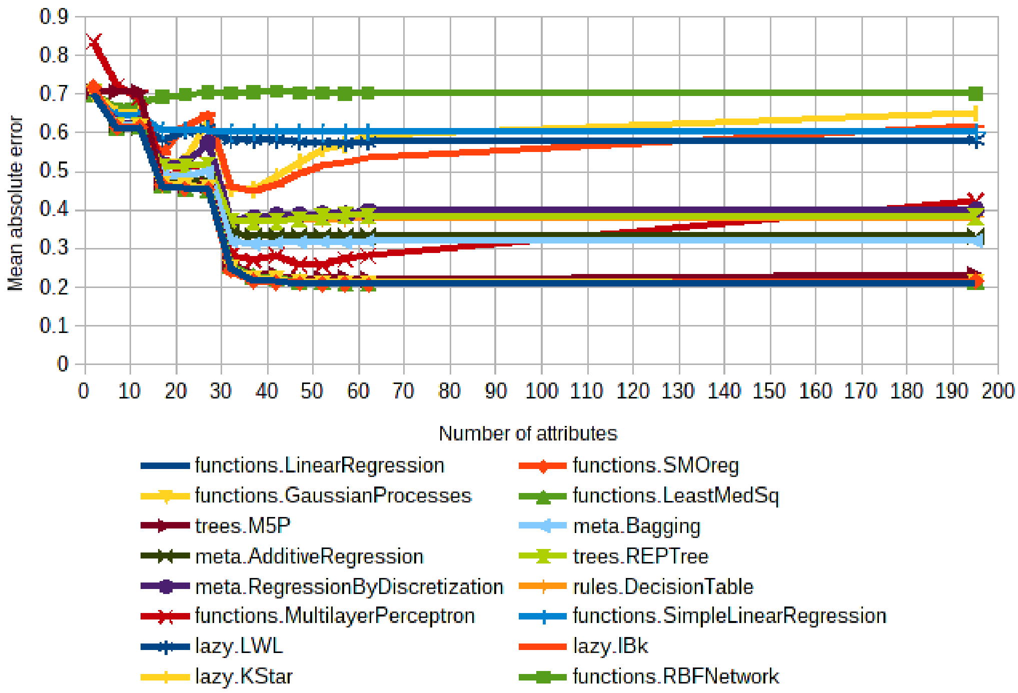

Figure C.11: Global Ranked Pruning (Goalkeepers): Mean Absolute Error Vs Number of Attributes

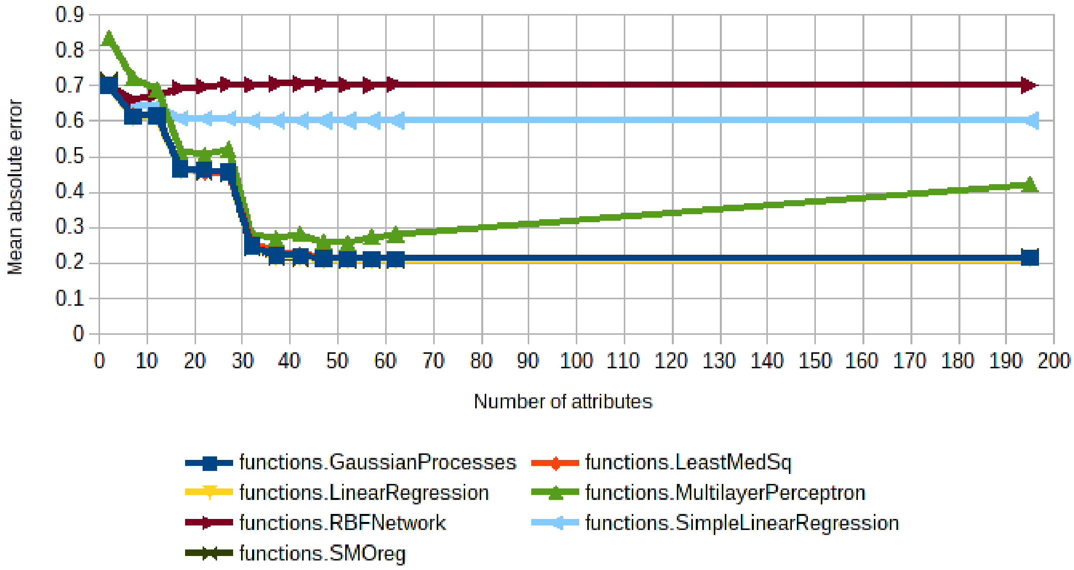

FIGURE C.12: Global Ranked Pruning (Goalkeepers): Function based algorithms -Mean Absolute Error Vs Number of Attributes

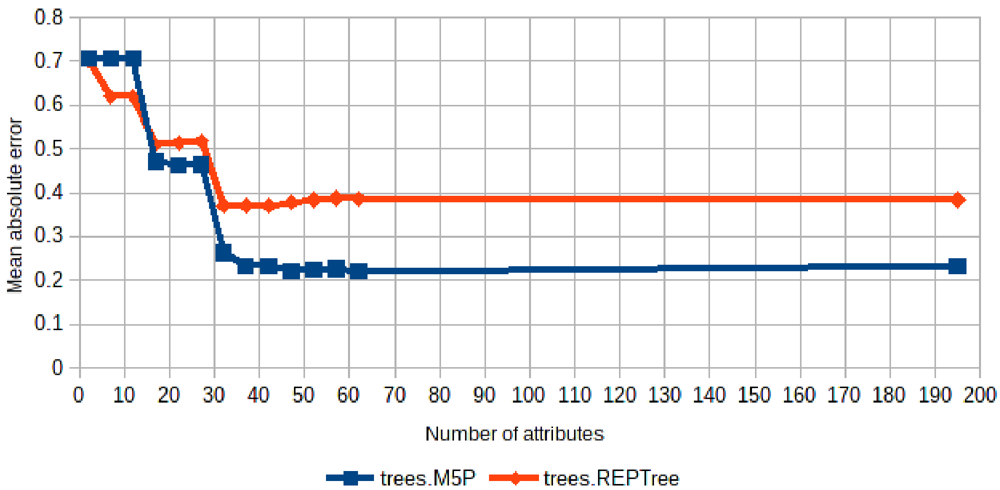

FIGURE C.13: Global Ranked Pruning (Goalkeepers): Tree based algorithms -Mean Absolute Error Vs Number of Attributes

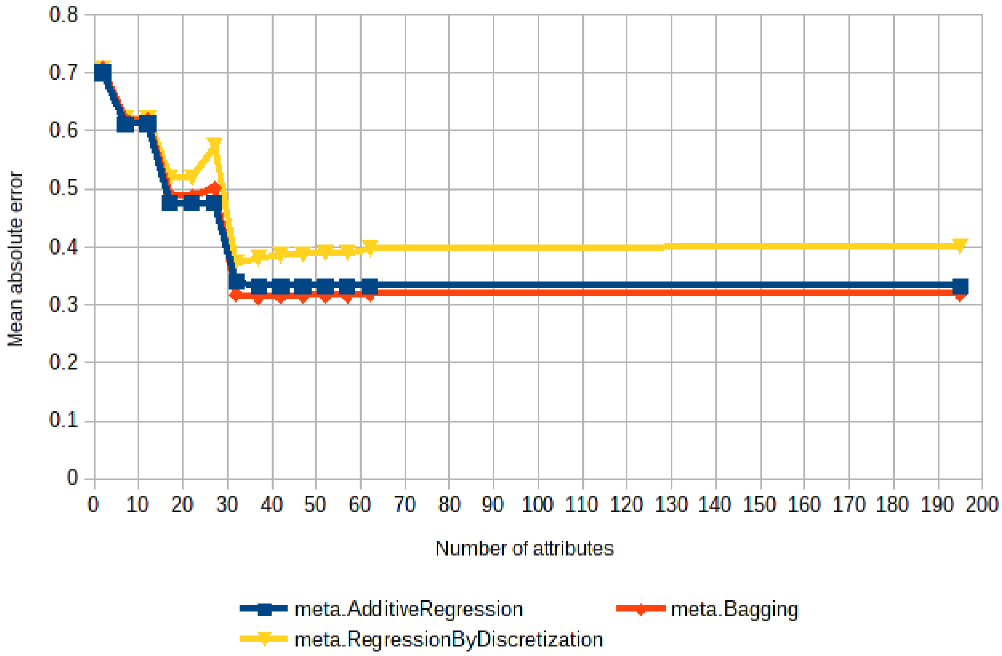

Figure C.14: Global Ranked Pruning (Goalkeepers): Meta algorithms -Mean Absolute Error Vs Number of Attributes

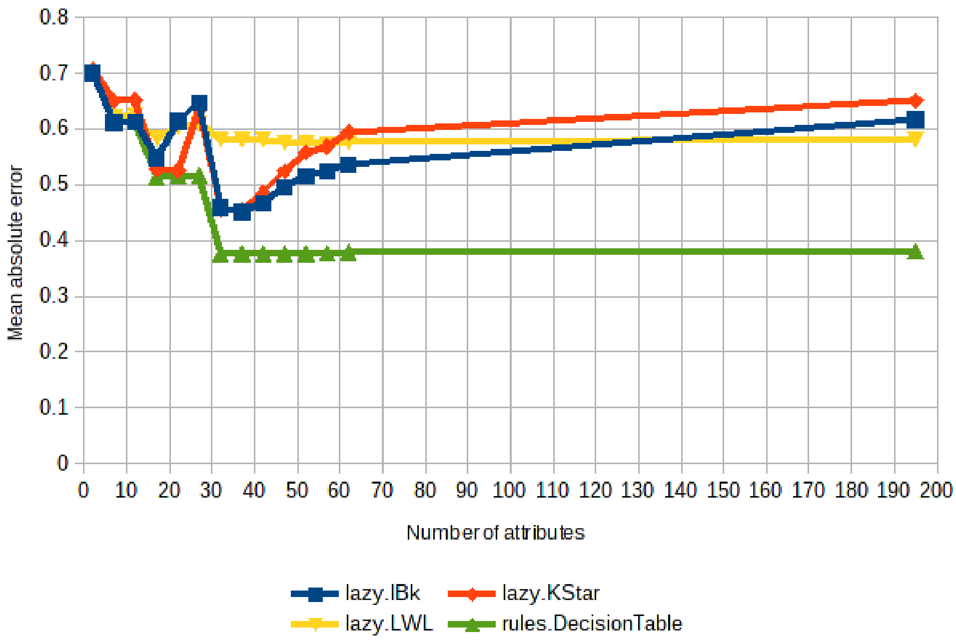

Figure C.15: Global Ranked Pruning (Goalkeepers): Lazy and Rule based algorithms -Mean Absolute Error Vs Number of Attributes

Table C.4: Iterative Local Pruning (Defenders): Linear Regression

NumOfAttrib	CC	MAE	RMSE	RelAE	BTime(s)	TTime(s)
195	0.8530	0.2728	0.3477	52.5298	14.7500	117.9580
95	0.8561	0.2700	0.3442	52.0044	0.1780	4.9200

Table C.5: Iterative Local Pruning (Defenders): LeastMedSq

NumOfAttrib	CC	MAE	RMSE	RelAE	BTime(s)	TTime(s)
195	0.8484	0.2766	0.3530	53.2780	33.4490	283.3330
174	0.8485	0.2766	0.3530	53.2734	29.5910	262.6080

Table C.6: Iterative Local Pruning (Defenders): M5P

NumOfAttrib	CC	MAE	RMSE	RelAE	BTime(s)	TTime(s)
195	0.8094	0.2989	0.3912	57.5652	12.6480	77.6500
69	0.7880	0.3122	0.4101	60.1254	3.8080	32.4830
63	0.7933	0.3085	0.4055	59.4149	3.3770	29.9020
61	0.7936	0.3086	0.4052	59.4281	3.2650	28.4730

Table C.7: Iterative Local Pruning (Defenders): REP Tree

NumOfAttrib	CC	MAE	RMSE	RelAE	BTime(s)	TTime(s)
195	0.6456	0.3935	0.5141	75.7797	1.4570	7.6540
43	0.6480	0.3939	0.5119	75.8534	0.2860	1.5220

Table C.8: Iterative Local Pruning (Goalkeepers): Linear Regression

NumOfAttrib	CC	MAE	RMSE	RelAE	BTime(s)	TTime(s)
195	0.9505	0.2126	0.2808	30.0825	1.0890	7.1390
63	0.9546	0.2071	0.2690	29.3006	0.0250	0.6610
62	0.9545	0.2071	0.2692	29.3101	0.0180	0.5810

Table C.9: Iterative Local Pruning (Goalkeepers): LeastMedSq

NumOfAttrib	CC	MAE	RMSE	RelAE	BTime(s)	TTime(s)
195	0.9484	0.2164	0.2861	30.6154	17.4200	126.4790
109	0.9484	0.2164	0.2861	30.6154	5.0480	42.6250

Table C.10: Iterative Local Pruning (Goalkeepers): M5P

NumOfAttrib	CC	MAE	RMSE	RelAE	BTime(s)	TTime(s)
195	0.9317	0.2330	0.3278	32.9767	2.2770	8.0710
38	0.9380	0.2232	0.3127	31.5775	0.3340	2.9320
35	0.9387	0.2231	0.3110	31.5669	0.2900	2.8260

Table C.11: Iterative Local Pruning (Goalkeepers): REP Tree

NumOfAttrib	CC	MAE	RMSE	RelAE	BTime(s)	TTime(s)
195	0.8134	0.3829	0.5275	54.1887	0.6490	1.4680
26	0.8217	0.3730	0.5165	52.7783	0.0150	0.1890

Table C.12: Iterative Local Pruning (Midfielders): Linear Regression

NumOfAttrib	CC	MAE	RMSE	RelAE	BTime(s)	TTime(s)
195	0.9504	0.1729	0.2356	29.2375	22.0670	150.3010
127	0.9507	0.1722	0.2347	29.1190	0.4190	12.6130

Table C.13: Iterative Local Pruning (Midfielders): LeastMedSq

NumOfAttrib	CC	MAE	RMSE	RelAE	BTime(s)	TTime(s)
195	0.9493	0.1764	0.2401	29.8332	36.3500	327.6870
178	0.9493	0.1764	0.2401	29.8332	34.0980	306.9560

Table C.14: Iterative Local Pruning (Midfielders): M5P

NumOfAttrib	CC	MAE	RMSE	RelAE	BTime(s)	TTime(s)
195	0.9310	0.1962	0.2764	33.1745	19.3820	149.5270
100	0.9367	0.1901	0.2650	32.1532	8.2800	74.4730
96	0.9368	0.1900	0.2649	32.1264	7.6080	70.9200

Table C.15: Iterative Local Pruning (Midfielders): REP Tree

NumOfAttrib	CC	MAE	RMSE	RelAE	BTime(s)	TTime(s)
195	0.8194	0.3232	0.4358	54.6480	2.1000	8.5080
75	0.8202	0.3232	0.4349	54.6511	0.3720	3.2790

Table C.16: Threshold Pruning (Defenders): Linear Regression (Threshold = mean - SD)

NumOfAttrib	CC	MAE	RMSE	RelAE	BTime(s)	TTime(s)	Threshold
195	0.8530	0.2728	0.3477	52.5298	13.9140	115.0140	0.0583
32	0.6482	0.3813	0.5071	73.4250	0.1080	1.0420	0.0618
18	0.6379	0.3832	0.5129	73.7993	0.0140	0.2100	0.0866
16	0.6085	0.4019	0.5285	77.4071	0.0140	0.2280	0.1256
14	0.6005	0.4052	0.5325	78.0437	0.0120	0.1470	0.1688
11	0.4723	0.4595	0.5870	88.4883	0.0090	0.1170	0.2903
10	0.4716	0.4596	0.5873	88.5209	0.0080	0.1120	0.3140
9	0.4721	0.4596	0.5871	88.5073	0.0080	0.1080	0.3331
8	0.4485	0.4656	0.5952	89.6732	0.0070	0.1020	0.3502
8	0.4485	0.4656	0.5952	89.6732	0.0070	0.0970	0.3502

Table C.17: Threshold Pruning (Defenders): Linear Regression (Threshold = 0.3 x SD)

NumOfAttrib	CC	MAE	RMSE	RelAE	BTime(s)	TTime(s)	Threshold
195	0.8530	0.2728	0.3477	52.5298	13.8510	114.8110	0.0765
27	0.5181	0.4431	0.5697	85.3369	0.0690	0.7180	0.0743
14	0.5157	0.4427	0.5706	85.2566	0.0120	0.1910	0.0588
14	0.5157	0.4427	0.5706	85.2566	0.0100	0.3300	0.0588

Table C.18: Threshold Pruning (Goalkeepers): Linear Regression (Threshold = mean - SD)

NumOfAttrib	CC	MAE	RMSE	RelAE	BTime(s)	TTime(s)	Threshold
195	0.9505	0.2126	0.2808	30.0825	1.1340	7.2250	0.0454
26	0.6112	0.5551	0.7144	78.5456	0.0280	0.2280	0.1173
6	0.5667	0.5718	0.7433	80.9062	0.0020	0.0580	0.0654
6	0.5667	0.5718	0.7433	80.9062	0.0010	0.0530	0.0654

Table C.19: Threshold Pruning (Goalkeepers): Linear Regression (Threshold = 0.3 X SD)

NumOfAttrib	CC	MAE	RMSE	RelAE	BTime(s)	TTime(s)	Threshold
195	0.9505	0.2126	0.2808	30.0825	1.0970	6.8600	0.0805
20	0.6117	0.5557	0.7138	78.6310	0.0130	0.1620	0.1467
4	0.5703	0.5796	0.7410	82.0188	0.0020	0.0590	0.1622
4	0.5703	0.5796	0.7410	82.0188	0.0020	0.0590	0.1622

Table C.20: Threshold Pruning (Midfielders): Linear Regression (Threshold = mean - SD)

NumOfAttrib	CC	MAE	RMSE	RelAE	BTime(s)	TTime(s)	Threshold
195	0.9504	0.1729	0.2356	29.2375	15.9650	144.9700	0.1473
13	0.6889	0.4340	0.5486	73.3887	0.0180	0.2690	0.3025
11	0.6790	0.4416	0.5557	74.6835	0.0180	0.3310	0.3933
8	0.6788	0.4417	0.5558	74.6978	0.0120	0.2020	0.7057
7	0.6739	0.4446	0.5592	75.1928	0.0100	0.1380	0.8016
6	0.6672	0.4486	0.5638	75.8702	0.0080	0.1310	0.8963
5	0.6644	0.4501	0.5657	76.1078	0.0090	0.1120	1.0096
4	0.5376	0.5057	0.6382	85.5130	0.0120	0.1290	1.2382
3	0.4466	0.5325	0.6772	90.0568	0.0070	0.1360	1.2308
3	0.4466	0.5325	0.6772	90.0568	0.0070	0.0880	1.2308

Table C.21: Threshold Pruning (Midfielders): Linear Regression (Threshold = 0.3 X SD)

NumOfAttrib	CC	MAE	RMSE	RelAE	BTime(s)	TTime(s)	Threshold
195	0.9504	0.1729	0.2356	29.2375	14.7490	129.0060	0.1068
14	0.7032	0.4240	0.5381	71.6937	0.0190	0.2740	0.0628
13	0.7032	0.4240	0.5381	71.6937	0.0140	0.2930	0.0628
13	0.7032	0.4240	0.5381	71.6937	0.0260	0.2300	0.0628

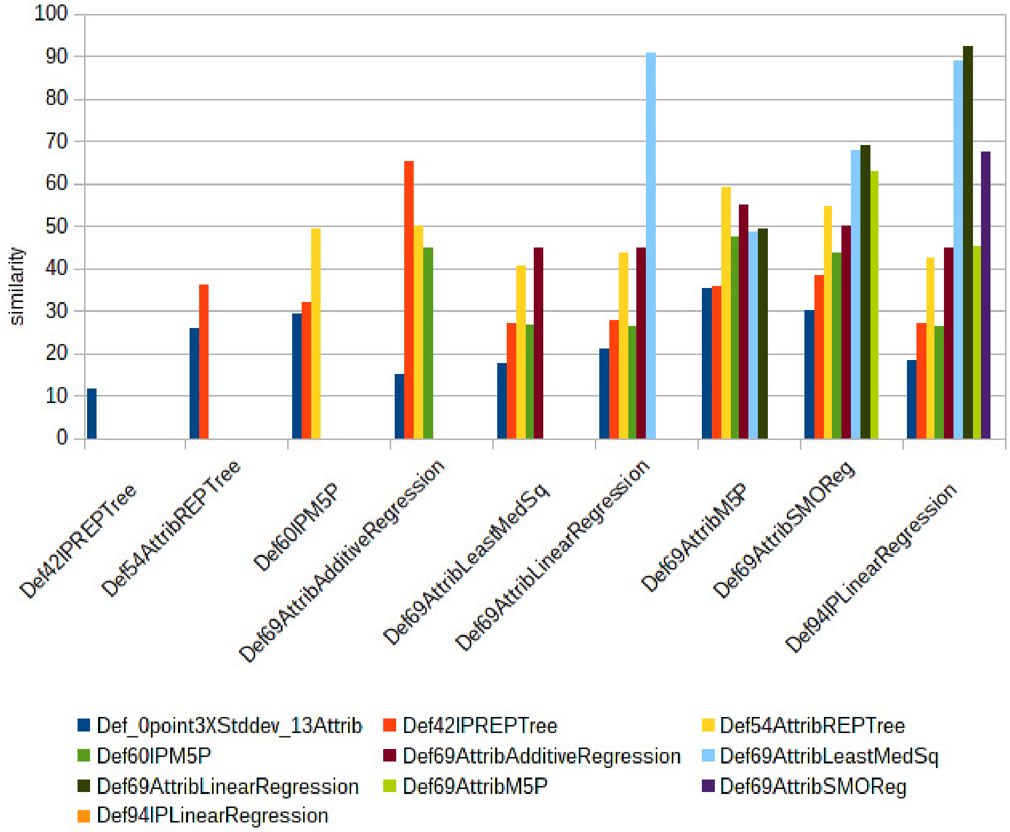

FIGURE C.16: Similarity between optimal lists for defenders

Table C.22: Legend for list of optimal defenders attribute lists

Name of Defender List	Algorithm used	NumOfAtt	Pruning
Def_0point3XStddev_13Attrib	Linear Regression	13	Threshold
Def42IPREPTree	REPTree	42	Iterative Local
Def54AttribREPTree	REPTree	54	Global Ranked
Def60IPM5P	M5P	60	Iterative Local
Def69AttribAdditiveRegression	AdditiveRegression	69	Global Ranked
Def69AttribLeastMedSq	LeastmedSq	69	Global Ranked
Def69AttribLinearRegression	Linear Regression	69	Global Ranked
Def69AttribM5P	M5P	69	Global Ranked
Def69AttribSMOReg	SMOreg	69	Global Ranked
Def94IPLinearRegression	Linear Regression	94	Iterative Local

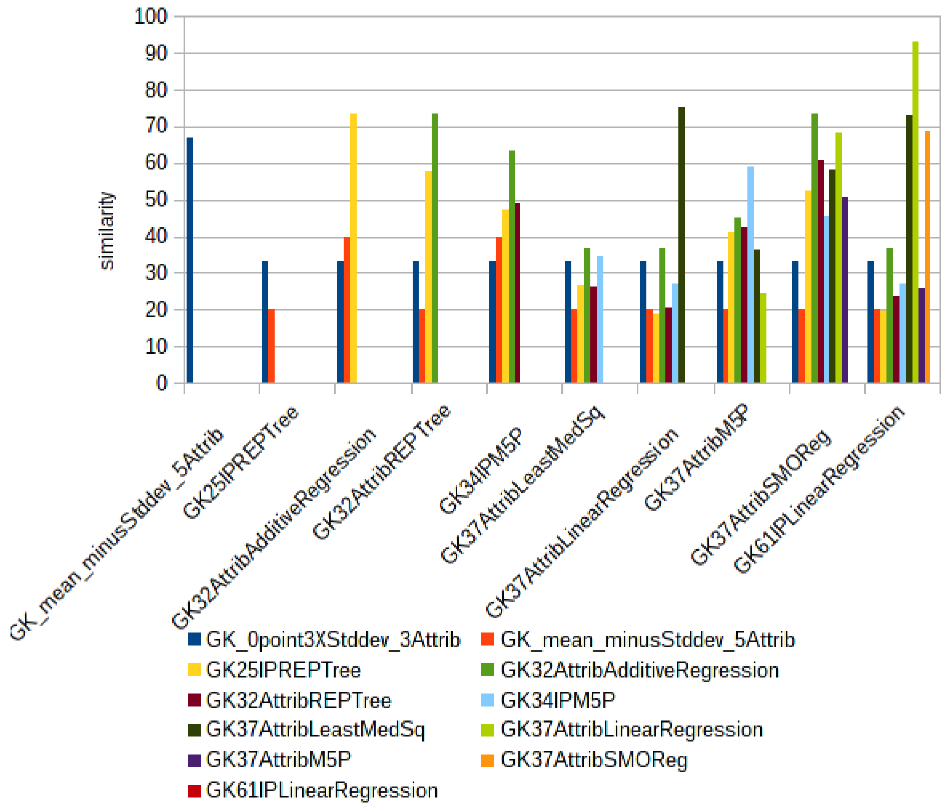

Figure C.17: Similarity between optimal lists for goalkeepers

TABLE C.23: Legend for list of optimal goalkeepers attribute lists

Name of Goalkeeper List	Algorithm used	NumOfAtt	Pruning
GK_mean_minusStddev_5Attrib	Linear Regression	5	Threshold
GK25IPREPTree	REPTree	25	Iterative Local
GK32AttribAdditiveRegression	AdditiveRegression	32	Global Ranked
GK32AttribREPTree	REPTree	32	Global Ranked
GK34IPM5P	M5P	34	Iterative Local
GK37AttribLeastMedSq	LeastmedSq	37	Global Ranked
GK37AttribLinearRegression	Linear Regression	37	Global Ranked
GK37AttribM5P	M5P	37	Global Ranked
GK37AttribSMOReg	SMOreg	37	Global Ranked
GK61IPLinearRegression	Linear Regression	61	Iterative Local

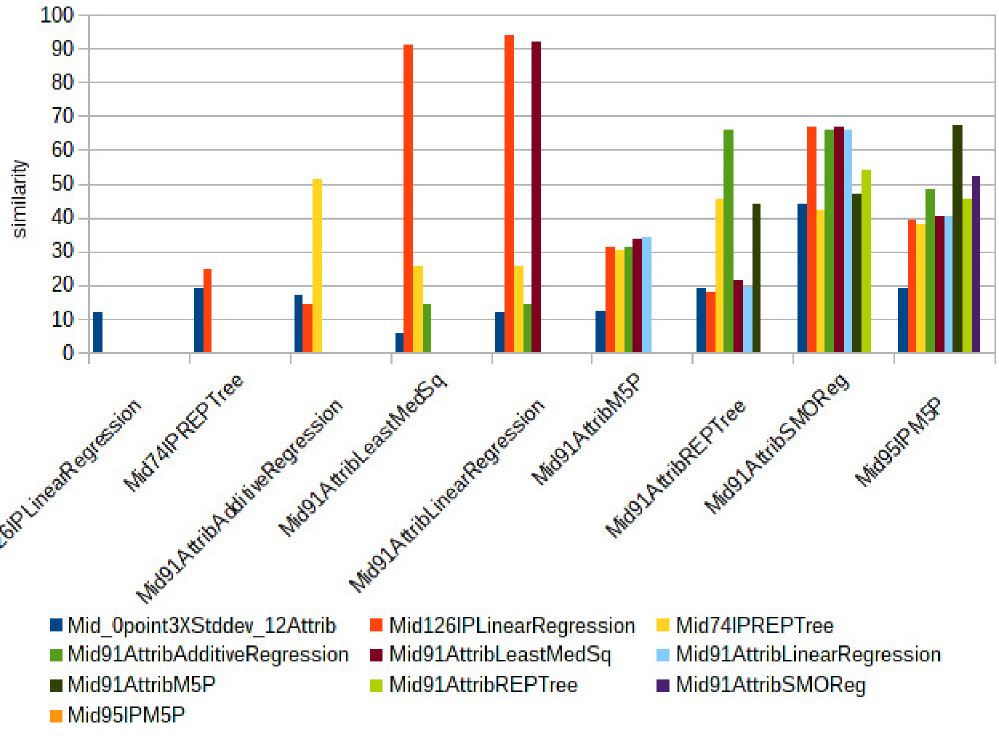

FIGURE C.18: Similarity between optimal lists for midfielders

Table C.24: Legend for List of optimal midfielders attribute lists

Name of Midfielder List	Algorithm used	NumOfAtt	Pruning
Mid__0point3XStddev__12Attrib	Linear Regression	5	Threshold
Mid126IPLinearRegression	Linear Regression	25	Iterative Local
Mid74IPREPTree	REPTree	32	Iterative Local
Mid91AttribAdditiveRegression	AdditiveRegression	32	Global Ranked
Mid91AttribLeastMedSq	LeastmedSq	34	Global Ranked
Mid91AttribLinearRegression	Linear Regression	37	Global Ranked
Mid91AttribM5P	M5P	37	Global Ranked
Mid91AttribREPTree	REPTree	37	Global Ranked
Mid91AttribSMOReg	SMOreg	37	Global Ranked
Mid95IPM5P	Linear Regression	61	Iterative Local

C.2 Results from Chapter 3

TABLE C.25: Algorithms performance with selected attributes (threshold=0.0277, number of attributes=59) on matchdata

Algorithm	% Correctly classified	ROC-Area	F-Measure	Kappa
MultiLayer Perceptron	63.9474	0.805	0.639	0.4406
FT	61.8421	0.713	0.619	0.4068
SMO	65.2632	0.754	0.642	0.4535
NaiveBayes	57.8947	0.778	0.587	0.3608
RandomForest	60	0.77	0.586	0.367
DecisionTable	58.1579	0.714	0.56	0.3317
FURIA	62.3684	0.712	0.588	0.3824
J48Graft	56.8421	0.656	0.563	0.3251
J48	54.2105	0.634	0.541	0.2883
JRip	57.1053	0.654	0.537	0.2922
REP Tree	55.2632	0.659	0.514	0.2769
LibSVM	45	0.5	0.279	0.3667
KStar	54.7368	0.72	0.55	0.3025

TABLE C.26: Algorithms performance with selected attributes (threshold=0.05, number of attributes=48) on matchdata

Algorithm	% Correctly classified	ROC-Area	F-Measure	Kappa
MultiLayer Perceptron	63.6842	0.803	0.638	0.4368
FT	56.3158	0.672	0.564	0.3226
SMO	64.2105	0.628	0.628	0.4317
NaiveBayes	56.0526	0.769	0.57	0.3348
RandomForest	57.3684	0.744	0.558	0.3214
DecisionTable	57.6316	0.711	0.555	0.3225
FURIA	57.3684	0.686	0.498	0.498
J48Graft	55	0.634	0.549	0.3001
J48	53.4211	0.628	0.537	0.2806
JRip	52.3684	0.627	0.469	0.2085
REP Tree	55	0.654	0.506	0.2714
LibSVM	45	0.5	0.279	0
KStar	52.8947	0.533	0.533	0.533

Table C.27: Predicting outcome with 27 rating attributes

	%Correctly classified	ROC Area	F-Measure	Kappa
SMO	89.7368	0.934	0.897	0.8401
LibSVM	88.6842	0.913	0.886	0.8235
BaggingFT	87.3684	0.971	0.872	0.8032
AdaFT	87.1053	0.956	0.87	0.7994
DaggingSMO	87.1053	0.959	0.959	0.7993
FT	86.5789	0.952	0.867	0.792
RBFNetwork	86.5789	0.953	0.865	0.7917
RandomForest	85.7895	0.955	0.856	0.7795
ClassificationViaRegression M5P	85.2632	0.958	0.852	0.7713
Ada LibSVM	85	0.851	0.939	0.7675
trees.J48	84.2105	0.89	0.841	0.7542
NaiveBayes	83.6842	0.964	0.845	0.7525
RandomCommittee RandomTree	83.9474	0.95	0.839	0.7512
J48graft	83.4211	0.887	0.831	0.7412
FURIA	83.1579	0.911	0.822	0.7341
REPTree	80.7895	0.971	0.802	0.6983
JRip	78.9474	0.88	0.79	0.6727
RandomTree	78.4211	0.842	0.785	0.6671
KStar	75.2632	0.914	0.757	0.6167
Stacking FT	45	0.5	0.279	0

Bibliography

R. Bate. Football chance: tactics and strategy. In T. Reilly, A. Lees, K. Davids, and J. Murphy, editors, *Science and Football*, pages 293–301. London: E & FN SPON., 1988.

D. Bunker and R. Thorpe. A model for teaching games in the secondary school. *Bulletin of Physical Education*, 18(1)(5-8).

C. Collet. The possession game? a comparative analysis of ball retention and team success in european and international football, 2007-2010. *Journal of Sports Sciences*, pages 1–14, 2012.

A. C. Constantinou, N. E. Fenton, and M. Neil. pi-football: A bayesian network model for forecasting association football match outcomes. *Knowledge-Based Systems*, 36:322-339, 2012.

M. Dixon and S. Coles. Modelling association football scores and inefficienies in the football betting markets. *Applied Statistics*, 46(2):265–280, 1997.

M. Ellis. Similarities and differences in games: a system for classification. *AEISE*, 1983.

E. Frank, Y. Wang, S. Inglis, G. Holmes, and I. H. Witten. Using model trees for classification. *Machine Learning*, 32(1), July 1998.

I. Franks. Analysis of association football. *Soccer Journal*, 33(5):35–43, 1988.

J. Goddard. Regression models for forecasting goals and match results in association football. *International Journal of Forecasting*, 21:331-340, 2005.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explorations*, 11(1), 2009.

T. Hooper. Teaching games for understanding using progressive principles of play. *CAHPERD*, 64:4–7, 1998.

C. Hughes. *The winning formula*. William Collins Sons & Co Ltd., 1990.

M. Hughes and S. Churchill. Attacking profiles of successful and unsuccessful teams in copa america 2001. *Journal of Sports Sciences*, 22:505, 2004.

M. Hughes and I. Franks. Analysis of passing sequences, shots and goals in soccer. *Journal of Sports Sciences*, 23:509–514, 2005.

M. Hughes and I. Franks. *The Essentials of Performance Analysis: An Introduction*. Routledge, 2007.

M. Hughes and I. M. Franks. *Notational analysis of sport*. London: E & FNSpon, first edition, 1997.

M. Hughes and I. M. Franks. *Notational analysis of sport*. London & New York: Routledge Taylor & Francis Group., second edition, 2004.

M. Hughes and N. Snook. Effectiveness of attacking play in the 2004 european championships. volume World Congress of Performance Analysis of Sport 7, pages 46–62. Cardiff: CPA UWIC Press, 2006.

M. Hughes, K. Robertson, and A. Nicholson. *Comparison of patterns of play of successful and unsuccessful teams in the 1986 World Cup for soccer*. London: E & FN SPON, in t. reilly, a. lees, k.davids and w. j. murphy (eds.) edition, 1988. Science and Football (pp. 363-367).

M. D. Hughes and R. M. Bartlett. The use of performance indicators in performance analysis. *Journal of Sports Sciences*, 20:739–754, 2002.

D. Karlis and N. I. Statistical modelling for soccer games. In *Proceedings of HERCMA 98, Athens*, pages 541–548, 1998.

D. Karlis and I. Ntzoufras. Analysis of sports data by using bivariate poisson models. *The Statistician*, 52(3):381–393, 2003.

A. Lee, J. Modeling scores in the premier league: is manchester united really the best? *Chance*, 10:15–19, 1997.

M. Lewis. *Moneyball: The Art of Winning an Unfair Game*. W.W. Norton & Company Inc., 2003.

F. Liu and Z. Zhang. Predicting soccer league games using multinomial logistic models. Course Project, 2008.

M. Maher, J. Modelling association football scores. *Statististica Neerlandica*, 36: 109–118, 1982.

T. McGarry and I. M. Franks. The science of match analysis. In T. Reilly and A. M. Williams, editors, *Science and Football*, pages 265–275. London & New York: Routledge Taylor & Francis Group, 2003.

D. Morris. *The Soccer Tribe*. Jonathan Cape, 1981.

F. Mosteller. A resistant analysis of 1971 and 1972 professional football. In H. J. Goldstein, editor, *Sports, Games & Play*, pages 371–401. New Jersey: Lawrence Erlbaum Associates, 1979.

E. Olsen and O. Larsen. Use of match analysis by coaches. In T. Reilly, J. Bangsbo, and M. Hughes, editors, *Science and Football*, volume III, pages 209–220. London: E & FN SPON., 1988.

R. Pollard, C. Reep, and S. Hartley. The quantitative comparison of playing styles in soccer. In T. Reilly, A. Lees, K. Davids, and W. Murphy, J, editors, *Science and Football*, pages 309–315. London: E & FN SPON., 1988.

M. C. Purucker. Neural network quarterbacking. *IEEE Potentials*, 15(3):9-15, 1996.

C. Reep and B. Benjamin. Skill and chance in association football. *Journal of Royal Statistical Society*, Series A(131):581–585, 1968.

A. Rotshtein, P, M. Posner, and A. Rakityanskaya, B. Football predictions based on a fuzzy model with genetic and neural tuning.

H. Rue and Øyvind Slavensen. Predictive and retrospective analysis of soccer matches in a league. Technical report, Department of Mathematical Sciences, NTNU, Norway, 1998.

A. Tenga. *Reliability and validity of match performance analysis in soccer : a multi-dimensional qualitative evaluation of opponent interaction*. PhD thesis, Norwegian School of Sport Sciences, 2010.

P. Werner and L. Almond. Models of games education. *JOPERD*.

Wikipedia. Fantasy football (association) — wikipedia, the free encyclopedia, 2013a. URL http://en.wikipedia.org/w/index.php?title=Fantasy_football_(Association)&oldid=567412407. [Online; accessed 25-August-2013].

Wikipedia. 2011-12 premier league — wikipedia, the free encyclopedia, 2013b. URL http://en.wikipedia.org/w/index.php?title=2011%E2%80%9312_Premier_League&oldid=569272806.

Wikipedia. Sabermetrics — wikipedia, the free encyclopedia, 2013c. URL http://en.wikipedia.org/w/index.php?title=Sabermetrics&oldid=569006358.

Wikipedia, 2013d. URL http://en.wikipedia.org/wiki/. [Online; accessed 25-August-2013].

Master thesis filing card

Student: Gunjan Kumar

Title: Machine Learning for Soccer Analytics

Dutch title: Automatisch leren voor voetbalanalytiek

UDC: 681.3*I20

Abstract:
Sports analytics has been successfully applied in sports like baseball and basketball. However, its application in soccer has been limited. There is a need to find out if the application of Machine Learning can bring better and more insightful results in soccer analytics. In this thesis, we perform descriptive as well as predictive analysis of soccer matches and player performances. In soccer, it is popular to rely on ratings by experts to assess a player's performance. However, the experts do not unravel the criteria they use for their rating. We attempt to identify the most important attributes of player's performance which determine the expert ratings. We performed a series of classifications with three different pruning strategies and an array of Machine Learning algorithms. We obtained a list of most important performance metrics for each of the four playing positions which approximates the attributes considered by the experts while assigning ratings. Then we find the most influential performance metrics of the players for determining the match outcome and we examine the extent to which the outcome is characterised by the performance attributes of the players. We found 34 performance attributes using which we can predict the match outcome with an accuracy of 63.4%. Next, we devise a method to aggregate individual player ratings to produce a set of team ratings and investigate how closely these team ratings can determine the match outcome. We create 27 different team rating attributes, and using 8 of these attributes aggregated over the player ratings for current match we obtained a prediction accuracy of 90%. This indicates that the expert ratings are influenced by the match outcome. Then we use a weighted average method to aggregate team ratings over past match performances. We find that match outcomes are best characterised by the ratings for current match. Next, we investigate how well the expert ratings for the past performances of players predict the next match outcome. We find that we could only manage a prediction accuracy of 53.4% because of the unpredictability of 'draw'. We conclude that the outcome of a match is strongly correlated to the ratings for the current match.

Thesis submitted for the degree of Master of Science in Artificial Intelligence, option Engineering and Computer Science

Thesis supervisor: Prof. dr. Jesse Davis

Assessors: Prof. dr. ir. Gerda Janssens
ir. Jan Van Haaren

Mentors: Daniele Alfarone
ir. Jan Van Haaren