UTRECHT UNIVERSITY

MASTER'S THESIS

Using Data Analytics to Make the Scouting and Training of Sports Talents More Effective

Author:
Chantal BLOM
Student number:
6312551

Utrecht University
1st supervisor:
prof. dr. A.P.J.M. (Arno) SIEBES
2nd supervisor:
Dr. ing. G.M. (Georg) KREMPL

Business Data Challengers
Daily supervisor:
Marten PENNINGA

Nederlands Handbal Verbond
Personal contact:
Edwin KIPPERS

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

Program: Master Business Informatics
Department: Information Sciences

July 15, 2019
Version 2.0

UTRECHT UNIVERSITY

Abstract

Faculty of Science
Information Sciences

Master of Science

Using Data Analytics to Make the Scouting and Training of Sports Talents More Effective

by Chantal BLOM

This research proposes methods to get insights from limited data from sports talents. Since the data is limited, the focus is on comparing talents. Data from Dutch handball talents is used as a case to create and test the methods. The research covers three main phases; data preparation, data analysis and data visualization. Substantiated with theory from the literature, we outline procedures for every phase. For the first phase we propose a way to estimate missing values by using multiple linear regression in combination with clustering. In the second phase, we propose a nearest neighbors regression approach to find the best distance range to compare talents. In the last phase, we show a way of visualizing comparable talents using the spider plot, to present insights to sports scouts and coaches. Based on the results from the methods we tested with the handball data, we conclude that the approach from phase 1 works sufficiently in the case we tested, but that it does not necessarily work for other cases or other variables due to limited data, which is a limitation. Furthermore, we consider both approaches in phases 2 and 3 applicable in the handball case. However, to improve the reliability, the methods could be tested more extensively with other data in a future research.

Acknowledgements

During the research I have encountered great support from different people. I would like to thank them in this section.

First of all, I would like to thank my first supervisor prof. dr. Arno Siebes for supporting me during the last eight months and giving valuable feedback. I would also like to acknowledge my second supervisor Dr. ing. Georg Krempl for providing me with valuable feedback near the end of the research.

Secondly, I would like to thank Business Data Challengers for the opportunity to do this research and especially my daily supervisor Marten Penninga for also sending me in the right directions and giving valuable feedback. Moreover, the Dutch Handball Federation and especially Edwin Kippers gave me the opportunity to do this research as well, by providing me with the data and useful knowledge. Therefore, I thank them too.

Last but not least, I would like to thank my family for always supporting me and always believing in me.

Chantal Blom - July 2019

Contents

List of Figures

List of Tables

List of Abbreviations

NHV	**N**ederlands **H**andbal **V**erbond
TVS	**T**alent**V**olg**S**ysteem
VIF	**V**ariance **I**nflation **F**actor
kNN	**k**-**N**earest **N**eighbors
MAE	**M**ean **A**bsolute **E**rror
RMSE	**R**oot **M**ean **S**quare **E**rror
RAE	**R**elative **A**bsolute **E**rror
RSE	**R**elative **S**quared **E**rror

Chapter 1

Introduction

1.1 Research context

Due to the great results of the Dutch women's handball selection in the last five years (semi-finals and finals in the European and World Championships and semi-finals in the Olympic Games in 2016), the Dutch Handball Federation (NHV) is growing (Waterval, 2017). Furthermore, the Dutch men's selection qualified themselves for the European Championships for the very first time in 2020 (HandbalNL, 2019). These successes are also due to the HandbalAcademie that the NHV has set up in 2006. The aim of the HandbalAcademie is to prepare handball talents to start playing for the national A-selection of The Netherlands, the national youth selections and main European teams.

For individual athletes from the aforementioned Dutch teams, data is collected. Examples of the collected data are results from physical tests like sprinting and jumping tests and data about the mental condition of the talents. Most of this data is collected at the location of the NHV[1].

The NHV is looking for a way to make the scouting and training of talents more effective by using data analytics. Currently, the federation does not use the information of current or former talents when looking for new talents. Therefore, the focus in this research project is on finding a way to help the scouts and coaches use the available data to make their daily activities easier.

1.2 Problem statement

As mentioned, quite some data is collected and the NHV would like to use this data more effectively. The main challenge is however, that data is not always collected consistently, meaning that not all talents performed the same or all physical tests and that not all tests are performed at the same moment during the year for example. Between different handball teams, there are differences in what kind of data they do and do not collect. The result of this is that the data has many missing values. Furthermore, results of tests of the men are gathered even less consistently. This means that there is imbalance of men and women in the data as well.

[1]Sportcentrum Papendal, Papendallaan 9, 6816 VD, Arnhem

1.3 Research objective and scope

With the challenge described in the previous section, the goal of this research is to find a method to handle the limited data to still be able to give relevant advice to the Dutch Handball Federation. Although the data is limited, there are still numerous ways of giving advice by using the data. For that reason, the scope of the research is set to comparing talents. With comparing talents, we mean both comparing young players with current talents as well as comparing talents within the same group or team. Therefore, the specific research objective is to find a method to compare talents to be able to give advice on talent selection and training, with the limited data.

1.4 Relevance

The relevance of this research can be split in two parts. On the one hand, it could be relevant in data mining. An approach in which small data is sufficient to still be able to give relevant advice could be used in different domains where data is limited. On the other hand, it could be relevant in sports and talent development more specifically. Small or upcoming sports, where data has not been collected consistently and for a long time yet, could benefit from this research.

1.5 Thesis structure

This section explains what the structure of this thesis looks like. Chapter 2 discusses the research approach. In this chapter, we will mention the research questions, the main steps we took and we will provide details about which tools we used. Chapter 3 gives a theoretical background of the techniques we used throughout the research. We will discuss the algorithms, k-fold cross validation and we will give information about relevant other studies on sport data and talent development. In Chapter 4 we will provide more information about the data we received from the Dutch Handball Federation and we will explain what we did with the data before we could compare the talents. Chapter 5 discusses the method that can be used to find the best models to compare talents. In this chapter we will also give the results of applying the method to the handball data. In Chapter 6 we will describe possibilities of visualizing the results obtained when the best model of Chapter 5 is used to compare talents. Lastly, Chapter 7 gives a conclusion and discusses the limitations and possible future research of this study.

Chapter 2

Research approach

This Chapter mentions the research question in Section 2.1. Furthermore, in Section 2.2, the main phases of this study and the tools we used throughout the research will be discussed.

2.1 Research question

As mentioned in Chapter 1, the Dutch Handball Federation is growing and they are looking for ways to use their data. We also mentioned that the scope of this research is set to comparing talents. Therefore, the research question is the following:

How can data analytics be used to assist scouts and coaches in finding and training sports talents in order to make more effective choices?

In the previous chapter we also stated the main problem. The data is limited and, therefore, the focus is on how we can answer the research question with this kind of data.

The main research question will be answered by answering the following four sub-questions:

- **Sub-question 1**: What data pre-processing needs to be done before the data analytics process can start?
- **Sub-question 2**: How can scouts be supported when looking for the best performing sports players?
- **Sub-question 3**: How can coaches be supported when training their sports talents?
- **Sub-question 4**: Which way of presenting the information to the scouts and coaches will be effective?

Sub-question 1 is the approach that needs to be taken before valuable information can be found in the data. Sub-question 2 and 3 are about how valuable information can be found in the data. By answering the fourth sub-question we will get a way of presenting the valuable information to the different stakeholders.

2.2 Research methods

This section describes the research methods. First, we will discuss the main phases of this research in Section 2.2.1. Second, we will give details about the tools used during the research in Section 2.2.2.

2.2.1 The main phases

This section describes the main phases that this research was divided into to answer the research question. In the main phases, we will answer the different sub-question as outlined in Section 2.1. Figure 2.1 illustrates the flow of the three main phases. The figure also shows that throughout all steps in the research, choices were based on theoretical background knowledge gathered during literature reviews. During the project we had contact moments with the domain expert[1] once in the two or three weeks to discuss and validate results. The motivation of this research came from the Dutch Handball Federation. However, we consider the methods we proposed in the different phases to be applicable in other upcoming sports and other domains where talents can be compared. Therefore, we consider this research as a handball case study to answer the research question. The next paragraphs give an overview of the phases.

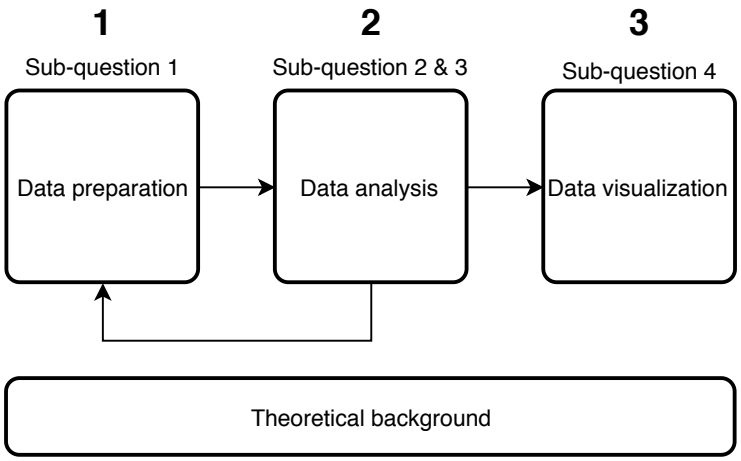

FIGURE 2.1: The main phases of the research.

Phase 1 - data preparation

The first main phase is data preparation and this phase is related to sub-question 1. In the beginning of this phase, we received the data from the Dutch Handball Federation from their talent tracking system. An important part of this phase was to inspect the data well together with the domain expert, to be able to select meaningful variables, to transform these variables and to fill the missing values that were left. There were quite some missing values for a certain variable. Therefore, we decided to handle these using predictive analytics instead of applying simple mean imputation for example. We tried many different regression models and chose the best one, meaning we performed exhaustive search. Although exhaustive search might not be the most efficient way of finding the best model, we considered it to be applicable in this case since the data set was not that big and the goal was not to find the fastest way of finding the best model. The results of these models were validated with the domain expert. What we did exactly during this phase is described in Chapter 4.

Phase 2 - data analysis

The second main phase is data analysis and this phase is related to sub-questions 2 and 3. During this phase we performed exhaustive search again to find the best

[1] Edwin Kippers, he is the assistant national coach of the Dutch men's handball selection.

model. Again we considered this applicable in this research since the goal was not to find an efficient way of finding the best model. We used a k-nearest neighbors regression approach to find the best performing model to compare talents. However, instead of looking at the k neighbors, we looked at the neighbors in a certain distance range. We considered this to be more applicable with the handball data and for this research, since we are focusing on limited data. Data points that are within the k number of neighbors can still be quite different if the data point is an outlier. Therefore, we tried to overcome this by looking at neighbors within a distance range only. To find the best performing model, we proposed to look at the five or ten best performing models on the training data and apply this model on the test data to find the best and most constant performing model. Since a big model base is created during this phase, we also propose to use this model base in another way to try to predict separate variables. What we did exactly during this phase and which algorithms we used is described in Chapter 5.

Phase 3 - data visualization

The third main phase is data visualization and this phase is related to sub-question 4. To find a way of visualizing the kind of data we got from phase 1 and 2 (multidimensional data), we looked into other studies. We found a visualization called the spider plot. A plot like this is applicable in different ways to give scouts and coaches insights about their talents. What we did exactly during this phase is described in Chapter 6.

2.2.2 The software

In this research we used the *R* programming language in all three phases. The *R*-version we used was 3.5.1, and we used RStudio (version 1.1.463) as the development environment. There are many packages available that facilitate data analytics in *R*. Table A.1 in Appendix A lists the packages that were used in this research and explains for which purposes they were used.

Chapter 3

Background

In this chapter we will first give a theoretical background of the machine learning algorithms we used throughout the research (Section 3.1) and about *k*-fold cross validation (Section 3.2). After that, we will provide some information about relevant other studies on sport data and talent development (Section 3.3).

3.1 Machine learning algorithms

Several machine learning algorithms are used in this research. We focus on two different types; supervised learning algorithms and unsupervised learning algorithms. Figure 3.1 shows the different types and algorithms used in this research[1]. Section 3.1.1 discusses supervised learning and describes the specific algorithms used in this research. Section 3.1.2 discusses unsupervised learning and describes the unsupervised algorithm used in this research.

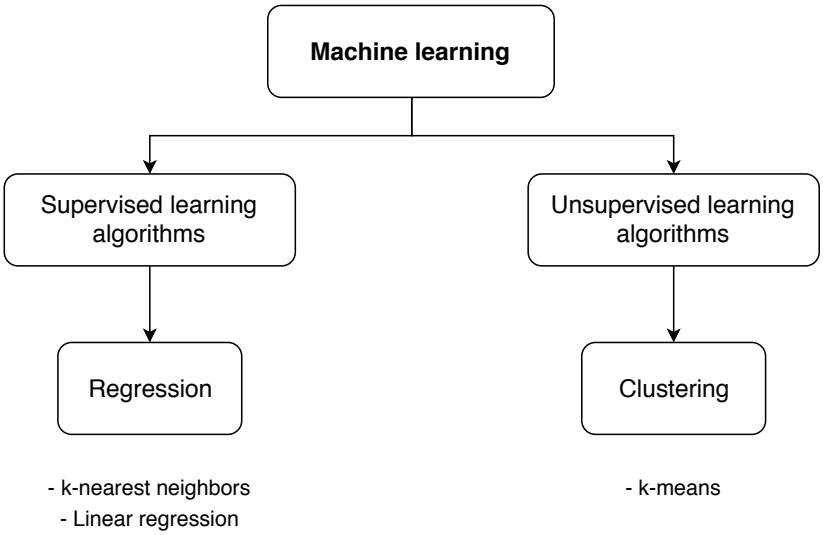

FIGURE 3.1: The types of algorithms used during the research.

3.1.1 Supervised learning

The first category of algorithms we discuss is supervised learning algorithms. These algorithms can be applied in cases where a response variable is known, as described in Chapter 6 of Machine Learning Using R by Ramasubramanian and Singh (2019). The goal of these kind of algorithms is to learn with the available data to be able to

[1]The figure does not show all the different algorithm categories, but only the ones used in this research.

give a prediction of the response variable. The response variable can be either continuous or categorical. If the response variable is categorical, the challenge is called a classification task. If the response variable is continuous, the challenge is called a regression task. In this research, we have problems regarding continuous variables. In the following two sections we will describe the two algorithms used; multiple linear regression and the *k*-nearest neighbors algorithm. The third paragraph discusses which performance measures could be used in regression tasks.

Multiple Linear regression

One way of estimating continuous variables from other variables is using linear regression. The variable to be predicted can be called the dependent variable or the response variable and the variables predicting this response variable can be called the independent variables, the predictor variables or the regressor variables (Montgomery, Peck, and Vining, 2015). The goal of regression analysis is to find relationships between the variables. An example is shown in Figure 3.2 and Figure 3.3. The left figure shows the result of throwing a ball from a standing position plotted against the result of throwing a ball when jumping. There is a clear relationship between the two variables. The right figure shows the same data, but now a regression line is drawn through the points. This line (or model) has a regression formula of the form:

$$y = \beta_0 + \beta_1 x + \varepsilon$$

In this equation, y represents the response variable, β_0 is the intercept (the value where the line crosses the y-axis) and ε is the difference between the real data point and the regression line. Furthermore, β_1 is the coefficient of the predictor variable x, and represents the change in y for a change of one unit in x. This is an example of simple linear regression. In multiple linear regression, we try to predict the value of the response variable with several predictor variables (Norouzian and Asadpour, 2012). Multiple linear regression models have equations with the following form:

$$y = \beta_0 + \beta_1 x_1 + ... + \beta_q x_q + \varepsilon$$

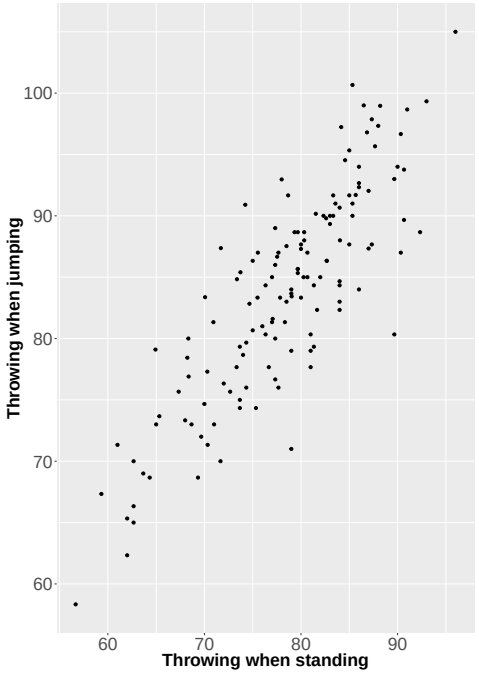

FIGURE 3.2: Scatter diagram.

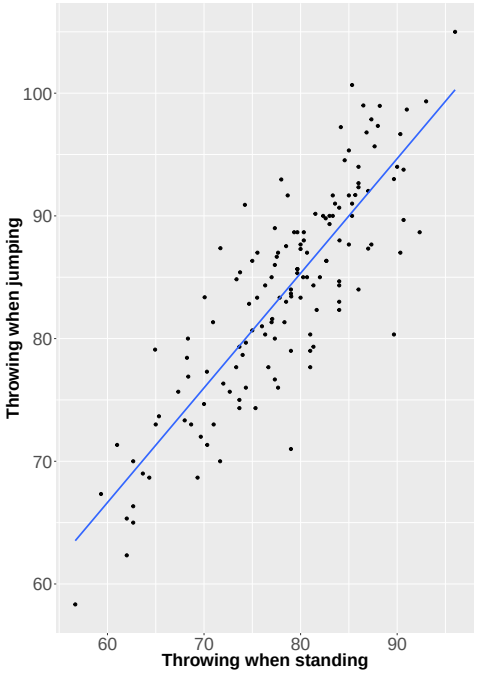

FIGURE 3.3: Scatter diagram with a regression line.

In multiple linear regression, a situation called multicollinearity can occur. This means that there is a linear relationship between two or more of the predictor variables (Adeboye, Fagoyinbo, and Olatayo, 2014). This is not really a problem when the goal of multiple linear regression is just to predict the dependent variable, according to Adeboye et al. (2014) and Paul (2006). However, the phenomenon can be a problem if we want to understand how the separate predictor variables influence the response variable. Paul (2006) also describes how multicollinearity can be detected. One of the methods is calculating the variance inflation factor (VIF). The VIF measures how much the variance of a regression coefficient increases due to multicollinearity, as explained by Alibuhtto and Peiris (2015). Both Paul (2006) and Alibuhtto et al. (2015) appoint that a VIF of above 5 or 10 for one or more predictors indicates multicollinearity.

Cook's Distance Data sets can contain observations that are extreme. These observations might influence regression models a lot (Montgomery et al., 2015). Stevens (1984) discusses several measures that can detect these influential data points. One of them is Cook's Distance. This measure looks at the change in regression coefficients if a certain data point is left out. This is done for every data point in the data set. To choose which of the data points are too influential, a cutoff level of $\frac{4}{n}$ can be used for example (Meer, Te Grotenhuis, and Pelzer, 2010), where n represents the number of data points. Data points that have a Cook's Distance that exceeds this cutoff level can be seen as influential and the decision can be made to remove this observation from the model.

Other linear regression assumptions We already mentioned that it is necessary to check for multicollinearity when performing multiple linear regression. Apart from this, there are other assumptions that should be reviewed when creating a linear

regression model. These are described by Elliot and Tranmer (2008) in the following way:

1. The residuals have a constant variance along values of the dependent variable. This is called homoscedasticity.
2. There are no extreme values in the data.
3. The residuals are normally distributed.
4. The residuals are not related to the independent variables.
5. The residuals are not correlated with each other.

Table 3.1 shows how the assumptions can be checked. It is clear from the table that the assumptions can be checked by looking at several plots or tests. This is also described by Elliot et al. (2008). The Breusch-Pagan test is a test to check whether there is constant error variance, and therefore homoscedasticity, proposed by Breusch and Pagan (1979). Outliers can be found by using the Bonferroni inequality test as described by Cook and Weisberg (1982). The Durbin-Watson is a test to check whether the residuals of a regression model are independent (Durbin and Watson, 1950).

TABLE 3.1: Ways to check the five linear regression assumptions.

Assumption	**Plot**	**Ways to check**
1	Residuals vs. predicted values	The residuals should be equally spread around $y = 0$, Breusch-Pagan test
2	Residuals vs. predicted values	Observations with a standardized residual or a predicted value of more than 3 or less than -3 can be seen as outliers, Bonferroni p-values
3	Quantile-Quantile plot/histogram	The residuals should be close to the diagonal line in the Q-Q plot
4	Residuals vs. predicted values	There should be no patterns visible
5	-	Durbin-Watson test

k-Nearest Neighbors

The second algorithm we used is the *k*-Nearest Neighbors algorithm (kNN). This algorithm can predict both categorical variables (classification) and continuous variables (regression) (Imandoust and Bolandraftar, 2013). In this research, kNN regression has been applied only. The idea of the kNN algorithm is to find the k most similar data points to a new data point, and use the values of the similar data points to estimate the value of the new data point (Zhang, Li, Zong, Zhu, and Cheng, 2017). In kNN regression, a way of predicting the value of new data points is by calculating the average value of the k nearest instances (Imandoust et al., 2013; Goyal, Chandra, and Singh, 2014). As Goyal et al. (2014) mention, the kNN algorithm is a non-parametric algorithm. This means that it does not make any assumptions about

the distribution in the data.

Figure 3.4 illustrates the idea of the kNN algorithm. If instance 1 is the new data point, and we want to find the two closest instances ($k = 2$) based on variables x_1 and x_2, then we will find that instances 2 and 3 are the closest to 1. We can now estimate variable x_3 for instance 1 by taking the average of variable x_3 of instances 2 and 3.

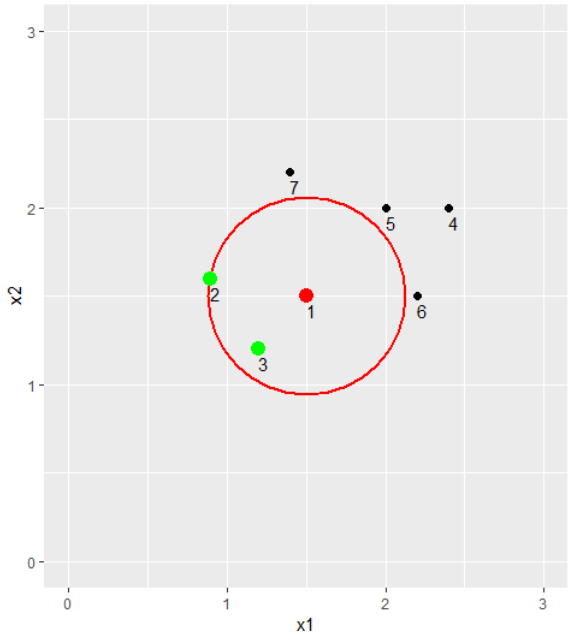

FIGURE 3.4: Finding the k nearest data points to a given data point.

What is clear from this example as well as from different other studies, is that there are three aspects that could change the outcome of an example like this (Goyal et al., 2014; Zhang et al., 2017; Punch, Goodman, Pei, Chia-Shun, Hovland, and Enbody, 1993). Firstly, this is the value of k; the number of other data points we look at when predicting the value of a new instance. Secondly, the features to look at when finding the nearest neighbors. Punch et al. (1993) mentions that if there are many features, selecting important features only can optimize the performance of the algorithm. Lastly, the distance metric used to find the nearest neighbors can determine the outcome.

Mulak and Talhar (2013) analyzed three different distance metrics for kNN classification which can also be used in kNN regression. The metrics they describe are the Euclidean distance, the Chebychev distance and the Manhattan distance. They calculate the distances between points as follows:

- **Euclidean distance**: The root square differences between two coordinates: $\sqrt{\sum_{k=1}^{m}(x_{ik} - x_{jk})^2}$, k is the kth variable, m is the number of variables, x_{ik} is the value of variable k for data point i, x_{jk} is the value of variable k for data point j.
- **Chebychev distance**: The absolute magnitude of the differences between two coordinates:
  $\max_k |x_{ik} - x_{jk}|$, k is the kth variable, x_{ik} is the value of variable k for data point i, x_{jk} is the value of variable k for data point j.

- **Manhattan distance**: The differences between two coordinates: $\sum_{k=1}^{m} |x_{ik} - x_{jk}|$, k is the kth variable, m is the number of variables, x_{ik} is the value of variable k for data point i, x_{jk} is the value of variable k for data point j.

Most studies and theoretical articles state that the Euclidean distance is the most commonly used distance metric. This is for example stated by Imandoust et al. (2013) and Goyal et al. (2014).

If variables have different ranges, it is necessary to standardize these variables, otherwise the variables with big scales will have a bigger impact than the variables with a smaller scale when selecting the neighbors (Shalabi, Shaaban, and Kasasbeh, 2006). There are several ways to standardize variables to comparable scales. One way Shalabi et al. (2006) mention is z-score normalization. The standardized value of a certain variable can be calculated as follows:

$$x_{inorm} = \frac{(x_i - \bar{x})}{\sigma_x}$$

In this formula, x_i is the ith value of variable x, $\bar{x}$ is the mean of variable x and σ_x is the standard deviation of variable x. As mentioned by Abdi (2010), the calculated z-scores of variables have a mean of 0 and a standard deviation of 1.

Fixed-radius near neighbors search Instead of using kNN to predict values of new data points based on the k neighbors of a certain data point, the distance between points principle can also be used to find the closest points of a certain data point, to compare data points. This can be called nearest neighbor search and the nearest neighbor search problem can be described as finding these nearest neighbors efficiently (Muja and Lowe, 2009). Instead of finding the fixed number of k neighbors, it is also possible to look at all neighbors within a certain distance from a data point. This can be called fixed-radius near neighbors search, as described by Dickerson and Drysdale (1990).

Performance measures

To compare models created with the supervised algorithms mentioned in this section to estimate continuous variables, several performance measures can be considered. This section describes the measures that were found in other studies and used in this research.

Two well-known measures are the mean absolute error (MAE) and the root mean square error (RMSE) (Chai and Draxler, 2014; Galdi and Tagliaferri, 2019). Those measures can be calculated as follows:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |\hat{y}_i - y_i|$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2}$$

n is the number of predictions, $\hat{y}_i$ is the ith prediction and y_i is the observed value of the ith observation. The difference between both measures is that the MAE

weighs each error the same, while the RMSE gives more weight to bigger errors.

Both the RMSE and the MAE have the same scale as the data. However, sometimes it might be more convenient to use a scale-independent measure. An example is when we want to compare results on different variables or if we want to get an overall performance of models where different variables are included. Galdi et al. (2019) name two of those scale-independent metrics; the relative absolute error (RAE) and the relative squared error (RSE). Those measures can be calculated as follows:

$$\text{RAE} = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{\sum_{i=1}^{n} |\bar{y}_i - y_i|}$$

$$\text{RSE} = \frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{\sum_{i=1}^{n} (\bar{y}_i - y_i)^2}$$

In this formula, $\bar{y}$ is the mean of the observed values.

Another metric to measure the performance of linear models that predict continuous variables is R^2 (Alexander, Tropsha, and Winkler, 2015; Healy, 1984). R^2 is also called the coefficient of determination and is a measure of variation explained by the regressor (Montgomery et al., 2015). Both Alexander et al. (2015) and Healy (1984) mention that there are several definitions of R^2. The former gives a definition that is appropriate for different predictive models. The formula they give is the following:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y}_i)^2}$$

Healy (1984) mentions that R^2 can be misleading when more predictors are added to the model. This should therefore be taken into account when the R^2 measure is used to compare models with different predictors. Moreover, Alexander et al. (2015) give a note on that the goal of prediction models is mainly to get a good accuracy and not a rate of how well it explains the variation in a chosen data set. Therefore, they recommend using a measure like the RMSE to check the model's usefulness as well.

3.1.2 Unsupervised learning

The second category of algorithms discussed is unsupervised learning algorithms. These algorithms can be applied in cases where a response variable is not known, as described in Chapter 6 of Machine Learning Using R by Ramasubramanian et al. (2019). The algorithms try to find similarities between data points. One category of unsupervised learning is cluster analysis. In this research we used the *k*-means clustering algorithm. This algorithm will be discussed in the following section.

k-means

The *k*-means clustering algorithm is a well-known algorithm used to find *k* clusters that are preferably not overlapping (Wu, 2012). In the book, Wu (2012) describes the main steps of the *k*-means algorithm as follows:

1. *k* initial centroids have to be chosen;

2. The distance from each data point to each cluster centroid is determined;
3. Each data point is assigned to the closest cluster centroid;
4. The centroid of each cluster will be recalculated to the mean of the data points belonging to the cluster centroid;
5. The above steps are repeated until the cluster centroids do not change anymore (convergence).

According to Jain (2010) there are three different parameters that can be tuned in the algorithm. First of all, this is the value of k which represents the number of clusters. An easy method of choosing a value of k is to try the algorithm a number of times and choosing the k that leads to the smallest error. The second parameter is the distance function. Just like with kNN, different distance measures can be used. A commonly used distance measure is the Euclidean distance, as described in the kNN paragraph of Section 3.1.1. Since this algorithm is also based on the distance between data points, the data should be standardized, just like for the kNN algorithm. The third parameter Jain (2010) mentions, is the cluster initialization. In the first step, random clusters are chosen. Each time the algorithm runs, different cluster centroids can be chosen, which means that the algorithm will return different results each time. Just like for the value of k, the algorithm can be executed several times with different initializations to find a good initialization.

Performance measures

To compare different clustering models like k-means, several indices can be considered. One of the indices mentioned by Maulik and Bandyopadhyay (2002) and Rendón, Abundez, Arizmendi, and Quiroz (2011) is the Davies-Bouldin (DB) index. With this index, it is possible to compare models with different values for k. Rendón et al. (2011) give the following formula:

$$\mathrm{DB} = \frac{1}{c} \sum_{i=1}^{c} \max_{i \neq j} \frac{d(X_i) + d(X_j)}{d(c_i, c_j)}$$

In this formula, c is the total number of clusters, $d(X_i)$ are the distances from samples in cluster i to the centroid of cluster i and $d(c_i, c_j)$ is the distance between the centroid of cluster i and the centroid of cluster j. As mentioned by both Abdi (2010) and Rendón et al. (2011), a smaller DB value is preferred over a bigger DB value.

3.2 k-fold cross validation

During the research, different models with different parameters are tested and compared several times. k-fold cross validation is a way of assessing the performance of predictive models (Rohani, Taki, and Abdollahpour, 2018). It can be used to select parameters. The procedure in k-fold cross validation is described in many researches. Rohani et al. (2018) and Anguita, Ghio, Ridella, and Sterpi (2009) explain the procedure as follows:

1. The data is split in k subsets;
2. $k - 1$ subsets are used to train a specific model;

3. The subset left is used to test the model;

4. Step 2 and 3 are repeated until each subset has been used as a test set once.

A rule-of-thumb is to use values for k of 5 or 10 (Anguita et al., 2009).

Varma and Simon (2006) describes that cross-validation can be used to select models. However, the test error can be biased. A way to overcome this is by leaving a few data points separately as a test set. The remaining data will be the input for cross-validation.

3.3 Sport data

The sports industry is an industry where a lot of data can be collected, and where this data can be used to improve teams or individual athletes just to make the difference compared to their competitors. Within teams, this can be done by selecting the right players. This is where talent identification can be very useful. There are quite some studies from recent years where data is used to determine whether someone is a talent in a certain sport and what kind of measures determine that someone is a talent.

An example is a study by Musa, Taha, Majeed, and Abdullah (2019). In their research, they first cluster their available data using k-means to create two types of performances in archers. After this, they try classification techniques like kNN and logistic regression to find out which technique classifies the archers the best according to their cluster. The authors followed this procedure for four different groups of variables; bio-physiological variables like heart rate when in rest, psychological variables like confidence, anthropometry measurements like height and arm span, and fitness performance like the vertical jump. For each of the groups of variables, Musa et al. (2019) found that there were some variables that had different average values in the clusters that were determined in the beginning. Furthermore, they analyzed the classification results of all techniques in all four groups of measurements to find out which algorithm could predict the class most accurately to be able to identify talents.

Another research in which test results are used to identify the most potential sport for a school child is done by Papić, Rogulj, and Pleština (2009). They created a web-based application with fuzzy logic. They mainly used the knowledge of experts to find out which values for certain tests are important for a specific sport.

Lastly, a research by Mohamed, Vaeyens, Matthys, Multael, Lefevre, Lenoir, and Philppaerts (2009) investigates which physical characteristics and which performance measures could identify handball talents to create a talent identification model. In their study, Mohamed et al. (2009) found that height, running speed and agility are important features to check to identify handball talents.

The studies mentioned in this section are all dealing with discovering values of measurements that could identify whether someone is a talent or not. However, the studies that we found were not dealing with the comparison of actual single talents.

Chapter 4

Data description and preparation

The goal of this chapter is to answer the first sub-question: What data pre-processing needs to be done before the data analytics process can start? Before answering this question, this chapter will give a description of the data in Section 4.1. Section 4.2 discusses which data we selected and how we handled the missing values.

4.1 Data description

The data for this research is data collected from "Het TalentVolgSysteem" (TVS), a talent tracking system that the NHV uses to track the progress and developments of handball talents. The data is an export from the system per 13-03-2019. The raw database consists of seven data tables. Appendix B shows the logical database structure. The collected data tables will now be described.

Talents This is a table with 3,440 rows and 18 columns. Each row represents a talent, meaning that there are 3,440 unique talents in the system at the moment of data collection. Every talent has a unique talent number, a date of registration and a date of deregistration if available. Furthermore, the data table contains a name, a date of birth (if available), the gender, and information about for which team(s) a talent plays. The last columns give information (if available) about their position in the field and whether the talent is right or left handed. The table also has an `id`, but since the talent number is also a unique identifier, this `id` column will be deleted. The resulting table has 3,440 rows and 17 columns.

Assessment This is a table with 637 rows and 46 columns. Each row represents an assessment of a talent by either the talent themselves or a scout. There are assessments of 345 unique talents in the period from 18-10-2017 until 02-03-2019. Each assessment has an `id` and the talent number of the talent being assessed is known. There is also a column with the identifier of the person (scout or talent themselves) assessing the talent. The majority of the columns represent grades on five different categories. The categories are: offensive, defensive, physical, performance and others. Each category is divided in grades of 1 until 3 on four or five more specific sub-categories, and a remark for that category. Moreover, there is a column for the average grade scored on a certain category and a column that represents the number of sub-categories that were filled in for an assessment.

Measurement types This is a table with 885 rows and 42 columns. Each row is a type of measurement for a certain group. However, the relevant information for a type of measurement is the same for every unique group. Therefore, to use this

table, it is cleaned up by removing the group column, and only keeping the distinct measurement types. The result is a table with 19 rows and 40 columns. Each row represents a measurement type. The measurement types are divided into six measurement categories; condition, functional movement, physical, power, speed, TRIMP (Training Impulse) and endurance/recovery. Each category has 1 until 5 measurement types with their own code. The majority of the columns (with names T01 until T34) contain information on how to read the `measurement_results` table. There is a physical measurement where T01 until T03 represent length, weight and fat percentage respectively. This means that the values W01 until W03 in the `measurement_results` table represent length, weight and fat percentage as well.

Measurement results This is a table with 8,641 rows and 38 columns. Each row represents the results of one kind of measurement or test for one talent. There are columns W01 until W34 that contain values. To see what a number in one of these columns represents, the `measurement_types` table should be used as described in the previous paragraph. There are 740 unique talents that have results for at least one type of measurement. The tests saved in the data table were taken in a period from 01-01-2007 until 21-02-2019. The remaining column is the unique talent number of the talent who performed the test.

Training types This is a table with 1,114 rows and 6 columns. Each row is a type of activity for a certain group. However, the relevant information for a type of activity is the same for every unique group. Therefore, to use this table, it is cleaned up by removing the group column, and only keeping the distinct training types. The result is a table with 11 rows and 4 columns. Each row represents a type of activity. The activity types are divided into three categories; training, match and others. Each category has 2 until 7 sub-categories. There is a column with a code that can be used when looking at the `training_log` table, to see what kind of training was logged.

Training logs This is a table with 140,841 rows and 16 columns. Each row has an `id` and represents a log of an activity for a certain talent. The talents are identified by their unique talent number. One of the columns represents the morning pulse. However, this is only measured for about a tenth of the logs. To see from what kind of activity the log is, the code can be used to find more information about the activity in the `training_types` table. The columns with the RPE grade, the duration and the feeling about an activity are the most important. RPE stands for Rated Perceived Exertion and it is a scale from 1 through 10 to measure the intensity of an exercise. The column with information about the feeling also has a scale from 1 through 10. The activities were logged in a period from 29-09-2005 until 12-03-2019.

Profile of mood states This is a table with 17,718 rows and 15 columns. Each row in this table has an `id` and gives information about how a talent is feeling and what their mental condition is, according to 11 categories. The categories are: sleep, stress, muscular pain, tiredness, nutrition, training performance, training hard, fun, motivation, self confidence and concentration. The values are on a scale from 1 through 5. A 0 is shown if a category was not filled in. A lower number means that a talent had a better feeling about a certain aspect. There are 728 unique talents that gave information about their mental condition. The talents are identified by their unique talent number. The mental condition was tracked during a period from 04-09-2011

until 12-03-2019.

With information from these tables, a final data table for analysis was created. Section 4.2 discusses how we got to this final data table.

4.2 Data preparation

To create the final data table, two main steps were taken. Firstly, the physical tests to be included in the final data set had to be chosen. Secondly, the missing values in the data had to be handled as much as possible. The following sections explain these steps in more depth.

4.2.1 Data selection

The data can be split in subjective data and objective data. The subjective data is the data according to perceptions of the talents or coaches and the objective data are results from physical tests. The goal table for analysis in this research is a table with a talent on each row and information about these talents in the columns. How the information for the columns was created will now be discussed for the subjective and objective data.

Subjective data There are three types of subjective data that we looked into during this research. They include data from the assessments, data from the training logs and data from the mood of talents. How these data types were combined into one cell of information for the final data table is described in Table 4.1.

TABLE 4.1: Subjective data: types and transformations.

Category	**Transformation**	**Number of features**
Assessments	The average score for each of the main assessment categories were averaged if there were several assessments, resulting in five features. There is also an average of these averages, and the average age for all the assessments was recorded.	7
Training logs	The average daily load of activities of the talent was calculated by multiplying the RPE grade with the duration and dividing the sum of the daily loads by the number of activities. The grades for the feeling feature the talent filled in for each activity were added up and averaged as well. The average age during all the logs was also used as a feature for a talent	3
Profile of mood states	The average score for each of the eleven categories were averaged for all the mood states of a certain talent, resulting in eleven features. There is also an average of these averages, and the average age for all the mood states was recorded.	13

Objective data We inspected the physical tests data and found that some tests were performed more regularly than others. A description of the most common tests can be found in Table 4.2. If a talent performed a certain test more than once, the best result of the last two years[1] was used as information for the final data table. The last column in the table gives the number of features that the test resulted in. Next to the features in the description, this number also includes a feature with the age of the talent at the time the test was performed.

TABLE 4.2: Objective data: the most commonly performed physical tests.

Category	Test	Description	Number of features
Physical	Length and weight	Length in centimeters and weight in kilograms.	3
Strength	Throwing speed	Throwing speed in kilometers per hour when jumping, when standing and when running.	4
	Jumping	Vertical jump in centimeters with the left leg, right leg or both legs, and long jump in centimeters with the left leg, right leg or both legs.	7
Speed	Sprinting	The time in seconds to sprint 5 meter, 10 meter, 15 meter and 20 meter starting with the left or right leg.	9
	T-test	The time in seconds to run a T-shape between cones starting with the left or right leg.	3
	Pro-agility test	The time in seconds to run 5 meter, turn and run 10 meter, turn and run 5 meter.	2
Condition	YO-YO test[2]	The score of the number of running two 20 meter shuttles followed by a break of 10 seconds.	2
	Interval shuttle run test	The number of 20 meter shuttles run in blocks of 4, 5, 6 or 8 shuttles with 15 seconds of rest halfway through the block and at the end of a block.	2

During the data selection process we tried to keep the the number of talents in the final data table as high as possible with the number of missing entries for these talents as low as possible. We divided the talents into four groups as follows:

1. **Professional players**: These talents play for the A-selection.
2. **Potentially professional players**: These are talents that play for Dutch selection teams, but not yet for the A-selection (B-selection, under 16/18/21, HandbalAcademie).
3. **The bench**: These are talents that have been playing for one of the selections before, but stopped for some reason.

[1] For this research, the last two years was a date between 23-05-2017 and 23-05-2019.
[2] YO-YO Intermittent Recovery test Level 1.

4. **The remaining groups**: These are talents from regional handball schools for example.

The group number of each talent was added to the data table as a feature. The created data table with all available talents and all collected information had 3,440 rows and a total of 69 columns of which 55 columns with information from the subjective and objective data and 14 columns with information about a talent like the talent number as the unique identifier, date of birth, date of registration and date of deregistration, gender, position and the group a talent plays for.

With this data table, the next step was to create a final data table with as few missing cells as possible. Since the talents in group 4 mainly have data for the subjective data types, they were removed as a first step. The resulting data table has 639 talents left. This data table has still a lot of missing values for some of the talents and some of the features. Only 89 of the talents left have assessment data. We decided to remove the assessment data from the final data table. Moreover, although the mood and perceptions of talents can be an important factor in the performance of players as also described in Section 3.3, we decided to limit this research to the objective data types only, since it might be hard to compare the performance of talents based on average grades of how someone is feeling. Therefore, to limit the missing values and in consultation with Edwin Kippers from the NHV, we decided that the following tests in these five main categories are the most important:

- Throwing speed:
  - In a standing position
  - When jumping
- Jumping:
  - Vertical jump with both legs
  - Long jump with both legs
- Sprinting:
  - The best result of the 20 meter sprint test starting with either the left or right leg
- Agility:
  - The best result of the T-test starting with either the left or right leg
- Condition:
  - The YO-YO test

Of the 639 talents left, there were 155 talents that performed the throwing speed tests, the jumping tests and the 20 meter sprint test[3]. Only fourteen of these talents are men. Therefore, to do fair analyses, we decided to only look at the female talents. This means that there are 141 talents left in the final data table. Each of these talents performed the throwing speed tests, the jumping tests and the 20 meter sprint test. 95 of these talents performed the T-test and 85 of these talents performed the YO-YO test. Although there are still quite some missing cells for these last two tests, we decide to include them in the final data table. The reason for this is that including the agility and condition tests gives a more complete view of a handball talent.

[3]The T-test and the YO-YO test will be discussed in Section 4.2.2.

4.2.2 Handling missing values

In the previous section we stated that the agility and condition tests are important to include to get a more complete view of the performance of a talent. Therefore, this section discusses the steps taken to limit the missing values for these categories.

Agility

There are results from two types of agility tests that talents in the final data set completed. These are the results from the T-test as described by Semenick (1990) and results from the Pro-Agility test as described by Harman and Garhammer (2008). Both tests require talents to run a certain distance in certain directions between cones. The results include the number of seconds from the start until the talent reaches the final cone. There are 26 talents of the final 141 talents who performed both agility tests.

In consultation with Edwin Kippers from the handball federation, it was agreed that it was most convenient to convert the results from the Pro-Agility test to the T-test, since the T-test is the test they are conducting nowadays. After inspecting the data, it seemed that there was no strong correlation between both tests. Therefore, we looked at how the remaining variables in the final data table could predict the result of the T-test.

As mentioned in Section 4.2.1, from the 141 talents in the final data table, there are 95 talents that performed the T-test at least once at a certain age. We decided to divide the ages of performing the different tests into three different categories. The categories correspond to the age categories of the different national handball selections and are the following:

1. The age of 17 or younger
2. The age between 17 and 19
3. The age above 19

Each test category (throwing a ball, jumping, sprinting and the T-test) could have been performed at different ages, so, each test category got its own age category variable. To fairly compare several models, we used the same data to train and test different models. 95 talents is a limited number of talents and certain observations in this data partition can have a big effect on the performance of final models. To find those data points, we followed the process illustrated in Figure 4.1. Firstly, we determined that the T-test is the dependent variable and that the other tests and the age categories are the independent variables. In step 3, we fitted a linear regression model with the chosen variables. With this model we created a Cook's Distance plot to find the influential observations. We removed the observations that were clearly influential and fitted the linear regression model again. After a few iterations we found that the influential observations found with the Cook's Distance plot were not very extreme anymore. We found fifteen influential data points, and the remaining 80 observations were used to train and test different models. At the end of this section, we will describe what we did with the influential observations.

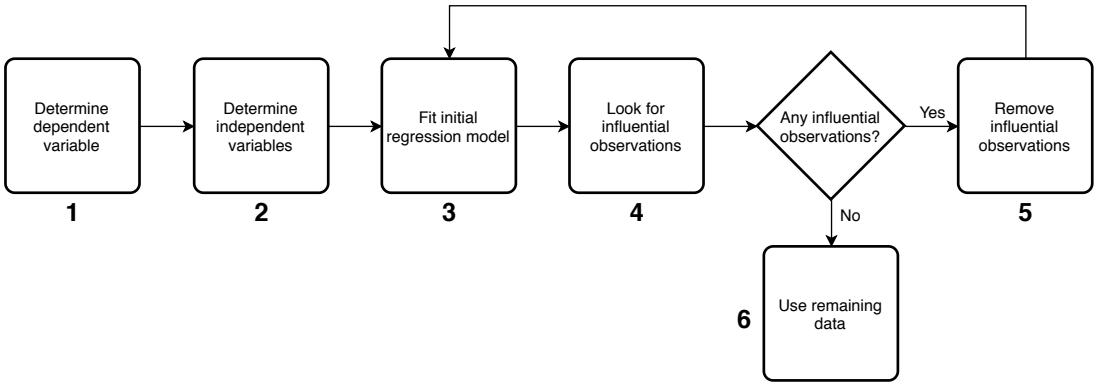

FIGURE 4.1: The process of finding and eliminating influential observations.

We will now describe how we used the remaining 80 observations to create a final model. To evaluate different models, we kept a test set of 15% of the data separate (12 observations). The training set of 68 observations was used to train models with different parameter settings in 10-fold cross validation. From all models trained, we recorded the following measures:

- RMSE of the full model on the training set:
  $\sqrt{\frac{\sum_{i=1}^{n}(P_i - O_i)^2}{n}}$, P_i = predicted value, O_i = observed value, n = number of observations.
- MAE of the full model on the training set:
  $\frac{1}{n}\sum_{i=1}^{n}|P_i - O_i|$, P_i predicted value, O_i = observed value, n = number of observations.
- R^2 of the full model on the training set:
  Collected from the linear model summary in *R*.
- Adjusted R^2 of the full model on the training set:
  Collected from the linear model summary in *R*.
- Average RMSE of the cross validation models:
  Collected from the cross validation result in *R*.
- Average MAE of the cross validation models:
  Collected from the cross validation result in *R*.
- Average R^2 of the cross validation models:
  Collected from the cross validation result in *R*.

Multiple linear regression The supervised learning algorithm we tested was multiple linear regression. All different tests (throwing when standing and when jumping, vertical jump, long jump and 20 meter sprint) and the age category of each test category can be used in all possible combinations in the regression formula. With eight variables, this means that there are 255 combinations (and thus, 255 formulas) to start with. With each formula, we fitted a linear model with all 80 observations and checked whether this model was significant ($\alpha = 0.05$), whether all separate

coefficients were significant and whether the R^2-value was higher than 0.5. Combinations of variables that met these conditions were used with the 10-fold cross validation for regression with the 68 observations from the training set. 10-fold cross validation was performed as described in Section 3.2. This resulted in different models from which we recorded the performance measures.

Clustering and multiple linear regression The second group of different models used to predict the T-test result was created by first clustering the data and subsequently performing multiple linear regression. The procedure followed was mainly the same as described in the previous paragraph. However, an extra variable was added to the regression formula; the cluster an observation belongs to. The differences in models in the linear regression of the previous paragraph were only in the regression formulas. Since with this method we first cluster the data, we got two additional parameters we needed to set. Firstly, the number of clusters to calculate and secondly, the variables to cluster on. We clustered the data with the k-means clustering algorithm as described in Section 3.1.2. To properly do this, we first standardized the data by calculating the z-score. We tried values for k ranging from 2 through 20 (19 values) and for every value of k we clustered the data on each of the combinations possible (31 combinations) on the five test results (throwing when standing and when jumping, vertical jump, long jump and sprint test). For each k and each combination of the five tests, we fitted a linear model with all 80 observations. We again tried all combinations of the eight variables in the regression formula[4]. This means that there were $19 \times 31 \times 255 = 150{,}195$ possible models. For each of the resulting models, we checked whether this model was significant ($\alpha = 0.05$), whether all separate coefficients were significant and whether the R^2-value was higher than 0.5. Combinations of variables that met these conditions were used with the 10-fold cross validation for regression with the 68 observations from the training set. 10-fold cross validation was performed as described in Section 3.2. This resulted in different models from which we recorded the performance measures.

Once we trained the possible models and collected the performance measures, we looked at the five best models based on the average RMSE value of the cross validation models. For these five different models, the RMSE, the MAE and the R^2-value on the separate test set of 12 observations were calculated. The RMSE and the MAE were calculated as described above and the R^2 of the test set was calculated with the R^2-formula in the performance measures paragraph in Section 3.1.1. The results are shown in Figures 4.2, 4.3 and 4.4. Table 4.3 shows the models corresponding to the model names in the figure.

[4]The cluster variable always had to be in the regression formula, since the result would otherwise be the same as for the multiple linear regression as described in the previous paragraph.

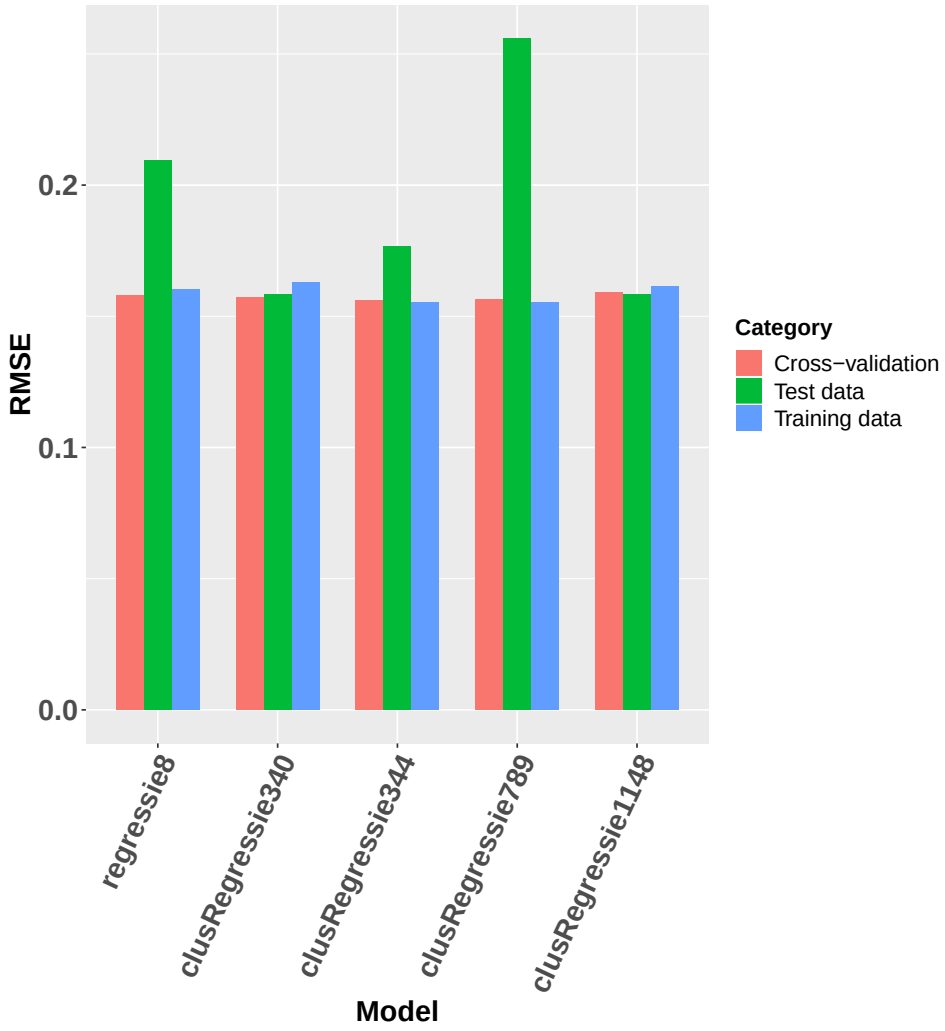

FIGURE 4.2: The RMSE of the regression models on the training set, cross validation and the test set.

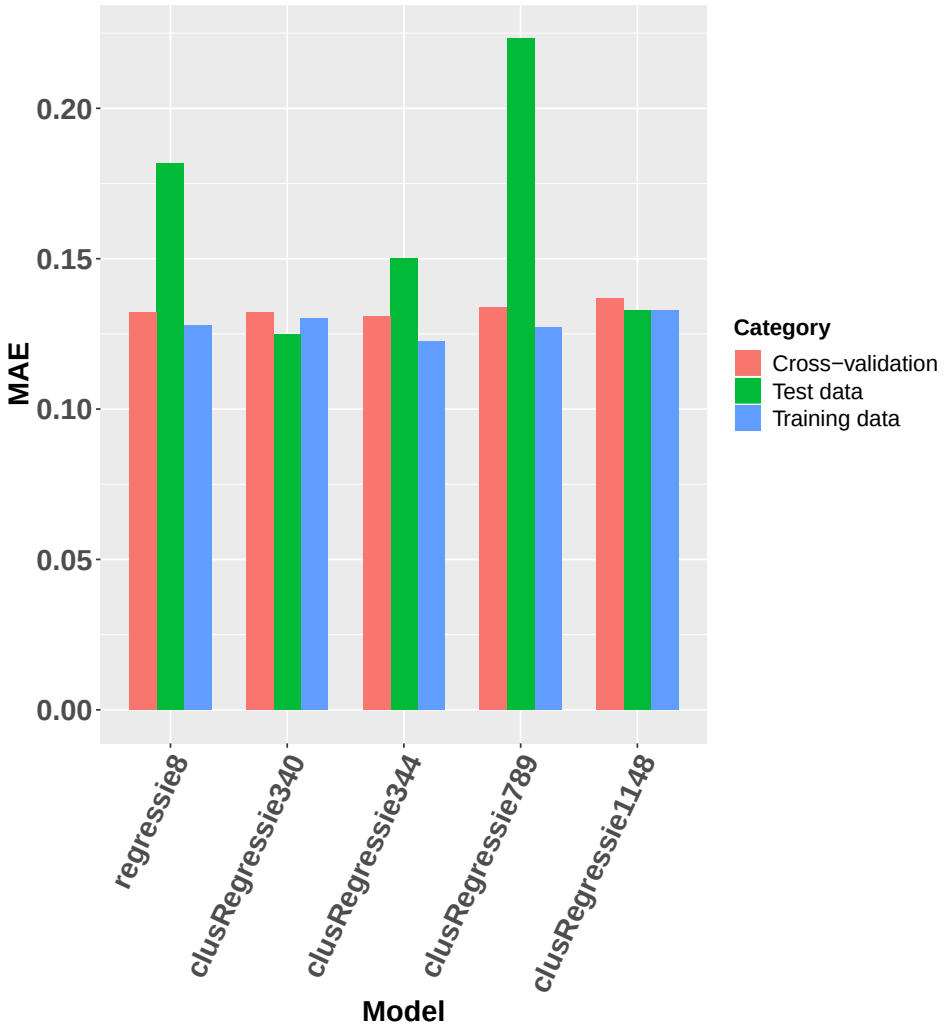

FIGURE 4.3: The MAE of the regression models on the training set, cross validation and the test set.

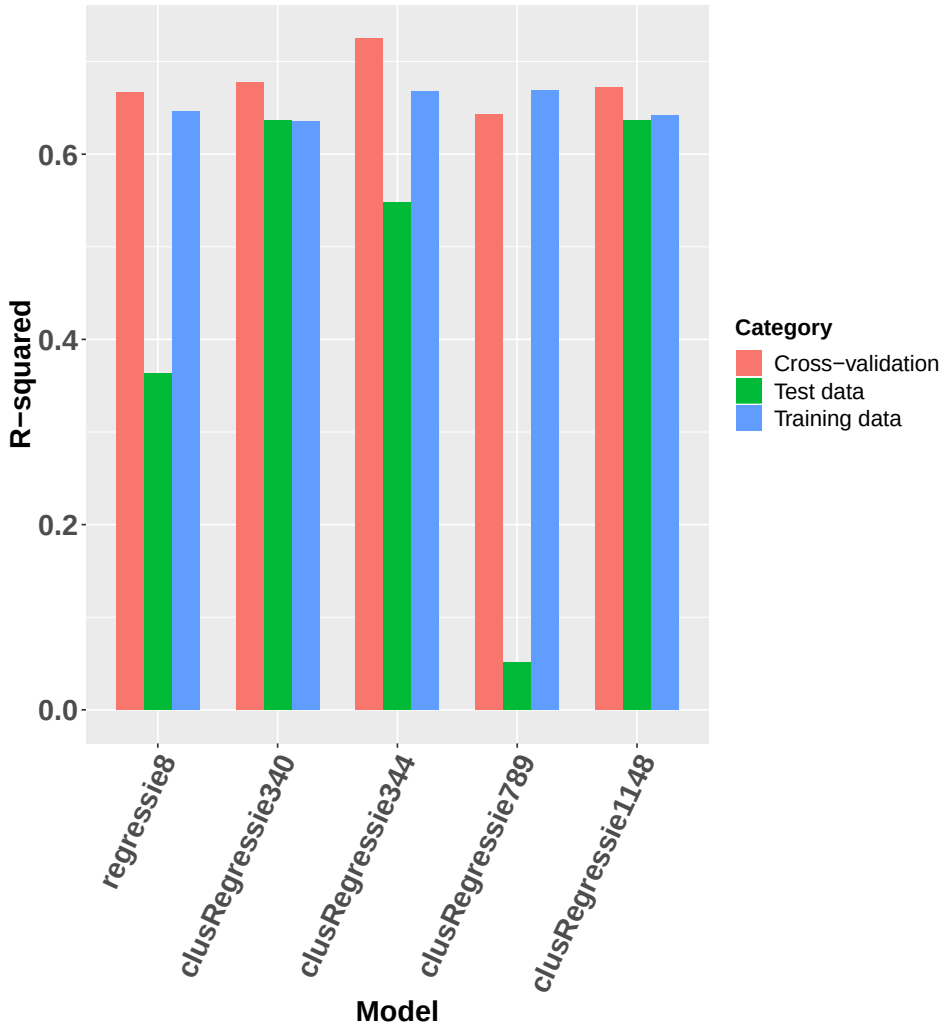

FIGURE 4.4: The R^2 of the regression models on the training set, cross validation and the test set.

TABLE 4.3: The models corresponding to the model names in Figures 4.2, 4.3 and 4.4.

Code	k	Clustering variables	Regression variables
regressie8	-	-	Throwing when standing, long jump, sprint
clusRegressie340	3	Throwing when standing, vertical jump	Vertical jump, sprint, sprinting age category, cluster
clusRegressie344	3	Vertical jump	Vertical jump, long jump, sprint, throwing age category, cluster
clusRegressie789	5	Throwing when standing, sprint	Long jump, sprint, cluster
clusRegressie1148	8	Vertical jump	Throwing when standing, sprint, throwing age category, cluster

From the figures it is clear that both *clusRegressie340* and *clusRegressie1148* perform most constantly on all three measures. The differences in performance of both models are not big. Therefore, we looked into the clustering performance of the models. We calculated the DB index of both clusterings on all 80 observations using the `clusterSim`-package in *R*. Table 4.4 shows the index values for both models. As mentioned in the performance measures paragraph in Section 3.1.2, a smaller DB index value is preferred over a bigger DB index value. Therefore, we decided to choose *clusRegressie340* to estimate the T-test result of talents who did not perform the T-test. Table 4.5 shows the performance of this model. With the performance on the complete data we mean the performance when the model is built on the data with the 80 observations.

TABLE 4.4: The DB index for the most constant performing models.

Model	**DB index**
clusRegressie340	1.61
clusRegressie1148	3.78

Measure	**Complete data**	**Training set**	**Cross-validation**	**Test set**
RMSE	0.156	0.163	0.157	0.158
MAE	0.121	0.130	0.132	0.125
R^2	0.671	0.636	0.678	0.636

TABLE 4.5: The performance of the final model to estimate the T-test result.

The final regression formula is the following:

$$\text{T-test} = 6.23828 - 0.06767 \times \text{vertical jump} + 0.13497 \times \text{sprint} - 0.09520 \times \text{sprintingAgeCategory} + 0.09543 \times \text{cluster}$$

In Appendix C.1 we present the regression summary. Moreover, in the multiple linear regression paragraph in Section 3.1.1 we mentioned that it is necessary to check some assumptions before we can deploy the model. The results of testing the assumptions for the model above can be found in Appendix C.1. The conclusion is that all assumptions are satisfied.

Influential observations The goal of the process we just described, is to estimate the T-test value of talents in the final data table, who did not perform a T-test. There were 46 talents who did not perform the specific test. As mentioned in the beginning of this section, we kept fifteen influential observations separately. For each talent, we wanted to check whether we could use the model created as described above, or whether we should use the influential observations to estimate the T-test result. We did this following the process in Figure 4.5. The first step is to find the optimal nearest neighbors distance. This can be done by following the method in Chapter 5. In the case of this example, the optimal nearest neighbors distance was 1.71 and it was found using all test results except the T-test result in a training set of 101 talents and a test set of 40 talents. In step two we looked for the nearest neighbors within the chosen distance for every new data point of which we wanted to estimate the T-test result. Since we looked at the neighbors within the distance only, it meant that for

each new data point, the number of neighbors taken into account could be different. For each neighbor of a certain new data point, we checked whether it was an influential data point, or a data point which was used to built the regression model as described above. The percentage influential data points in the set of neighbors was calculated. After this, we checked for each new data point whether this percentage was below or above a chosen threshold. The threshold we considered was 50%. This meant that if the percentage influential observations of all neighbors was 50% or less, the regression model was used and if the percentage influential observations was above 50% a model where the influential observations were included was used. The outcome was that the percentage of influential neighbors was 50% or more in 4 of the 46 cases and that for the remaining 42 talents, the percentage of influential neighbors was below 50%.

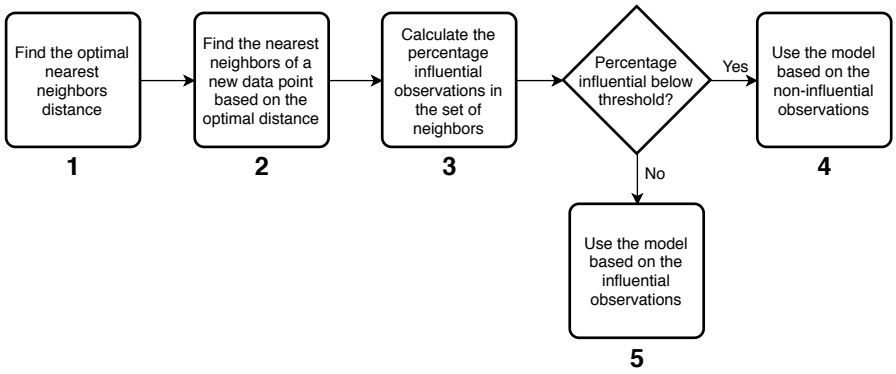

FIGURE 4.5: Choosing the model to use when trying to estimate a variable of new data.

For the 42 talents we used the regression model as described in the previous paragraph. For the four other cases we had to look for a different model. We followed a similar procedure as the regression model without the influential observations. First of all, the data was split into a training set and a test set again to evaluate different models. We got a training set of 80 observations and a test set of 15 observations. Again, we tried all possible combinations of features in regression formulas for the normal multiple linear regression, which means that there were 255 different models. We also performed the clustering before the multiple linear regression with the same combinations of parameter settings as before, which means that there were 150,195 different clustering with multiple linear regression models. The same requirements were used to decide whether a model could be used in the 10-fold cross validation step. The five best performing models based on the RMSE were tested with the test set of 15 observations. Figures 4.6, 4.7 and 4.8 show the performances regarding the RMSE, the MAE and the R^2 respectively.

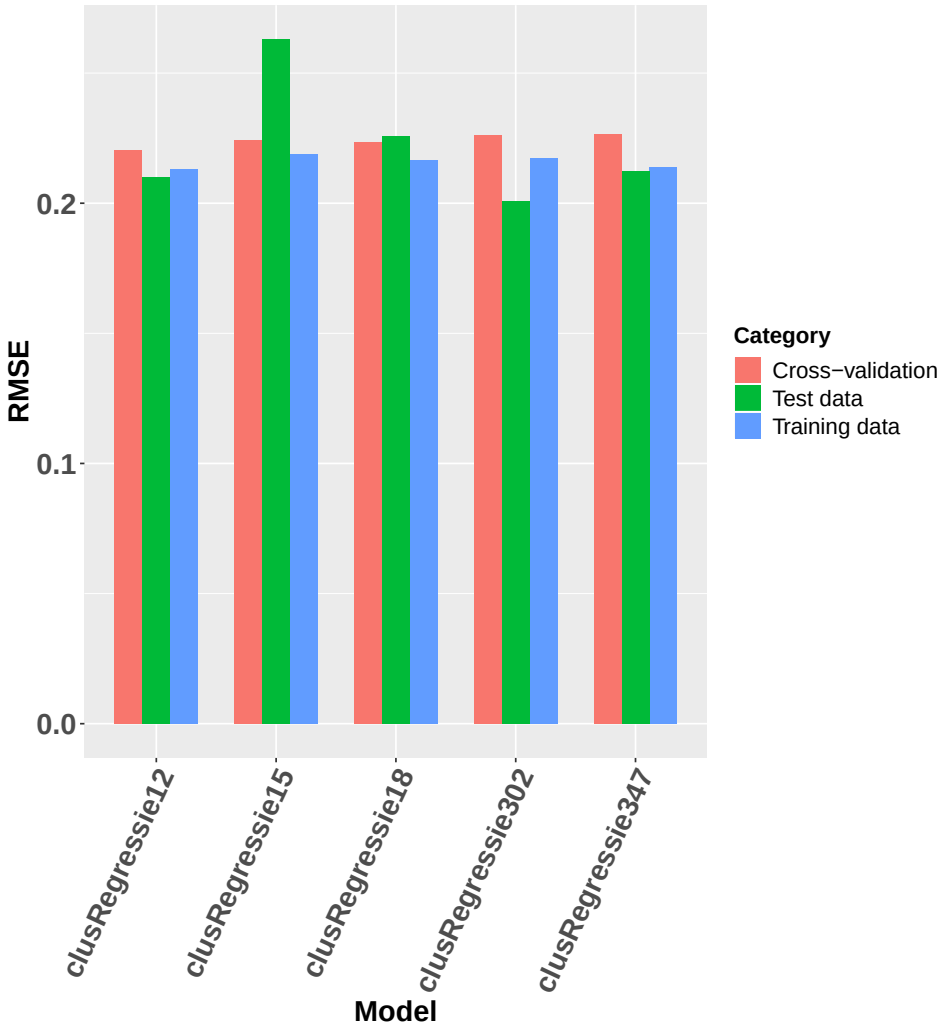

FIGURE 4.6: The RMSE of the regression models on the training set, cross validation and the test set, including the influential observations.

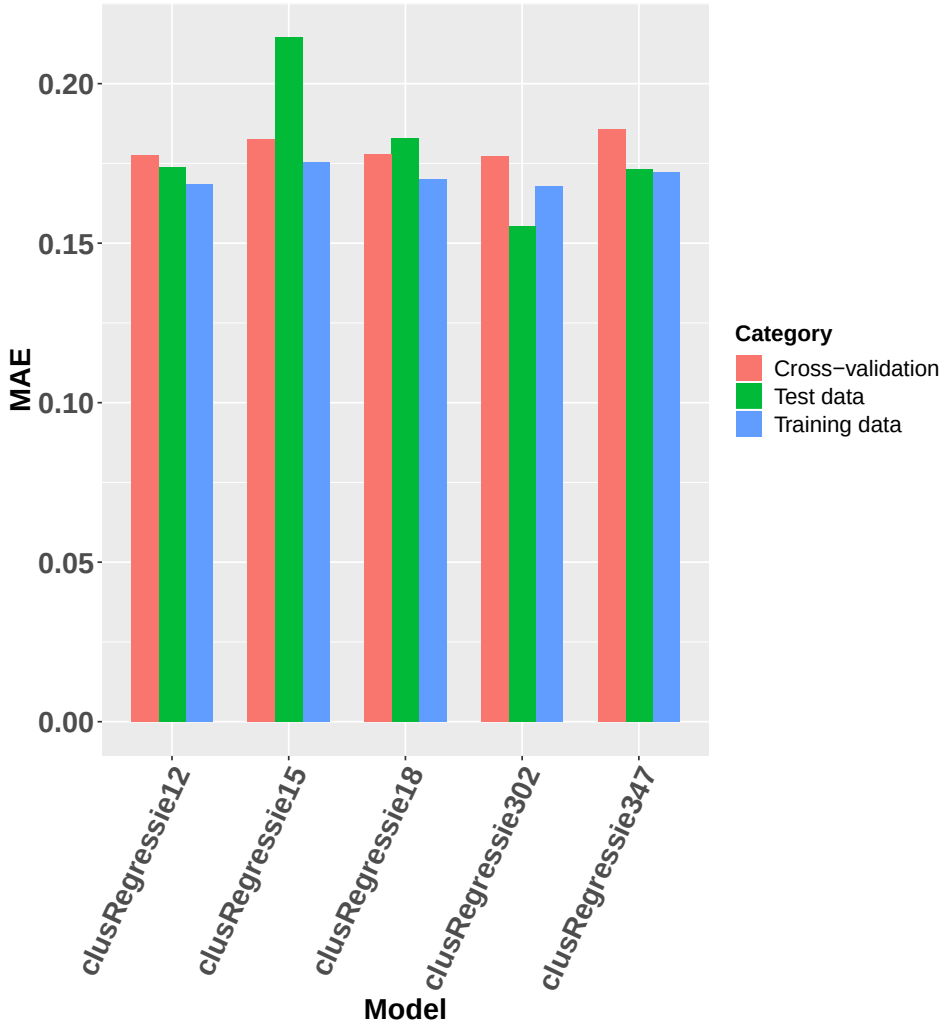

FIGURE 4.7: The MAE of the regression models on the training set, cross validation and the test set, including the influential observations.

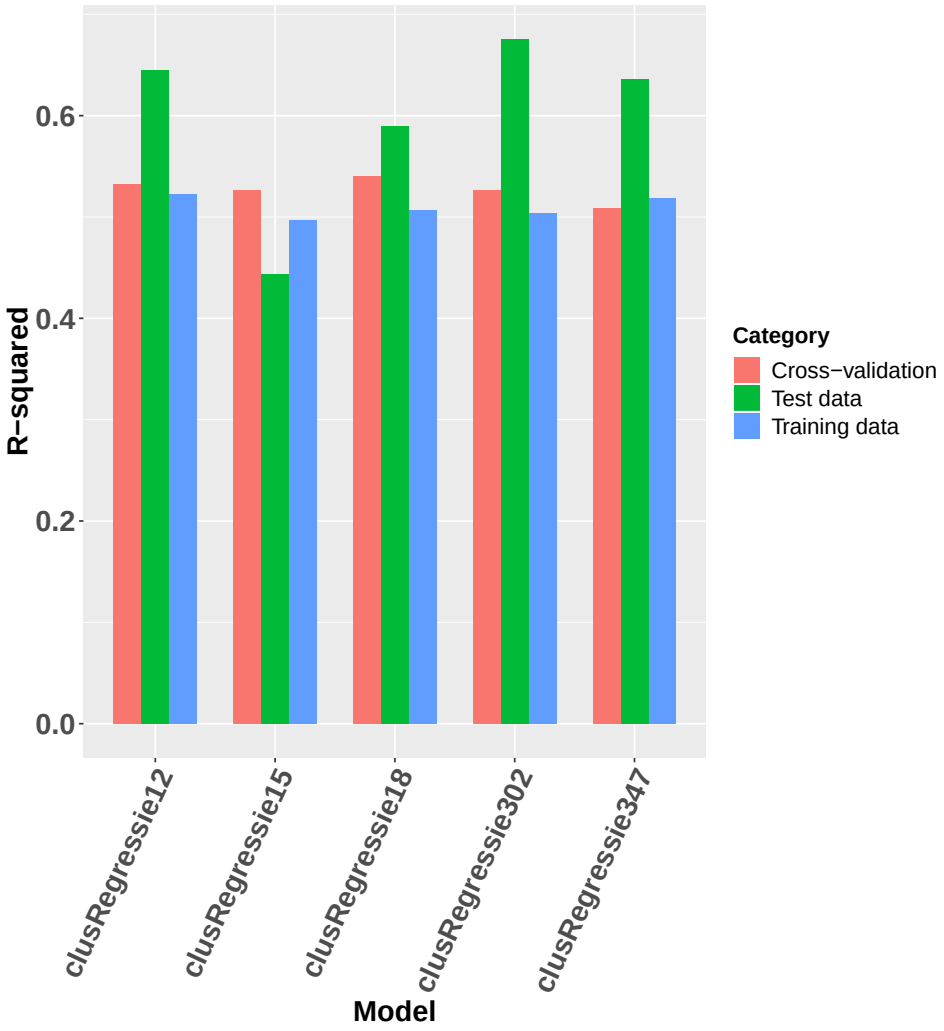

FIGURE 4.8: The R^2 of the regression models on the training set, cross validation and the test set, including the influential observations.

TABLE 4.6: The models corresponding to the model names in Figures 4.6, 4.7 and 4.8.

Code	k	Clustering variables	Regression variables
clusRegressie12	2	Throwing when jumping, long jump	Throwing when standing, throwing when jumping, sprint, cluster
clusRegressie15	2	Throwing when jumping, sprint	Throwing when standing, sprint, cluster
clusRegressie18	2	Throwing when standing, long jump	Throwing when standing, throwing when jumping, sprint, cluster
clusRegressie302	9	Throwing when standing, long jump, sprint	Throwing when standing, throwing when jumping, sprint, cluster
clusRegressie347	12	Throwing when jumping, throwing when standing, vertical jump, long jump, sprint	Throwing when standing, vertical jump, sprint, jumping age category, cluster

From the figures it is clear that mainly *clusRegressie12* and *clusRegressie18* perform most constant regarding the RMSE and the MAE measures. The only difference between both models is the variables that we clustered on; with *clusRegressie12* we clustered on the throwing when jumping and the long jump variables, and with *clusRegressie18* we clustered on the throwing when standing and long jump variables. Regarding the R^2, the measure is higher on the test set for both models. We also checked the DB index of both models again. The indices are shown in Table 4.7. Since *clusRegressie12* performs a little better on all measures on all data sets (except for the R^2 on the cross validation set which is a little better with *clusRegressie18*), and because the models' DB index is a bit smaller, we decided to use the *clusRegressie12* model to estimate the T-test value of the four cases where the percentage of influential neighbors was above the threshold of 50%. Table 4.8 shows the performance of this final model.

TABLE 4.7: The DB index for the most constant performing models, inclueding the influential observations.

Model	DB index
clusRegressie12	1.61
clusRegressie18	1.71

Measure	Complete data	Training set	Cross-validation	Test set
RMSE	0.212	0.213	0.220	0.210
MAE	0.169	0.169	0.177	0.174
R^2	0.552	0.522	0.532	0.645

TABLE 4.8: The performance of the final model to estimate the T-test result of observations whose percentage influential observations is higher than the threshold.

The final regression formula is the following:

$$\text{T-test} = 6.57899 - 0.13908 \times \text{throwingStanding} + 0.15075 \times \text{throwingJumping} + 0.17429 \times \text{sprint} - 0.19955 \times \text{cluster}$$

In Appendix C.2 we present the regression summary. Moreover, in the multiple linear regression paragraph in Section 3.1.1 we mentioned that it is necessary to check some assumptions before we can deploy the model. The results of testing the assumptions for the model above can be found in Appendix C.2. The conclusion is that all assumptions are satisfied.

Condition

There are results from two types of condition tests that talents in the final data set completed. These are the results from the YO-YO Intermittent Recovery test Level 1 as described by Bangsbo, Iaia, and Krustrup (2008) and results from the Interval Shuttle Run test as described by Lemmink, Verheijen, and Visscher (2004). Together with Edwin Kippers from the handball federation, we decided to try to estimate the YO-YO test result, since this is the test they are performing nowadays. We performed similar analyses as we did for the T-test. Unfortunately, due to the limited data, we did not get results that were satisfactory.

4.2.3 Conclusion

As soon as the test variables were selected and the missing data of the results from the T-test were handled, we got the final data table for analysis in this research. The final data table consists of fourteen columns and 141 rows. The first column is the unique talent number of each talent. The remaining columns are in three categories, namely: physical test results, age when the test was taken and the age category when the test was taken. From the T-test result we do not have an age variable. This is because we only know the age for a part of the talents. Table 4.9 gives descriptive statistics of the physical test variables. Appendix D shows the complete information of all thirteen variables.

TABLE 4.9: Descriptive statistics of the physical tests in the final data table.

Test	Minimum	Maximum	Mean	Median	Standard deviation	Skewness
Throw jumping (km/h)	58.33	105.00	83.76	84.67	8.6767791	-0.29969420
Throw standing (km/h)	56.67	96.00	78.33	79.00	7.8764698	-0.42862161
Vertical jump (cm)	20.97	60.00	35.94	34.90	8.0421036	0.56961147
Long jump (cm)	155.67	267.50	214.44	218.00	24.9381297	-0.18991300
T-test (s)	5.51	7.19	6.25	6.23	0.2976720	0.3103657
Sprint 20m (s)	2.92	3.68	3.22	3.21	0.1301325	0.59357458

Chapter 5

Comparing talents

The final data table that was created in the previous chapter can be used for analysis. The goal in this case is to be able to give sports scouts and coaches advice by comparing sports talents. Thereby, we will answer the second and third sub-questions: How can scouts be supported when looking for the best performing sports players and how can coaches be supported when training their sports talents? With comparing talents, we mean both comparing young players with current talents as well as comparing talents within the same group or team, as mentioned in Section 1.3. In this chapter we will first describe the method in Section 5.1 and subsequently we will describe the results of applying the method to the handball case in Section 5.2.

5.1 The method

This section describes the idea in general. The method is based on the kNN regression algorithm and the goal is to find the best nearest neighbors model for a given data set. The process is separated in four main steps. Section 5.1.1 will describe these steps. Instead of just finding a model to compare talents with, models that can predict certain variables well might be created. Section 5.1.2 describes how this can be done.

5.1.1 Finding the best model to compare talents

The following paragraphs discuss the four steps that can be followed to find the best performing model to compare talents with.

Step 1 - Preparing the data

The input data for this method is a data table with a person on each row and results or characteristics (as numeric variables) of a person in each column. Figure 5.1 shows the procedure in the first step. As mentioned in the paragraph about kNN in Section 3.1.1, it is necessary to standardize the data to create comparable scales for each variable (1). This can be done by calculating the z-score for each cell as follows:

$$x_{inorm} = \frac{(x_i - \bar{x})}{\sigma_x}$$

In this formula, x_i is the ith value of variable x, $\bar{x}$ is the mean of variable x and σ_x is the standard deviation of variable x.

As soon as the data is standardized, the data needs to be split in a training set and a test set to be able to test models that perform well on the training data, on data

it has not seen before (2). A percentage of the total data can randomly chosen as a test set. However, since this is a kNN procedure, it is recommended to keep an eye on which data points are chosen to be in the test set to prevent the test set to have data points that have big distances from the training set only. We take this measure in this case since with the limited data this chance exists. As mentioned in the kNN paragraph in Section 3.1.1 the calculated z-scores have a mean of 0. Therefore, a way of supervising which data points could be chosen to be in the test set is by choosing from the data points that have standardized values between -1 and 1 for all or almost all variables. From the set of data points that meet these requirements a test set can be created by randomly selecting a number of data points.

As soon as the standardized data is split in a training and a test set, the people in the standardized training set can be selected from the original (unstandardized) data table (3). This is the unstandardized training set. This training set has to be standardized again by calculating the z-score for each variable for each person in the training set (4). Since the test set should be seen as data where the model is not built upon and we want to pretend it is new and unseen data, it is necessary to standardize the test set with the parameters of the training set (the mean and the standard deviation of each variable) (5). This can be done by taking the original, unstandardized data of the people in the test set from the original data table and then by calculating the z-scores with the parameters from the training set. The output of this first step is a standardized training set and a standardized test set.

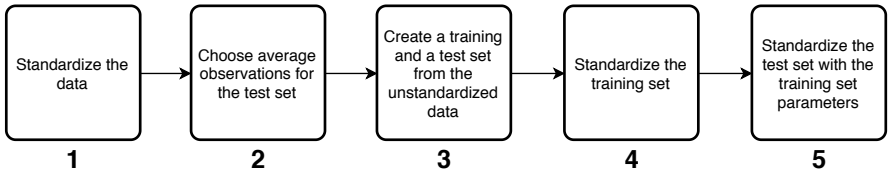

FIGURE 5.1: Comparing talents - step 1.

Step 2 - Choosing different model parameters

Figure 5.2 shows the sub-steps in the second step. In the kNN paragraph in Section 3.1.1 we mentioned three different aspects (parameters) that can change the outcome of a model; the value for k or the distance range (1), the distance measure (2) and the set of variables to find neighbors with (3). Different combinations of these parameters create different models. The more different values for each parameter we want to test, the more different models will be created. As an example we can try the following parameters:

- The neighbors:
  - k: We can choose values of k ranging from 1 through 5, then we will get 5 possibilities.
  - Distance range: We can choose maximal distances ranging from 0.6 through 1 with steps of 0.1, then we will get 5 possibilities.
- Distance measure: We can choose the Euclidean distance to measure the distance between data points.
- Set of variables: If we have two variables and we want to try every possible combination of one or more of these variables, we get three possibilities.

In the situation described above, we will get a total of 5 (either values for k or maximal distance) * 1 (distance measure) * 3 (combinations of features) = 15 different models.

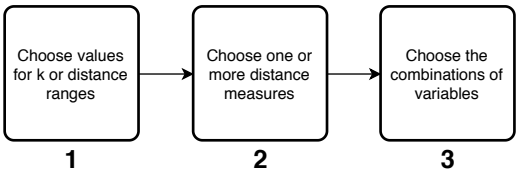

FIGURE 5.2: Comparing talents - step 2.

Step 3 - Calculating the results of different models

Figure 5.3 shows the sub-steps of the third step. In this step, for each of the parameter settings (1), we will create a data table with the observed results (standardized) from the training set and additional columns for estimations for each variable and additional columns for the differences between the observed values and the estimated values (2). Table 5.1 gives an example of the described data table. This data table should be copied as many times as the number of different models we would like to test. In the example from Section 5.1.1 this means that we will get 15 data tables. Since we try all possible combinations, it means that an exhaustive search is implemented. The next step is to fill in the blanks of the data tables. For every row (3) we will look for the nearest neighbors based on the chosen distance measure and based on the variables to look at in the specific model (4). In the example in the table below it is possible to look at three different combinations of variables; x_1 only, x_2 only, or both x_1 and x_2. If we are looking for the neighbors based on x_1 only, we will still estimate both x_1 and x_2 for each row. If values for k are specified, we will look for the k neighbors of a certain row. If values of distances (d) are specified, we will look at all the neighbors that are d away from the data point or closer. In this case each row can have different numbers of neighbors. We can now estimate the x_1 value for the first row by for example taking the average of the x_1 values of the neighbors (5). This should be done 15 times; once for each model. The result is 15 data tables with results of applying a model to the training set. For each of the data tables we can calculate the RMSE by averaging the RMSE values of all variables (6). There will be 15 RMSE values that correspond to the performance of the 15 models trained on the training set.

TABLE 5.1: An example of a data table used for every combination of parameters.

ID	x_1 observed	x_1 estimated	x_1 error	x_2 observed	x_2 estimated	x_2 error
1	0.60	-	-	1.10	-	-
2	0.50	-	-	1.15	-	-
3	0.75	-	-	0.95	-	-
4	0.60	-	-	1.30	-	-

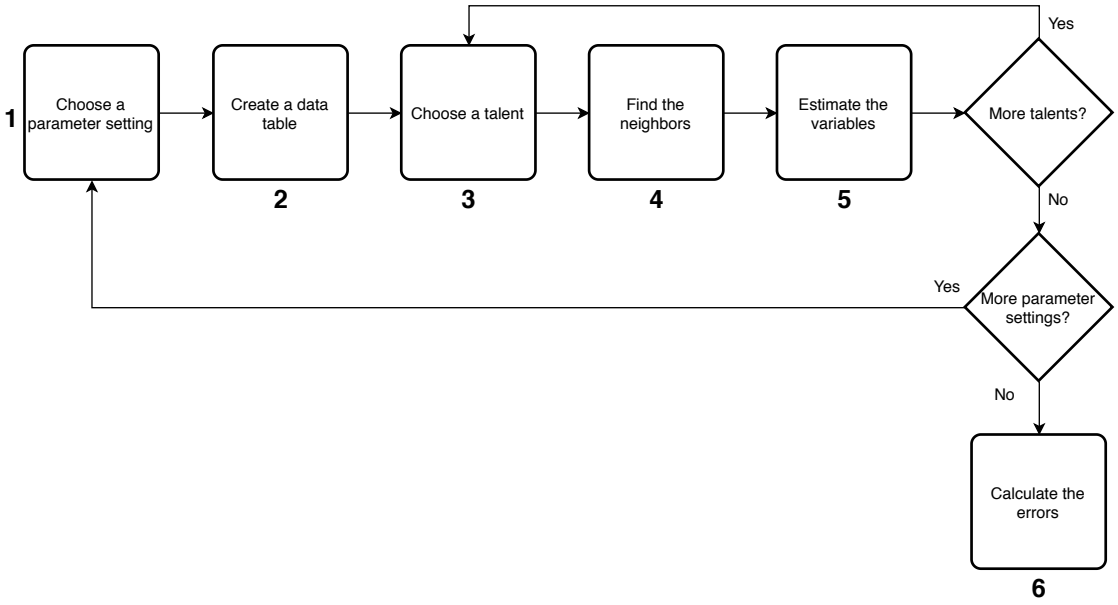

FIGURE 5.3: Comparing talents - step 3.

Step 4 - Finding the best model

Figure 5.4 shows the sub-steps of the last step. In the previous step, we found the performance of the specified models. The next step is to find the best and most constant performing model. This can be done by choosing the model which has the smallest RMSE (1), and testing these parameters with the test set. Testing with the test set can be done by creating a data table similar to the ones in the previous step (2) and then for each talent (3) we will find the closest data points in the test set from the data points in the training set using the parameters in the chosen model (4). The average of each variable of the neighbors can be used as the estimation again (5). The risk of choosing only the best model on the training set with small data is that it might perform quite different on the test set. A way to overcome this risk is by taking not just the best performing model on the training set, but the five or ten best performing models for example. These five or ten models should then be tested with the test set and the RMSE of these model should be calculated (6). The model performing similarly on the test data compared to the training data, can be chosen as the final model (7).

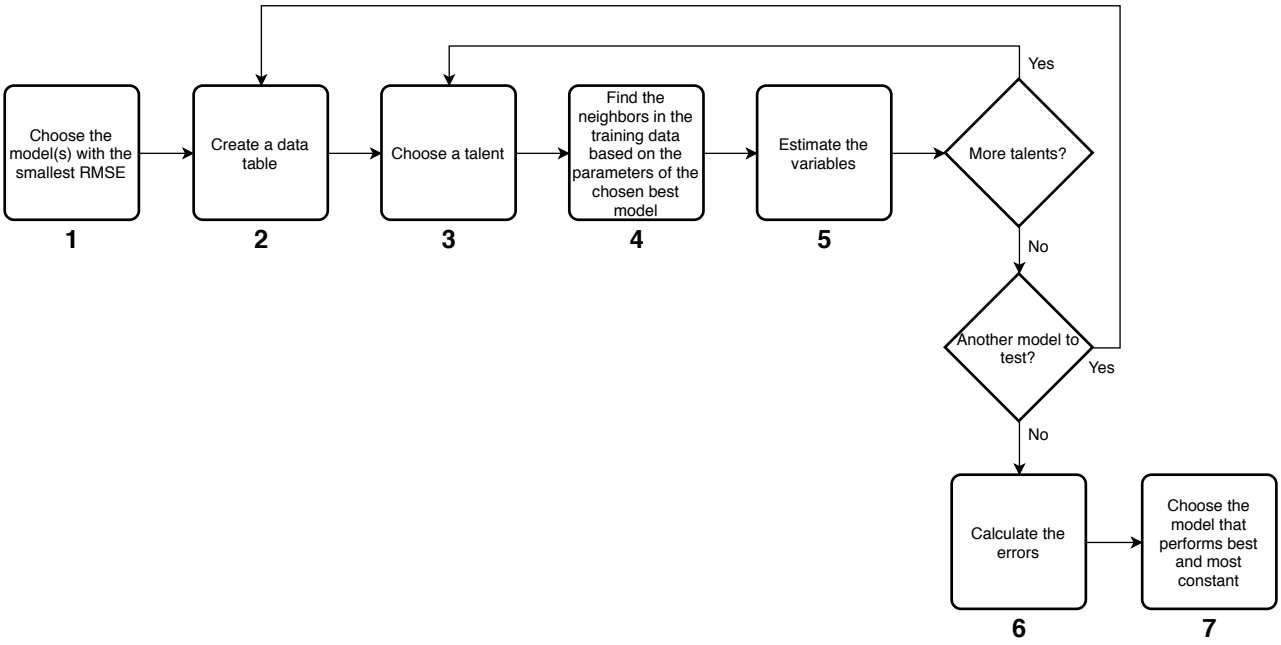

FIGURE 5.4: Comparing talents - step 4.

5.1.2 Find the best models to predict variables

In the previous section we described the steps to find the best model to compare talents. During step 3, many different models are trained. Of every single model the error will be calculated. This means that a big model base is created in which not every model is based on finding the neighbors based on all variables. Therefore, it might be possible that some models predict certain variables well. Whether this is the case for a certain variable, can be checked by following a similar process to step 4 in the previous section. Instead of choosing the models with the smallest overall error, we can look at errors of a certain variable in models where that variable is not taken into account when finding the neighbors. Again we can for example choose the five or ten best models and test these models on the test set. The best and most constant performing model can be chosen. It is possible that the best and most constant performing model is not performing sufficiently to actually predict the variable. Whether a model is sufficient or not can be discussed with a domain expert.

5.2 The results

To find the best and most constant performing model to compare handball talents, we applied the method described in Section 5.1.1. This section describes how we created a training set and a test set in Section 5.2.1, which models we tried in Section 5.2.2 and what the results were in Section 5.2.3. Furthermore, we discuss which models can possibly be used to predict certain variables in Section 5.2.4.

5.2.1 The data preparation

Before we could try different models, we had to split the data in a training and a test set. We followed the procedure in step 1 from the previous section. We decided to use around 25%-30% of the 141 talents as test data. We standardized the data set

with 141 talents and decided to choose talents who had standardized values between -1 and 1 for five out of six variables. There were 65 talents who met this condition and we randomly chose 40 of them to be in the test set. We used the unstandardized data of the 101 talents in the training set and the 40 talents in the test set, because as described in the previous section, we want to standardize these data sets separately. We standardized the 101 training samples and we used the mean and standard deviation of each variable to standardize the test data. The result is two data sets which are both standardized and ready to be used to train and test different models.

5.2.2 The chosen model parameters

We had to decide which models we wanted to try with the training data, as explained in step 2 in the previous section. Table 5.2 shows which model parameters we chose. What is clear from the table is that we decided to use different distance ranges to find the neighbors instead of a value of k to find the neighbors. In total, we tried 281 distance ranges. In this research we only used the Euclidean distance to find the distance between data points since this is the most commonly used metric as described in Section 3.1.1. However, it is possible to check if there are any differences in outcome when another distance measure is tried. For the variables parameter we chose all possible combinations of the six variables. This means that there were 63 possibilities. We tried each possible combination of parameters. Therefore, the total number of models we trained was: $281 \times 1 \times 63 = 17{,}703$. We believed that except for the distance metric, we tried all important parameter settings.

TABLE 5.2: The chosen model parameters and the total number of models trained.

Parameter	**Chosen values**	**Number of models**
Distance range	0.2 through 3.0 in steps of 0.01	281
Distance measure	Euclidean distance	1
Variables	Throwing when standing, throwing when jumping, vertical jump, long jump, sprint, T-test	63

5.2.3 The results of the best models

From all the 17,703 different models, we calculated the estimation for each variable for each talent using the procedure in step 3 in the previous section. From every model we calculated the average RMSE value of all six variables. As described in step 4 of the previous section, we can choose the ten best performing models on the training data to find out which model is performing most constant. In the case of the handball data, the best overall results were obtained with models that were quite similar. The results of the best models on the training data regarding the average standardized RMSE are shown in Figure 5.5. On the x-axis we show the model. The part before the hyphen corresponds to the distance range in the model and the part after the hyphen corresponds to the variables that were looked at. Variable combinations 63 and 62 are the following:

- 62: Throwing when standing, vertical jump, long jump, T-test, sprint
- 63: Throwing when jumping, throwing when standing, vertical jump, long jump, T-test, sprint

As shown in the bar plot in the figure, all top ten models performed quite similar on the training data. There were only some very small differences. The differences of performance of the models on the test data were not as big either. However, the most right model in the figure (*1.6-62*) performed worse on the test set compared to the models where all variables (combination 63) were taken into account to find the neighbors. To pick the final best model, we looked at those nine best models. Model *1.87-63* is the best performing model on the training set and the second best performing model on the test set. Therefore, we decided to choose this models as the final best model to compare talents with. This means that we will look at a Euclidean distance of 1.87 when searching for the neighbors, and we will look at all features (throwing when standing, throwing when jumping, vertical jump, long jump, T-test, sprint). This model has an RMSE on the complete data of 0.3855604.

Since the best models had distance ranges that were close to each other, we concluded that we did not have to look any further at smaller distance ranges than 0.2 or bigger distance ranges than 3.0.

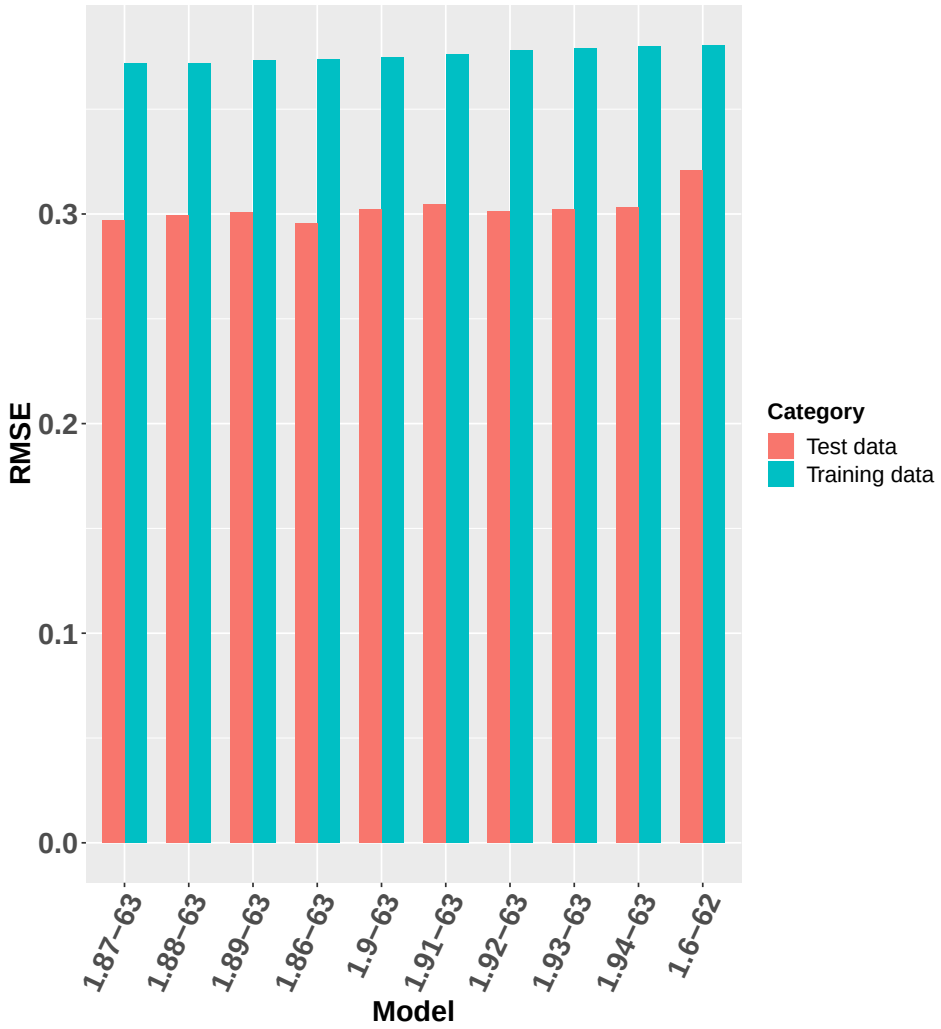

FIGURE 5.5: The ten best models on the training data regarding the RMSE.

Table 5.3 shows the RMSE values on the training set, on the test set and on the complete data for each separate variable of the chosen model.

TABLE 5.3: The RMSE on the training set, the test set and the complete data of the best model of the separate variables.

Feature	RMSE training	RMSE test	RMSE complete
Throwing when jumping	3.094571	1.845691	3.074837
Throwing when standing	3.238799	2.856922	3.309062
Vertical jump	2.957705	2.679951	2.821639
Long jump	9.706269	8.404966	8.979888
T-test	0.1433081	0.1118249	0.1317311
Sprint	0.05494259	0.04093093	0.050151

The final model can be used by scouts and coaches to compare their talents. The scouts can use the model to compare young players with current talents to find promising players based on the physical tests. The coaches can use the model to compare talents within groups to find points of improvement for each talent. We will give a more extensive answer to the second and third sub-questions of how the scouts and coaches can actually put the model into practice in Chapter 6.

5.2.4 The results of predicting variables

As discussed in Section 5.1.2, many models are created and it might be possible that some models predict certain variables well. We checked this for each of the six variables in the handball case and the results are listed in the following paragraphs. For each variable we looked at the ten best performing models where the specific variable was not taken into account and we tested these models on the test set. We analyzed these models and we will give a conclusion about whether the model performs sufficiently or not as well.

Throwing when jumping

Figure 5.6 shows the performance on the training and test set of the ten best performing models to predict the throwing when jumping test. The x-axis represents the model codes again, where the part before the hyphen is the distance range and the part after the hyphen is the combination of variables taken into account when searching for the neighbors[1]. Two different combinations of features were found in the ten best performing models. They are the following:

- 12: Throwing when standing, vertical jump
- 32: Throwing when standing, vertical jump, long jump

From the figure it is clear that the performance on the training data is very similar for all models. Regarding the test set, we see that there are two models where the RMSE value is higher. These are the models with the lowest nearest neighbor distance ranges and variable combination 12. Therefore, we can say that variable combination 32 and a distance range between 1.14 and 1.21 is the best option to predict the throwing when jumping test with this split of the data in training data and test data and this nearest neighbors approach. If we apply the best model on the training data (*1.19-32*), on the complete data, we get an RMSE value of 4.59 km/h.

[1]This is also the case for the figures in the next paragraphs.

In Section 4.2.3 we calculated that the throwing when jumping test has values between 58.33 km/h and 105.00 km/h with a mean of 83.76 km/h.

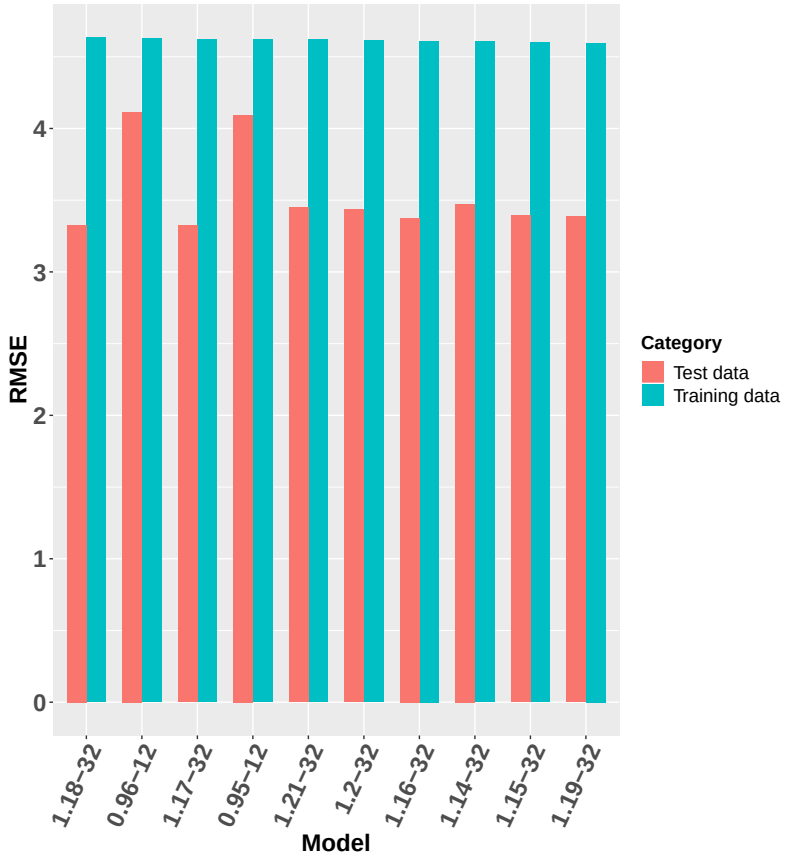

FIGURE 5.6: The RMSE on the training data and the test data to predict the throwing when jumping test with different models.

Throwing when standing

Figure 5.7 shows the performance on the training and test set of the ten best performing models to predict the throwing when standing test. Two different combinations of features were found in the ten best performing models. They are the following:

- 1: Throwing when jumping
- 10: Throwing when jumping, T-test

From the figure it is clear that the performance on the training data and the test data is very similar for all models. Although the models are very comparable, model *0.61-1* is quite different from the other models and the RMSE value on the training data of this model is a little worse than from the other models. Therefore we can say that variable combination 10 and a distance range between 1.04 and 1.12 is the best option to predict the throwing when standing test with this split of the data in training data and test data and this nearest neighbors approach. If we apply the best model on the training data (*1.08-10*), on the complete data, we get an RMSE value of 4.36 km/h.

In Section 4.2.3 we calculated that the throwing when standing test has values between 56.67 km/h and 96.00 km/h with a mean of 78.33 km/h.

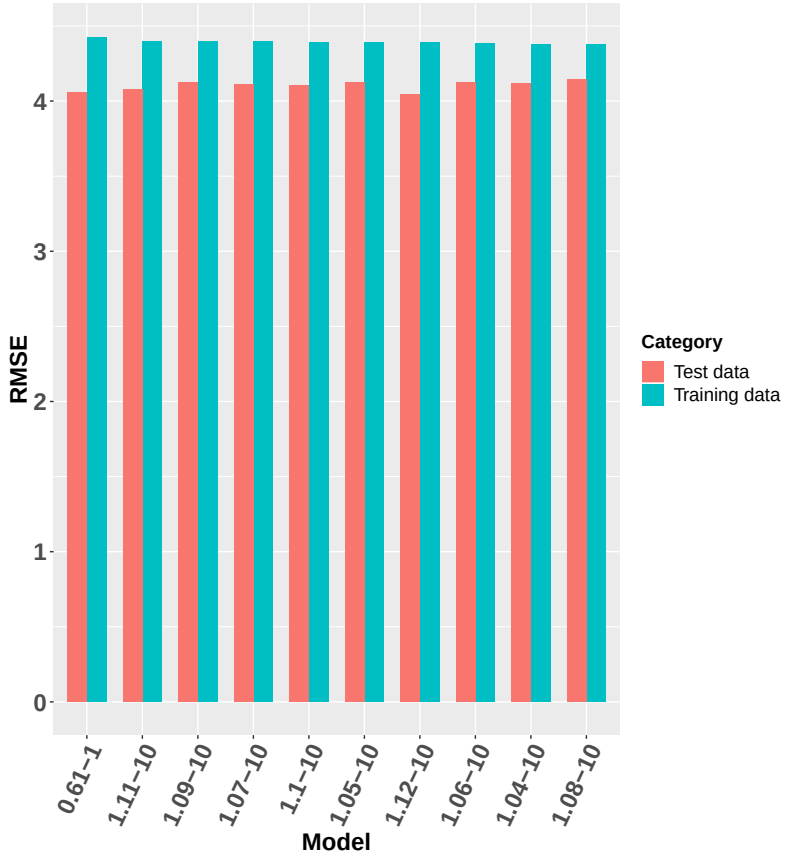

FIGURE 5.7: The RMSE on the training data and the test data to predict the throwing when standing test with different models.

Vertical jump

Figure 5.8 shows the performance on the training and test set of the ten best performing models to predict the vertical jump test. Two different combinations of features were found in the ten best performing models. They are the following:

- 41: Long jump, T-test, sprint
- 51: Throwing when jumping, long jump, T-test, sprint

From the figure it is clear that the performance on the training data is very similar for all models. Regarding the test set, we see that there are three models where the RMSE value is higher. These are the models with the lowest nearest neighbor distance ranges and variable combination 41. Therefore, we can say that variable combination 51 and a distance range between 1.67 and 1.82 is the best option to predict the vertical jump test with this split of the data in training data and test data and this nearest neighbors approach. If we apply the best model of these models on the training data (*1.82-51*), on the complete data, we get an RMSE value of 6.79 cm.

In Section 4.2.3 we calculated that the vertical jump test has values between 20.97 cm and 60.00 cm with a mean of 35.94 cm.

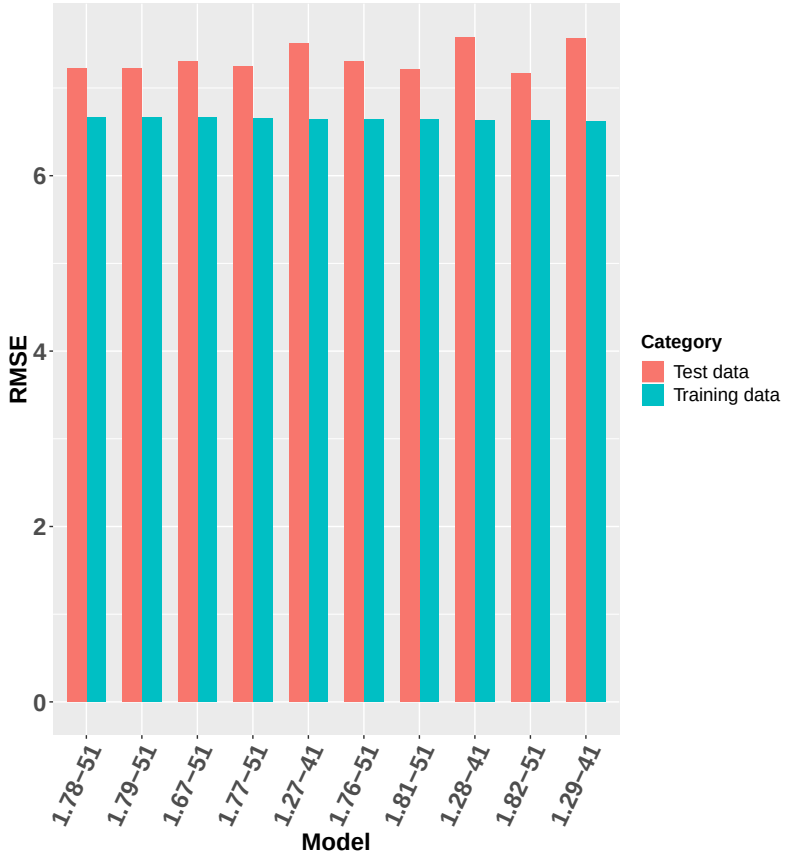

FIGURE 5.8: The RMSE on the training data and the test data to predict the vertical jump test with different models.

Long jump

Figure 5.9 shows the performance on the training and test set of the ten best performing models to predict the long jump test. Three different combinations of features were found in the ten best performing models. They are the following:

- 28: Throwing when jumping, vertical jump, sprint
- 34: Throwing when standing, vertical jump, sprint
- 44: Throwing when jumping, throwing when standing, vertical jump, sprint

From the figure it is clear that the performance on the training data is very similar for all models. Regarding the test set, we see that there are four models where the RMSE value is lower. These are the models with variable combination 28. Therefore, we can say that variable combination 28 and a distance range between 1.32 and 1.35 is the best option to predict the long jump test with this split of the data in training data and test data and this nearest neighbors approach. If we apply the best model of these models on the training data (*1.33-28*), on the complete data, we get an RMSE value of 20.09 cm.

In Section 4.2.3 we calculated that the long jump test has values between 155.67 cm and 267.50 cm with a mean of 214.44 cm.

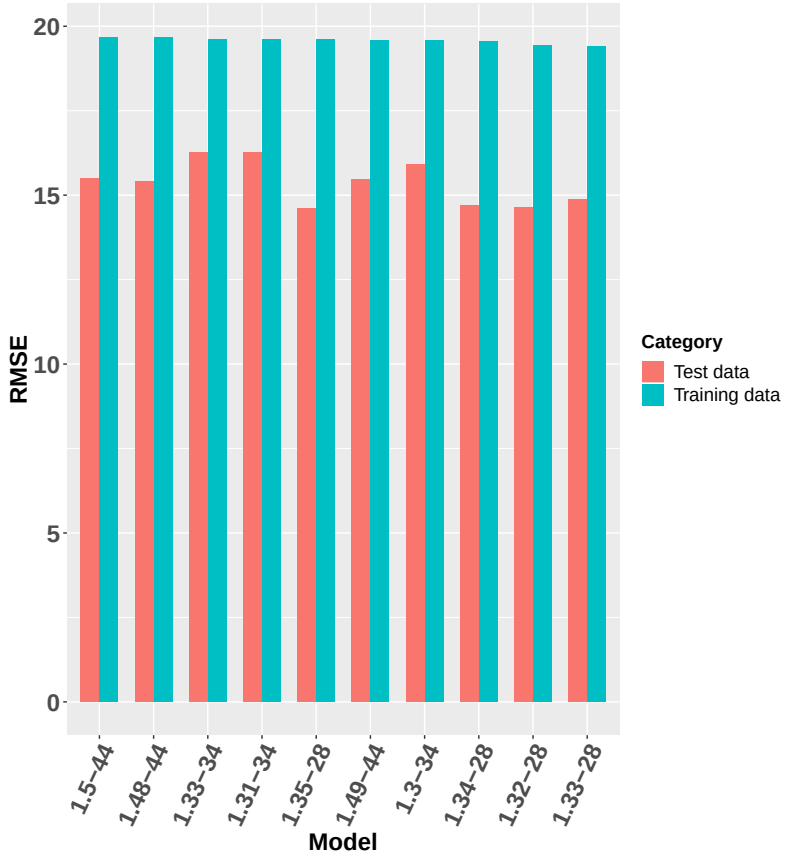

FIGURE 5.9: The RMSE on the training data and the test data to predict the long jump test with different models.

Sprint

Figure 5.10 shows the performance on the training and test set of the ten best performing models to predict the sprint test. Three different combinations of features were found in the ten best performing models. They are the following:

- 19: Long jump, T-test
- 38: Vertical jump, long jump, T-test
- 52: Throwing when standing, vertical jump, long jump, T-test

From the figure it is clear that the performance on the training data is very similar for all models. Regarding the test set, we see that there are five models where the RMSE value is the lowest. These are the models with variable combination 38. Therefore, we can say that variable combination 38 and a distance range between 1.33 and 1.37 is the best option to predict the sprint test with this split of the data in training data and test data and this nearest neighbors approach. If we apply the best model of these models on the training data (*1.37-38*), on the complete data, we get an RMSE value of 0.09 seconds.

In Section 4.2.3 we calculated that the sprint test has values between 2.92 seconds and 3.68 seconds with a mean of 3.22 seconds.

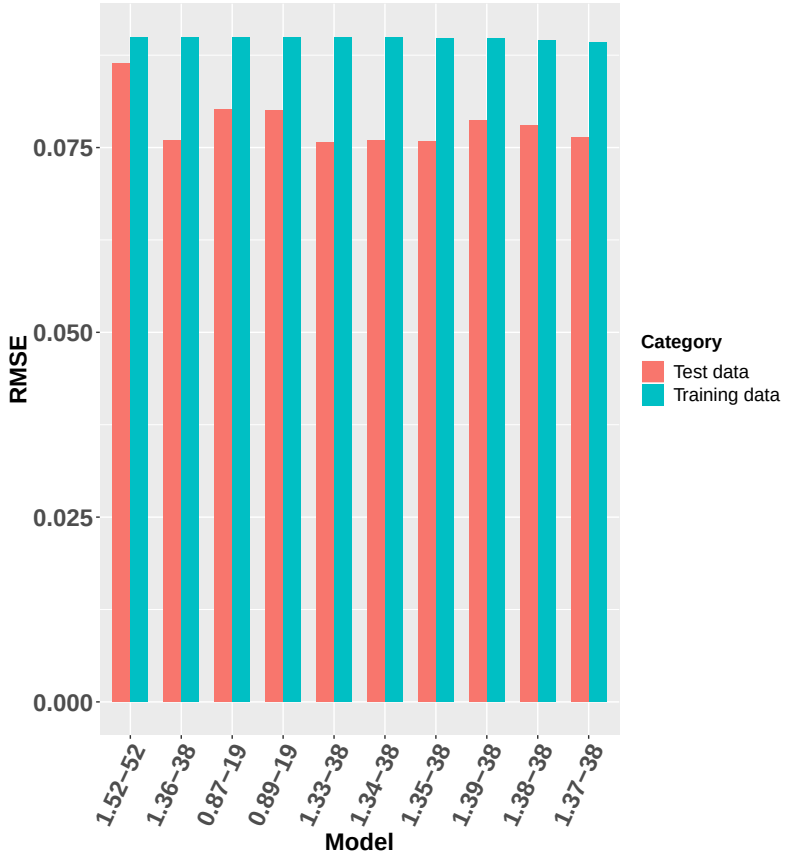

FIGURE 5.10: The RMSE on the training data and the test data to predict the sprint test with different models.

T-test

Figure 5.11 shows the performance on the training and test set of the ten best performing models to predict the T-test. Three different combinations of features were found in the ten best performing models. They are the following:

- 18: Vertical jump, sprint
- 34: Throwing when standing, vertical jump, sprint
- 39: Vertical jump, long jump, sprint

From the figure it is clear that the performance on the training data is very similar for all models. Regarding the test set, we see that there are six models where the RMSE value is the lowest. These are the models with variable combination 34. Therefore, we can say that variable combination 34 and a distance range between 1.30 and 1.36 is the best option to predict the T-test with this split of the data in training data and test data and this nearest neighbors approach. If we apply the best model of these models on the training data (*1.31-34*), on the complete data, we get an RMSE value of 0.21 seconds.

In Section 4.2.3 we calculated that the sprint test has values between 5.51 seconds and 7.19 seconds with a mean of 6.25 seconds.

FIGURE 5.11: The RMSE on the training data and the test data to predict the T-test with different models.

Conclusion

In the previous paragraphs we calculated the RMSE values to predict the separate physical tests. It is a challenge to give conclusions about whether these best models are really performing well enough to make good predictions. However, we can give some remarks on how the models perform compared to each other. Firstly, we saw that minimum, maximum and average value of the throwing when jumping and throwing when standing tests were bigger than the minimum, maximum and average value of the vertical jump test. In Section 4.2.3 we also saw that the standard deviations of these variables are comparable. The fact that the RMSE value of the model to predict the vertical jump (6.79) is quite a bit higher compared to the RMSE values of the models to predict the throwing when jumping test and the throwing when standing test (4.59, 4.36 resp.), could tell us that the vertical jump is harder to predict with a kNN model like this. Secondly, the RMSE value of predicting the T-test on the complete data set was 0.21. This is comparable to the results of estimating the T-test in Section 4.2.2 including the influential observations.

Although the performances of the models might not be that good, the predictions can still be used by the handball federation. For a certain talent in a certain team, it is possible to check how someone should approximately perform when compared to other talents in the team. Furthermore, as more data will be collected in the coming years, it is likely that the models will improve in accuracy.

Chapter 6

Visualization

The goal of this chapter is to answer the fourth sub-question: which way of presenting the information to the scouts and coaches will be effective? In the previous chapters we decided to compare talents on six different variables. This means that we have multivariate data. It can be a challenge to visualize these kinds of data without getting messy figures. Therefore, we will answer this sub-question by looking into the literature to find ways of visualizing multivariate data. We will discuss the results of this literature study in Section 6.1. Section 6.2 shows some examples of applying the visualization method to the handball case. With these examples, we will also give a more extensive answer to the second and third sub-questions.

6.1 Visualizing multivariate data

This section discusses one option of visualizing data observations of three or more variables, that was found in other studies. We will discuss this option, the spider plot, in Section 6.1.1.

6.1.1 Spider plot

In this section we will discuss the spider plot or the radar chart. There are studies where spider plots are used to do comparisons. We will describe the use of the spider plot by giving three examples of studies where spider plots were used. The first research was about comparing performances of several aspects of airplanes. This study was done by Joshi, Tidwell, Crossley, and Ramakrishnan (2004). The second research was about comparing several sensory attributes in different beers. This study was done by Vázquez-Araújo, Parker, and Woods (2013). The last study we looked into is a research about comparing health care aspects in different groups. This research was done by Saary (2008). All three publications have in common that they used a spider plot to visualize single groups or data points to visualize their data.

Saary (2008) describes the radar plot as a circular graph with a spoke from the middle for each variable. Each variable can have its own scale. If a data point is plotted on the spokes, it is possible to connect the dots of one observation to create a polygon to be able to clearly superpose different observations in the circular graph. Next to plotting single observations, it is also possible to plot means of groups of observations for example. An example of a spider plot with several observations is shown in Figure 6.1. In the figure, three different students are compared on five different subjects. The corners of the plot represent maximum values of the subjects.

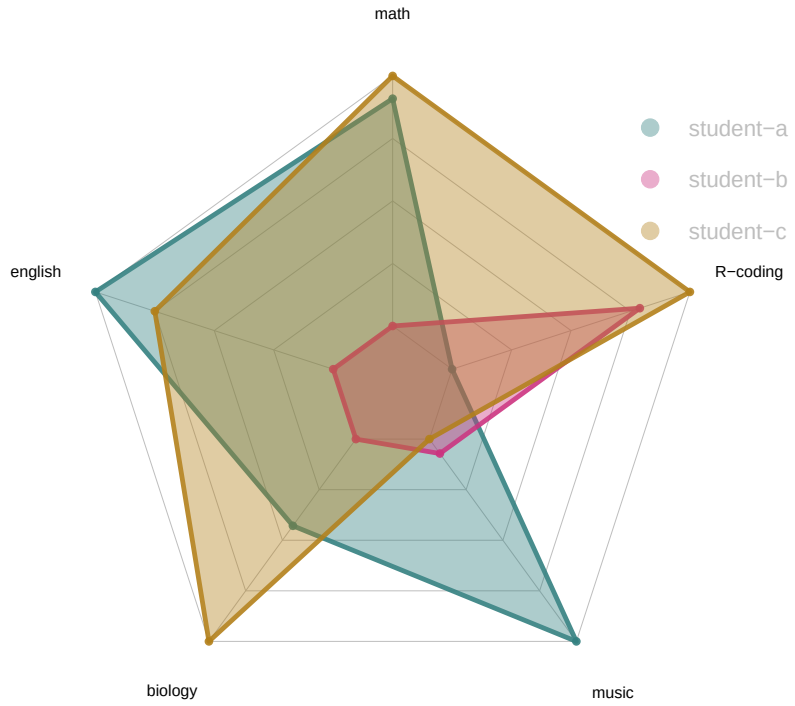

FIGURE 6.1: Spider plot example.

6.2 Visualizing the handball data

In Section 6.1 we discussed a way to visualize multivariate data points; the spider plot. In the following sections we will give a few examples of how this plot can be used in the handball case. In the plots in the different sections, the pink plot is always the player we compared to some other talents. Furthermore, the talent labeled with "1" is always the talent the player looks most alike and the talent labeled with "2" is the talent that the player looks second most alike and so on. The maximum and minimum values of the axes are the maximum and minimum values of the group in which we compare talents.

6.2.1 Comparing new players with current talents

In handball as well as in other sports, scouts look for promising players to add them to their selection or to train them to get ready to be in their selection. Figure 6.2 shows an example in which we compared one player with all 141 talents in the current handball talents data set using the model parameters determined in Section 5.2.3. There are three talents within the distance range of 1.87 from the potential talent. It is clear that this player performs similar to these talents. This is obvious, since we look for similar talents in the table with quite some talents. We also see that the player is most similar to the talent performing best on the vertical jump test (compared to the talents drawn in the chart). Lastly, we can see that on most of the axes, the player performs a little above average, or even best on the T-test. A scout might conclude from this that the player is worth selecting. However, a scout might

also want to create another plot where the player is compared to the whole group for example, to really see their position within the complete set of handball talents.

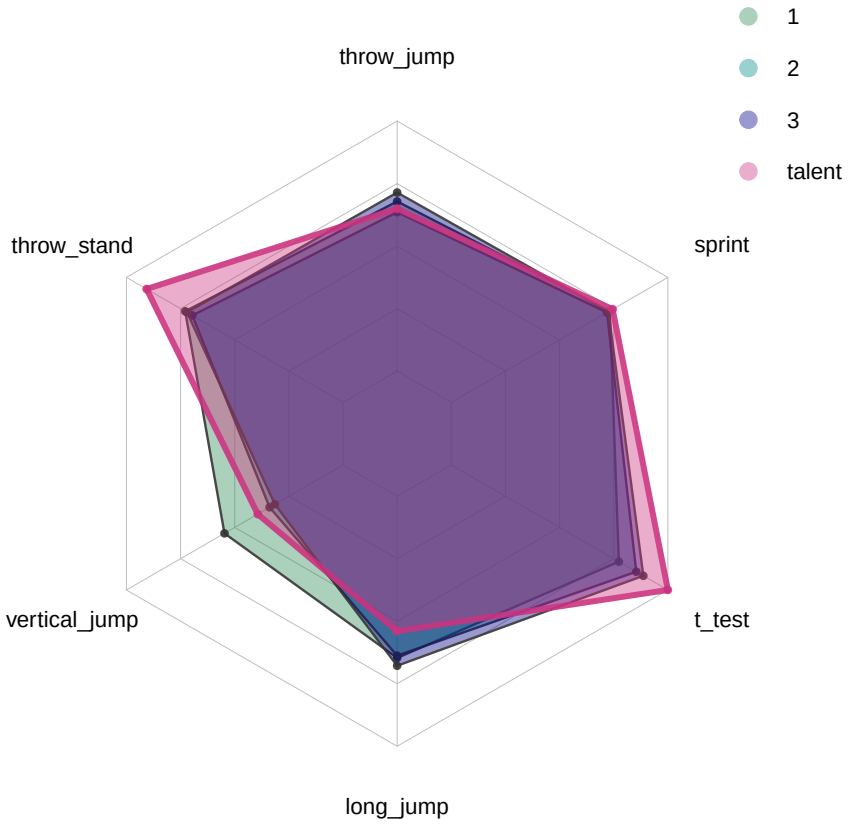

FIGURE 6.2: Comparing a new player with current talents.

6.2.2 Comparing talents within a group

A second example is one more for coaches within a certain team. In Figure 6.3 we compared one talent in the A-selection to the other talents in the A-selection using the model parameters determined in Section 5.2.3. The differences between the talents seem to be a bit bigger than in the previous section. There can be a few reasons for this. Firstly, we only look at one selection, which means we have less talents to compare with. Secondly, since we look at the A-selection only, it is likely that the maximum and minimum values of the axes are closer to each other. When we look at the positions of the points on the axes, we see that the talents perform quite on average. When we look at the pink talent we compared specifically, we see that she could mainly improve on two points; the T-test and the vertical jump. The coach might decide to specifically work on that with the talent.

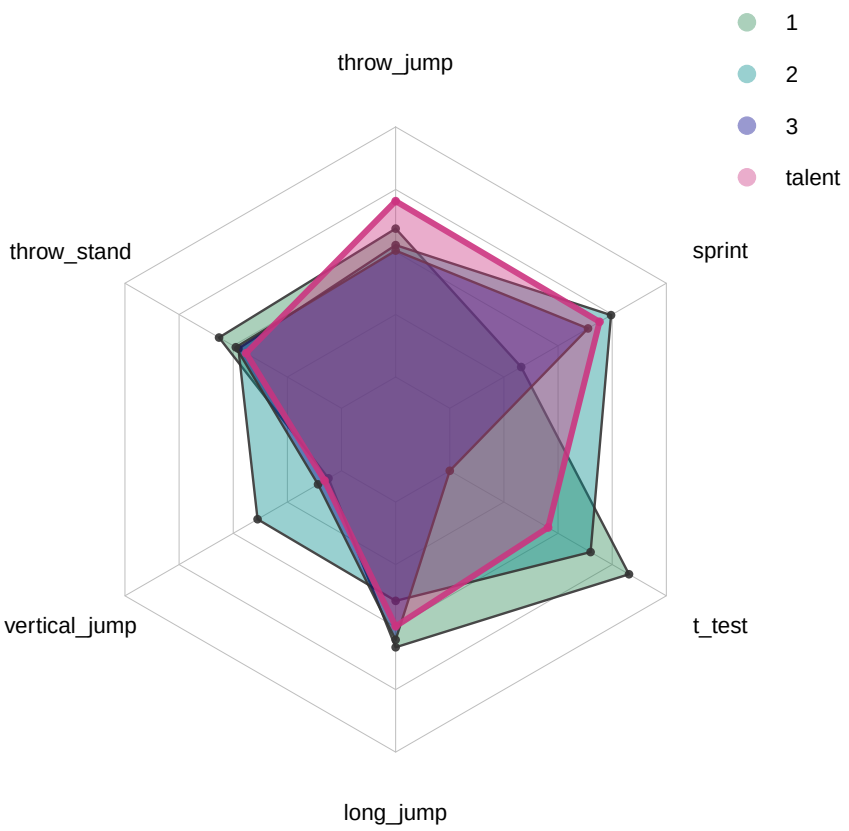

FIGURE 6.3: Comparing a talent with talents within the same group.

6.2.3 Comparing talents with groups

In the previous paragraphs we compared single talents with other single talents. However, since we are using a neighbors distance range, it is possible that the number of comparable talents is big. A spider plot will become unclear when too many different observations are superposed. Therefore, it is recommended to limit the number of talents in the plot. We propose this number to be no more than four or five talents. A way to still compare one talent to other talents in a group, is by creating segments in the group. Figure 6.4 shows an example where one talent is compared to a group split in three segments. The maximum segment is determined by looking at the 33% of best performing players, the minimum segment is determined by looking at the 33% of least performing players and the average segment is determined by looking at the 33% of average performing players. We can see that the single talent perform similar to the maximum segment regarding the throwing when jumping test and the sprint test. Regarding the throwing when standing, the T-test and the vertical jump test, the talent seems to perform similar to the minimum segment. Lastly, regarding the long jump, the talent performs quite on average. From a plot like this we can conclude that the talent could start to improve on some of the physical tests to get closer to the average or maximum segment.

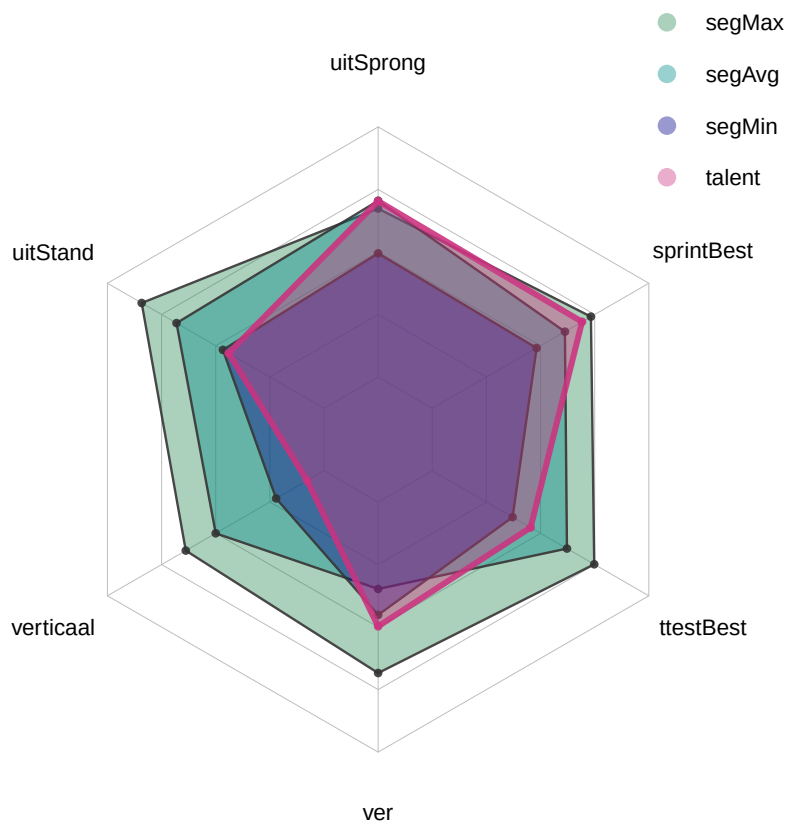

FIGURE 6.4: Comparing a talent with a group.

6.2.4 Comparing talents with talents in specific positions

The last example of the spider plot is one where we compare talents that are within the same position in the field. Figure 6.5 gives an example where one talent is compared to talents that play in the same position. Again, we used the model parameters we found in the previous chapter. We can see that the superposed shapes are a little similar. This might indicate that this is typical for the certain position. However, we are not sure about this until we compare this to other positions.

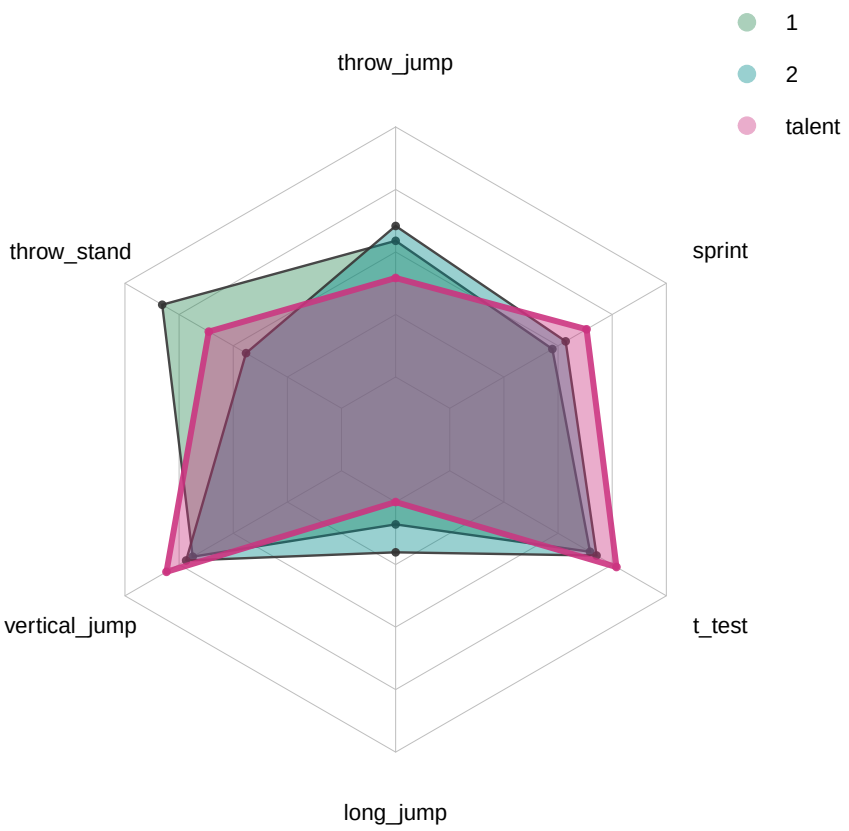

FIGURE 6.5: Comparing a talent with talents in the same position in the field.

Chapter 7

Conclusion and discussion

This research was divided in three main phases. In this chapter we will first describe the main findings in each phase in Section 7.1. Secondly, we will discuss the limitations of the research and possible future research in Section 7.2.

7.1 Conclusion

In the previous chapters, we discussed how to get insights from limited data from sports talents. By doing this, we tried to answer the main research question:

Main: *How can data analytics be used to assist scouts and coaches in finding and training sports talents in order to make more effective choices?*

We answered this question by looking at three main phases corresponding to four sub-questions. For each phase we proposed a method to handle the data. We created these methods by looking into data from the handball case. Furthermore, we applied the methods to the handball data to test them. This section will describe the main findings for each phase.

1: *What data pre-processing needs to be done before the data analytics process can start?*

In this phase we proposed a method to fill in values of variables that have a lack of data, but that are still considered important by the domain expert. By trying many different models, we found that performing multiple linear regression performs better when one of the independent variables is the cluster class of an observation when the data was first clustered using k-means. Furthermore, the final model improved when the influential observations were removed. Although we considered the final model to be performing sufficiently with this limited data for the T-test, we also found that this method was not performing sufficiently with the YO-YO test, which had even more missing values. Therefore, we conclude from this research and from this case, that the method could be useful in other cases as well, but that the performance depends on the type of variable and that it might depend on the number of missing values.

2: *How can scouts be supported when looking for the best performing sports players?* and *How can coaches be supported when training their sports talents?*

We answered these two sub-questions by finding a method to find the best model to compare talents. The goal of the method we proposed, is to find comparable talents and not to make very precise predictions, since this is a big challenge with the limited data. The method we proposed is based on nearest neighbors regression with a distance range to find the neighbors. With the method we tried to find the best

distance range and variable combination to compare talents with. The best model to compare talents, can be useful for both the scouts and the coaches. Young talents can be found by comparing them with current talents, and talents within teams can be compared to find points of improvement for each talent. Although this method is already applicable to data sets with a limited number of rows, it is also easy to adjust the model and make even better comparisons when more data is collected and more talents are included in the data set. Furthermore, since the performance of many models is discovered in the process, we also propose a way of using this model base to find models that can predict certain variables well. In the handball case we found that some variables are easier to predict with this method than others. We also found that the method works comparable to the multiple linear regression model with the cluster class and without removing the influential observations to predict the T-test from the first phase. The predictions can now be used to see how a talent should approximately perform compared to their comparable talents. As more data is collected in the future, it is likely that the accuracy of the models improve.

3: *Which way of presenting the information to the scouts and coaches will be effective?*

For the third phase we recommended to use a spider plot as a way of visualizing the performance of talents. With this plot, it is easy to superpose the performance of a few talents to see their similarities, but also their differences. In this phase we saw that this visualization can be helpful for both scouts and coaches to see this in one glance. However, we also stated that the number of plotted talents should not be too big, since this would create messy charts. Therefore, we proposed that if a talent is compared to more than 4 talents, the averages of these talents is plotted as one group. Furthermore, if a talent is compared to a certain group, we recommend to divide this group into three segments, where one segment represents the best talents, another segment represents the average talents, and the last segment represents the least performing talents. This can be done within teams, but also for a certain position for example.

7.2 Discussion

In this section we will first discuss limitations of this research in Section 7.2.1. Furthermore, we will give ideas of how this research can be extended in Section 7.2.2.

7.2.1 Limitations

This study has a few limitations. Our goal was to find ways of creating insights with limited data, and we accomplished this by finding a method to compare talents. However, during the research we found that the data is still quite limited to make real precise predictions for some variables. We believe that this is due to the data having not too many observations that are complete. Therefore, we also had to limit the case study to female handball talents only. However, we do not consider this last fact to be a big problem, since we could still test all methods with this data. Lastly, also due to the limited data we only looked into physical tests to compare talents. However, as also described in Section 3.3, to really get a complete picture, other aspects like mental skills could be taken into account as well.

Other limitations of this study mainly concern the fact that we did not test the methods several times. In the first phase we tested the method for two variables

and in the second phase we did not check whether the approach of using a distance range to find the neighbors instead of a fixed number of neighbors actually works better in this case. Lastly, when answering the last sub-question, we discussed the proposed visualization method with one domain expert.

7.2.2 Future research

We discussed the limitations in the previous section. To avoid those limitations, some future research possibilities will be discussed in this section.

Regarding the limitations about not testing the methods multiple times from the previous section, there are possibilities for further research. We could for example test the method of filling the missing values for other variables as well and we can compare results from using a distance range or a fixed number of neighbors to support the methods we proposed in phase 2. Another idea for the future might be to test the complete method of finding models to compare talents with on another data set. This could for example be a data set of another upcoming sport. Lastly, it is possible to do a survey or interviews with multiple scouts and coaches.

Moreover, in the first phase, we chose some important variables together with the domain expert. However, we did not check whether these variables are, according to the data, indeed important indicators of someone being a real talent or not. In the future, a way to find this out could be investigated.

Appendix A

The *R*-packages used during the research

TABLE A.1: The *R*-packages used during the research.

Package	Version	Use
dplyr	0.7.8	Manipulating data frames
lubridate	1.7.4	Dealing with dates
olsrr	0.5.1	Determining the Cook's distance
e1071	1.7-0	Determining the skewness of a variable
FNN	1.1	Finding the *k* nearest neighbors
reshape	0.8.7	Transforming the data for bar plots
caret	6.0-80	Applying train control for cross validation and calculating the RMSE
fractal	2.0-4	Finding the nearest neighbors within a certain distance
clusterSim	0.47-1	Calculating the DB index
MLmetrics	1.1.1	Calculating the RRSE
car	3.0-0	Calculating different test values to check regression assumptions
fmsb	0.6.3	Creating spider plots

Appendix B

The data

This section shows what kind of data is saved in each of the tables, and how the tables are structured. The following data tables were collected during the project:

- `Talents` - 3,440 rows, 18 columns.
- `Assessment` - 637 rows, 46 columns.
- `Measurement_types` - 885 rows, 42 columns.
- `Measurement_results` - 8,641 rows, 38 columns.
- `Training_types` - 1,114 rows, 6 columns.
- `Training_logs` - 140,841 rows, 16 columns.
- `Profile_of_mood_states` - 17,718 rows, 15 columns.

The data tables form a database with a logical structure as modeled in Figure B.1.

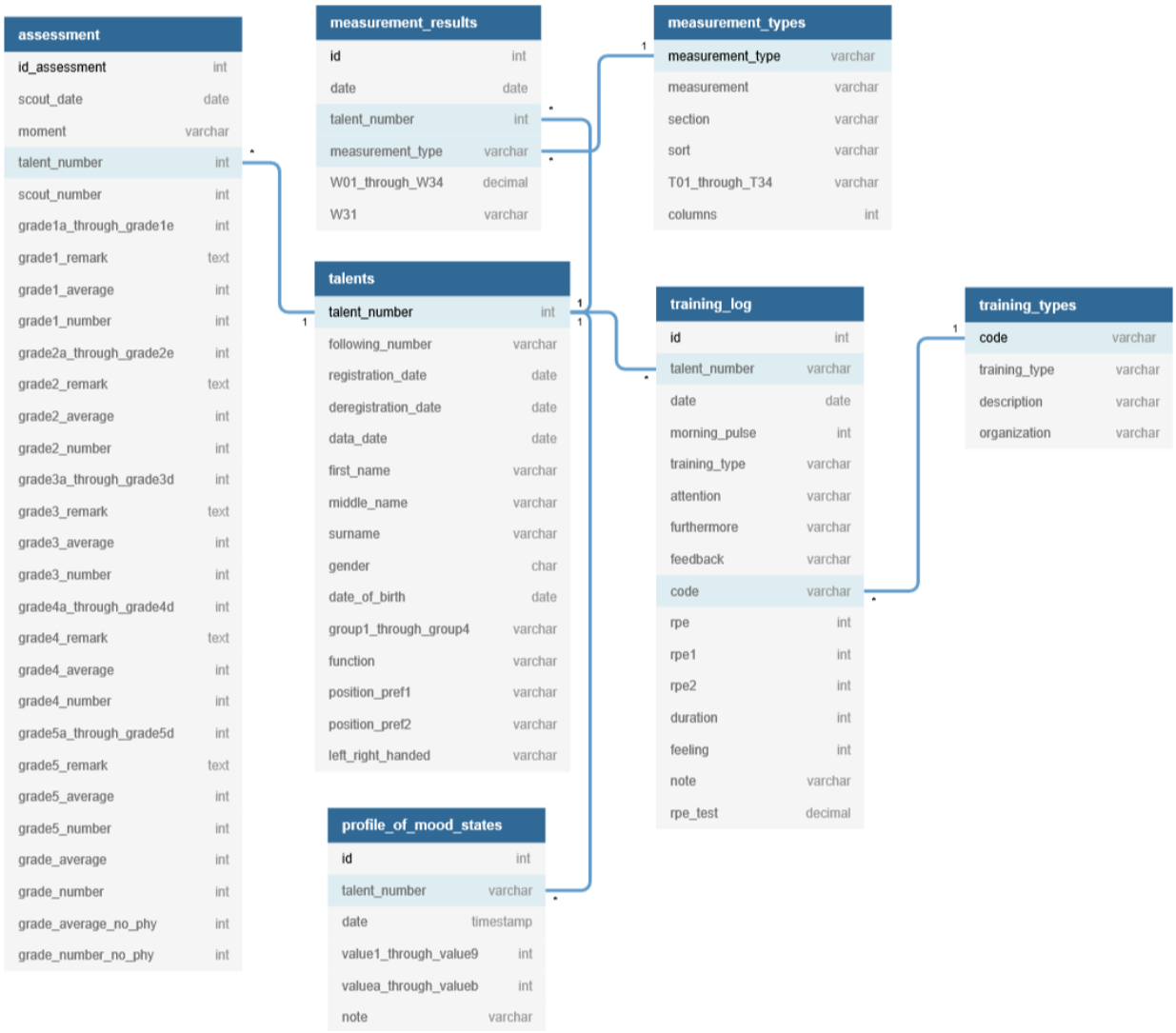

FIGURE B.1: Logical database structure.

Appendix C

Linear regression summary and assumptions

C.1 Multiple linear regression without the influential observations

In this section we present the regression summary and we will check the regression assumptions of the multiple linear regression model where we excluded the influential observations.

C.1.1 Regression summary

Table C.1 shows the regression coefficients and the p-values of the multiple regression model where we included the influential observations. Table C.2 presents the R-squared and the p-value of the full regression model.

TABLE C.1: Coefficients and p-values of the variables in the regression model.

Variable	Coefficient	p-value
Intercept	6.23828	$< 2 \times 10^{-16}$
Vertical jump	-0.06767	3.79×10^{-3}
Sprint	0.13497	1.32×10^{-7}
Sprinting age category	-0.09520	1.23×10^{-3}
Cluster	0.09543	2.10×10^{-5}

TABLE C.2: P-value and R-squared of the full model.

Measure	Value
P-value	2.2×10^{-16}
R-squared	0.6711

C.1.2 Regression assumptions

In this section we will check the linear regression assumptions of the multiple linear regression model without the influential observations from Section 4.2.2. First we will check the multicollinearity phenomenon in this model. We calculated the variance inflation factors (VIF) for the independent variables in the model using *R*. They are the following:

- Vertical jump: 1.568425

- Sprint: 1.640513
- Sprinting age category: 1.054328
- Cluster: 1.018185

All factors are clearly below 5. Therefore, we can assume that there is no multicollinearity in the model.

In the next sections, we will use the plots and tests described in Section 3.1.1 to validate the assumptions.

C.1.3 Homoscedasticity

Figure C.1 shows the residuals against the fitted values plot. If the homoscedasticity assumption is met, the data points should be equally spread around the $y = 0$ line. There might be a little variance as shown by the red line, but not too much. This assumption can also be checked using the Breusch-Pagan test which we calculated using *R*. The null hypothesis of this test is that there is constant variance, and therefore homoscedasticity. The p-value we got from this test was 0.5515213, meaning that we can not reject the null hypothesis. Together with the conclusion from the figure that there might be a little variance but not too much, we assumed that the homoscedasticity assumption is satisfied.

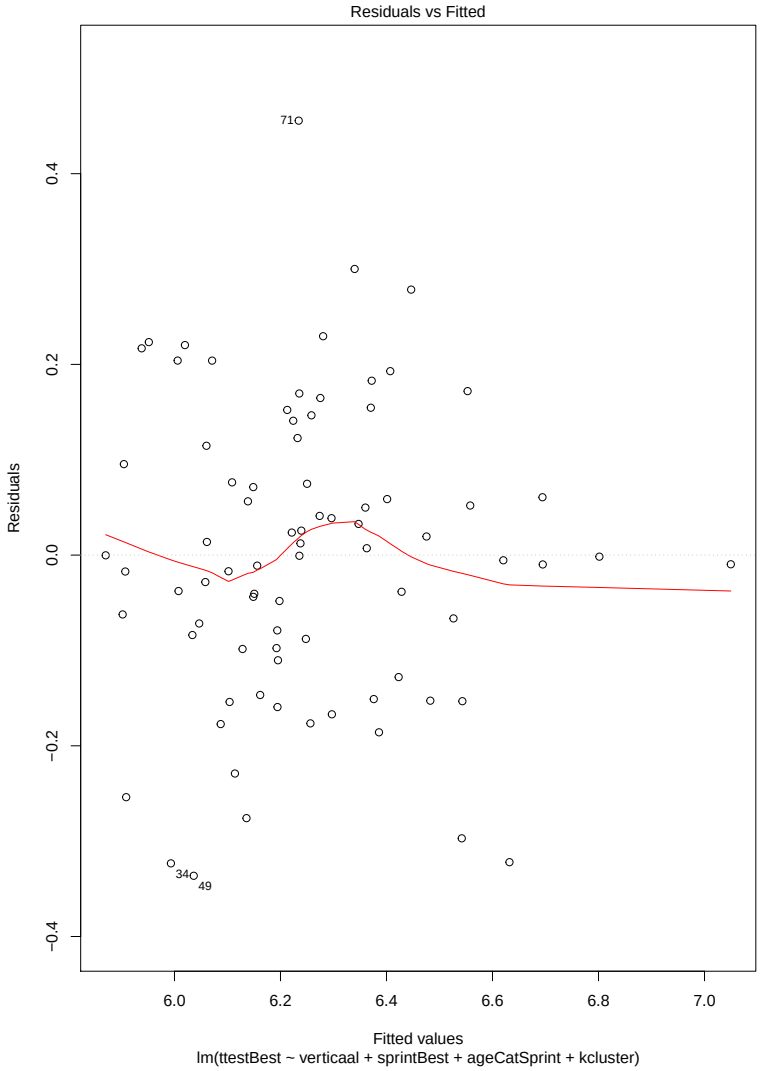

FIGURE C.1: Residuals vs. fitted values - linear model without influential observations.

C.1.4 No extreme values

In Figure C.1 we see that there are a few deviating values, but that overall the residuals are more close to each other. This assumption can also be checked using the Bonferroni p-values. We calculated these p-values using *R*. For each residual, the null hypothesis is that the residual is not an outlier. The result of using this test is that there were no residuals with a Bonferroni p-value smaller than 0.05. This means that this assumption is satisfied.

C.1.5 Normally distributed residuals

Figure C.2 shows the Quantile-Quantile plot of the residuals. The assumption is met when the residuals are close to the diagonal line. This is the case in this example. The normality assumption can also be substantiated by looking at the histogram with the distribution of the residuals in Figure C.3. In this figure we also see that he residuals are normally distributed. Therefore, this assumption is satisfied.

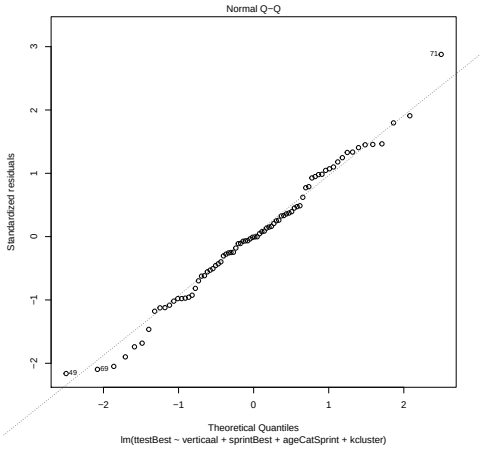

FIGURE C.2: Q-Q plot - linear model without influential observations.

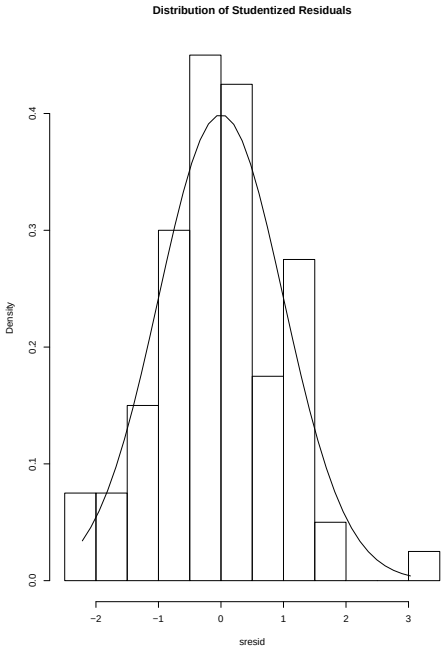

FIGURE C.3: Distribution of the residuals - linear model without influential observations.

C.1.6 The residuals are not related to the independent variables

In Figure C.1 we do not see any relationships or patterns. Therefore the assumption that the residuals are not related to the independent variables is satisfied.

C.1.7 The residuals are not correlated with each other

This assumption can be checked using the Durbin Watson test. The null hypothesis of this test is that the residuals are not correlated. The p-value we got from this test was 0.82, meaning that we can not reject the null hypothesis. This means that this assumption is satisfied.

C.2 Multiple linear regression including the influential observations

In this section we present the regression summary and we will check the regression assumptions of the multiple linear regression model where we included the influential observations.

C.2.1 Regression summary

Table C.3 shows the regression coefficients and the p-values of the multiple regression model where we included the influential observations. Table C.4 presents the R-squared and the p-value of the full regression model.

TABLE C.3: Coefficients and p-values of the variables in the regression model, including the influential observations.

Variable	Coefficient	p-value
Intercept	6.57899	$< 2 \times 10^{-16}$
Throwing when standing	-0.13908	2.52×10^{-4}
Throwing when jumping	0.15075	4.73×10^{-4}
Sprint	0.17429	5.1×10^{-10}
Cluster	-0.19955	3.37×10^{-3}

TABLE C.4: P-value and R-squared of the full model, including the influential observations.

Measure	Value
P-value	5.528×10^{-15}
R-squared	0.5515

C.2.2 Regression assumptions

In this section we will check the linear regression assumptions of the multiple linear regression model including the influential observations from Section 4.2.2. First we will check the multicollinearity phenomenon in this model. We calculated the variance inflation factors (VIF) for the independent variables in the model using *R*. They are the following:

- Throwing when standing: 2.642591
- Throwing when jumping: 3.426829
- Sprint: 1.241279
- Cluster: 1.966819

All factors are below 5. Therefore, we can assume that there is no multicollinearity in the model.

In the next sections, we will use the plots and tests described in Section 3.1.1 to validate the assumptions.

C.2.3 Homoscedasticity

Figure C.4 shows the residuals against the fitted values plot. If the homoscedasticity assumption is met, the data points should be equally spread around the $y = 0$ line. There might be a little variance as shown by the red line, but not too much. This assumption can also be checked using the Breusch-Pagan test which we calculated using *R*. The null hypothesis of this test is that there is constant variance, and therefore homoscedasticity. The p-value we got from this test was 0.8223801, meaning that we can not reject the null hypothesis. Together with the conclusion from the figure that there might be a little variance but not too much, we assumed that the homoscedasticity assumption is satisfied.

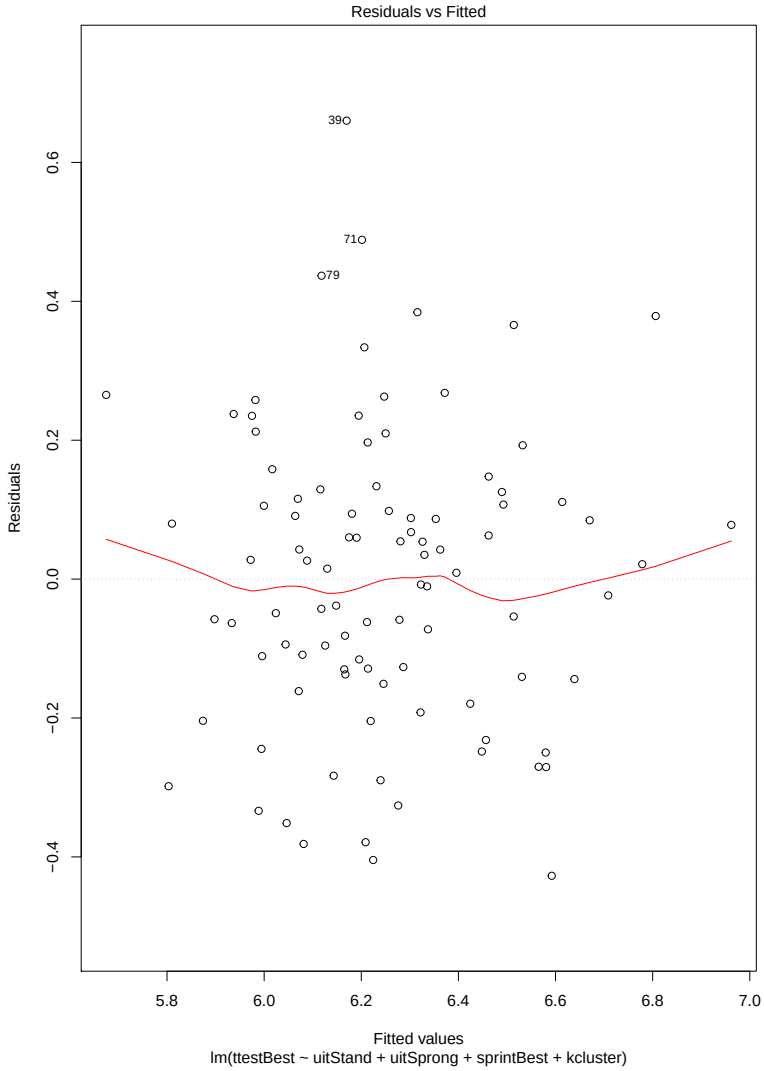

FIGURE C.4: Residuals vs. fitted values - linear model including influential observations.

C.2.4 No extreme values

In Figure C.4 we see that there are a few deviating values, but that overall the residuals are more close to each other. This assumption can also be checked using the Bonferroni p-values. We calculated these p-values using *R*. For each residual, the null hypothesis is that the residual is not an outlier. The result of using this test is that there were no residuals with a Bonferroni p-value smaller than 0.05. This means that this assumption is satisfied.

C.2.5 Normally distributed residuals

Figure C.5 shows the Quantile-Quantile plot of the residuals. The assumption is met when the residuals are close to the diagonal line. This is the case in this example. The normality assumption can also be substantiated by looking at the histogram with the distribution of the residuals in Figure C.6. In this figure we also see that he residuals are normally distributed. Therefore, this assumption is satisfied.

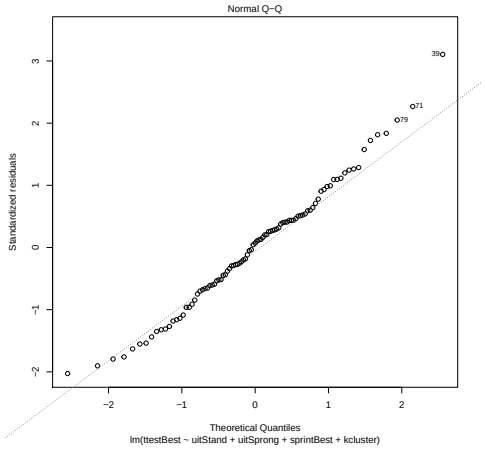

FIGURE C.5: Q-Q plot - linear model including influential observations.

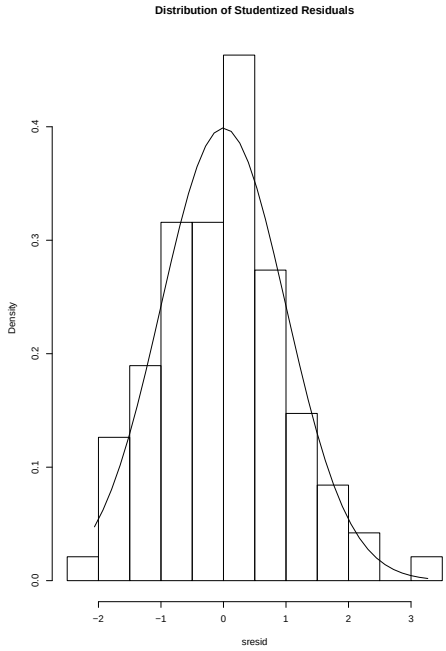

FIGURE C.6: Distribution of the residuals - linear model including influential observations.

C.2.6 The residuals are not related to the independent variables

In Figure C.4 we do not see any relationships or patterns. Therefore the assumption that the residuals are not related to the independent variables is satisfied.

C.2.7 The residuals are not correlated with each other

This assumption can be checked using the Durbin Watson test. The null hypothesis of this test is that the residuals are not correlated. The p-value we got from this test was 0.198, meaning that we can not reject the null hypothesis. This means that this assumption is satisfied.

Appendix D

The final data table for analysis

TABLE D.1: Descriptive statistics of the physical tests and ages in the final data table.

Test	Minimum	Maximum	Mean	Median	Standard deviation	Skewness
Age throwing	13.00	21.00	17.06	17.00	1.6310462	-0.08806145
Throw jumping (km/h)	58.33	105.00	83.76	84.67	8.6767791	-0.29969420
Throw standing (km/h)	56.67	96.00	78.33	79.00	7.8764698	-0.42862161
Age jumping	13.50	26.00	17.41	17.17	2.1449106	1.07822915
Vertical jump (cm)	20.97	60.00	35.94	34.90	8.0421036	0.56961147
Long jump (cm)	155.67	267.50	214.44	218.00	24.9381297	-0.18991300
T-test (s)	5.51	7.19	6.25	6.23	0.2976720	0.3103657
Age sprinting	14.00	26.00	17.35	17.00	2.1501942	1.18216100
Sprint 20m (s)	2.92	3.68	3.22	3.21	0.1301325	0.59357458

TABLE D.2: The number of talents in each age category of each physical test type.

Age category	#1	#2	#3
Throwing	74	59	8
Jumping	69	53	19
Sprinting	75	47	19

Bibliography

Abdi, H. (2010). "Normalizing data". In: *Neil Salkind (Ed.), Encyclopedia of Research Design*, pp. 1–4.

Adeboye, N.O, I. S Fagoyinbo, and T.O Olatayo (2014). "Estimation of the Effect of Multicollinearity on the Standard Error for Regression Coefficients". In: *IOSR Journal of Mathematics* 10.4, pp. 16–20. ISSN: 2319765X. DOI: 10.9790/5728-10411620.

Alexander, D. L. J., A. Tropsha, and D. A. Winkler (2015). "Beware of R2: simple, unambiguous assessment of the prediction accuracy of QSAR and QSPR models". In: *J. Chem. Inf. Model.* 55.7, pp. 1316–1322. DOI: doi:10.1021/acs.jcim.5b00206.

Alibuhtto, M. C. and T. S. G. Peiris (2015). "Principal Component Regression for Solving Multicollinearity Problem". In: pp. 231–238. URL: http://www.seu.ac.lk/researchandpublications/symposium/5th/pureandappliedsciences/29.pdf.

Anguita, D et al. (2009). "K-Fold Cross Validation for Error Rate Estimate in Support Vector Machines". In: *International Conference on Data Mining*.

Bangsbo, Jens, F Marcello Iaia, and Peter Krustrup (2008). "The Yo-Yo Intermittent Recovery Test: A Useful Tool for Evaluation of Physical Performance in Intermittent Sports". In: *Medicine & Science in Sports & Exercise* 38.1, pp. 37–51. DOI: 10.2165/00007256-200838010-00004.

Breusch, Author T S and A R Pagan (1979). "A Simple Test for Heteroscedasticity and Random Coefficient Variation". In: *Econometrica* 47.5, pp. 1287–1294.

Chai, T. and R. R. Draxler (2014). "Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature". In: *Geoscientific Model Development* 7.3, pp. 1247–1250. ISSN: 19919603. DOI: 10.5194/gmd-7-1247-2014.

Cook, Dennis R and Sanford Weisberg (1982). *Residuals and Influence in Regression*. Ed. by D R Cox and D V Hinkley. New York: Chapman and Hall. ISBN: 041224280X.

Dickerson, Matthew T and R Scot Drysdale (1990). "Fixed-Radius Near Neighbors and Segments". In: *Information Processing Letters* 35.5, pp. 269–273.

Durbin, B Y J and G S Watson (1950). "Testing for Serial Correlation in Least Squares Regression". In: *Biometrika* 37.3, pp. 409–428.

Elliot, Mark and Mark Tranmer (2008). "Multiple Linear Regression". In: *The Cathie Marsh Centre for Census and Survey Research (CCSR)*, pp. 1–47.

Galdi, Paola and Roberto Tagliaferri (2019). "Data Mining: Accuracy and Error Measures for Classification and Prediction". In: *Encyclopedia of Bioinformatics and Computational Biology* 1, pp. 431–436. DOI: 10.1016/b978-0-12-809633-8.20474-3.

Goyal, Rinkaj, Pravin Chandra, and Yogesh Singh (2014). "Suitability of KNN Regression in the Development of Interaction based Software Fault Prediction Models". In: *IERI Procedia* 6, pp. 15–21. ISSN: 22126678. DOI: 10.1016/j.ieri.2014.03.004. URL: http://dx.doi.org/10.1016/j.ieri.2014.03.004.

HandbalNL (2019). "Nederlands Herenteam op miraculeuze wijze voor het eerst naar EK". In: *HandbalNL*. URL: https://www.handbal.nl/blog/2019/06/16/nederlands-herenteam-op-miraculeuze-wijze-voor-het-eerst-naar-ek/.

Harman, Everett and John Garhammer (2008). "Administration, Scoring, and Interpretation of Selected Tests". In: *Essentials of Strength Training and Conditioning*. Ed. by Thomar R. Baechle and Roger W. Earle. 3rd ed. Human Kinetics Publishers. Chap. 12, pp. 249–292. ISBN: 9780736058032. DOI: 10.1016/s0031-9406(05)66120-2.

Healy, M. J. R. (1984). "The Use of R 2 as a Measure of Goodness of Fit". In: *Journal of the Royal Statistical Society. Series A (General)* 147.4, pp. 608–609. ISSN: 00359238. DOI: 10.2307/2981848.

Imandoust, Sadegh Bafandeh and Mohammad Bolandraftar (2013). "Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events: Theoretical Background". In: *International Journal of Engineering Research and Applications* 3.5, pp. 605–610.

Jain, A. K. (2010). "Data clustering: 50 years beyond K-means". In: *Pattern Recognition Letters* 31.8, pp. 651–666. ISSN: 01678655. DOI: 10.1016/j.patrec.2009.09.011. URL: http://dx.doi.org/10.1016/j.patrec.2009.09.011.

Joshi, Shiv P et al. (2004). "Comparison of Morphing Wing Strategies Based Upon Aircraft Performance Impacts". In: *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference* April, pp. 1–7. DOI: 10.2514/6.2004-1722.

Lemmink, K. A.P.M., R. Verheijen, and C. Visscher (2004). "The discriminative power of the Interval Shuttle Run Test and the Maximal Multistage Shuttle Run Test for playing level of soccer". In: *Journal of Sports Medicine and Physical Fitness* 44.3, pp. 233–239. ISSN: 00224707.

Maulik, Ujjwal and Sanghamitra Bandyopadhyay (2002). "Performance evaluation of some clustering algorithms and validity indices". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.12, pp. 1650–1654. ISSN: 01628828. DOI: 10.1109/TPAMI.2002.1114856.

Meer, Tom Van der, Manfred Te Grotenhuis, and Ben Pelzer (2010). "Influential Cases in Multilevel Modeling: A Methodological Comment". In: *American Sociological Review* 75.1, pp. 173–178. ISSN: 0003-1224. DOI: 10.1177/0003122409359166.

Mohamed, Hasan et al. (2009). "Anthropometric and performance measures for the development of a talent detection and identification model in youth handball". In: *Journal of sports sciences* 27.3, pp. 257–66. ISSN: 0264-0414. DOI: 10.1080/02640410802482417. URL: http://www.ncbi.nlm.nih.gov/pubmed/19153859.

Montgomery, Douglas C., Elizabeth A. Peck, and Geoffrey G. Vining (2015). *Introduction to Linear Regression Analysis*. 5th ed. Hoboken, New Jersey: John Wiley & Sons, Inc. ISBN: 9780470542811. DOI: 10.1111/insr.12020{_}10.

Muja, Marius and David G Lowe (2009). "Fast Approximate Nearest Neighbors With Automatic Algorithm Configuration". In: *Proceedings of the Fourth International Conference on Computer Vision Theory and Applications*, pp. 331–340. ISSN: 00301299. DOI: 10.5220/0001787803310340. URL: http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0001787803310340.

Mulak, Punam and Nitin Talhar (2013). "Analysis of Distance Measures Using K-Nearest Neighbor Algorithm on KDD Dataset". In: *International Journal of Science and Research* 4.7, pp. 2319–7064. URL: www.ijsr.net.

Musa, Rabiu Muazu et al. (2019). *Machine Learning in Sports : Identifying Potential Archers*. Singapore: Springer. ISBN: 9789811325915. DOI: 10.1007/978-981-13-2592-2. URL: http://link.springer.com/10.1007/978-981-13-2592-2.

Norouzian, M. A. and S. Asadpour (2012). "Prediction of feed abrasive value by artificial neural networks and multiple linear regression". In: *Neural Computing*

and Applications 21.5, pp. 905–909. ISSN: 09410643. DOI: 10.1007/s00521-011-0579-5.

Papić, Vladan, Nenad Rogulj, and Vladimir Pleština (2009). "Identification of sport talents using a web-oriented expert system with a fuzzy module". In: *Expert Systems with Applications* 36.5, pp. 8830–8838. ISSN: 09574174. DOI: 10.1016/j.eswa.2008.11.031.

Paul, Ranjit Kumar (2006). "Multicollinearity: causes, effects and remedies". In: *IASRI, New Delhi*, p. 14. DOI: 10.1111/j.1755-148X.2008.00460.x.

Punch, W F et al. (1993). "Further research on feature selection and classification using genetic algorithms". In: *Proc. International Conference on Genetic Algorithms 93*, pp. 557–564.

Ramasubramanian, Karthik and Abhishek Singh (2019). "Machine Learning Theory and Practice". In: *Machine Learning Using R*. Berkeley, CA: Apress. Chap. 6, pp. 253–481. ISBN: 9781484223345. DOI: 10.1007/978-1-4842-2334-5.

Rendón, Eréndira et al. (2011). "Internal versus External cluster validation indexes". In: *International Journal* 5.1, pp. 27–34. URL: http://www.universitypress.org.uk/journals/cc/20-463.pdf.

Rohani, Abbas, Morteza Taki, and Masoumeh Abdollahpour (2018). "A novel soft computing model (Gaussian process regression with K-fold cross validation) for daily and monthly solar radiation forecasting (Part: I)". In: *Renewable Energy* 115, pp. 411–422. ISSN: 18790682. DOI: 10.1016/j.renene.2017.08.061. URL: https://doi.org/10.1016/j.renene.2017.08.061.

Saary, M. Joan (2008). "Radar plots: a useful way for presenting multivariate health care data". In: *Journal of Clinical Epidemiology* 60.4, pp. 311–317. ISSN: 08954356. DOI: 10.1016/j.jclinepi.2007.04.021.

Semenick, Doug (1990). "Tests and Measurements: The T-test". In: *National Strength and Conditioning Association Journal* 12.1, pp. 36–37.

Shalabi, Luai Al, Zyad Shaaban, and Basel Kasasbeh (2006). "Data Mining: A Preprocessing Engine". In: *Journal of Computer Science* 2.9, pp. 735–739. ISSN: 15493636. DOI: 10.3844/jcssp.2006.735.739. URL: http://www.thescipub.com/abstract/?doi=jcssp.2006.735.739.

Stevens, James P. (1984). "Outliers and influential data points in regression analysis". In: *Psychological Bulletin* 95.2, pp. 334–344. ISSN: 00332909. DOI: 10.1037/0033-2909.95.2.334.

Varma, Sudhir and Richard Simon (2006). "Bias in error estimation when using cross-validation for model selection". In: *BMC Bioinformatics* 7, pp. 1–8. ISSN: 14712105. DOI: 10.1186/1471-2105-7-91.

Vázquez-Araújo, Laura, Debbie Parker, and Eleanor Woods (2013). "Comparison of Temporal-Sensory Methods for Beer Flavor Evaluation". In: *Journal of Sensory Studies* 28.5, pp. 387–395. ISSN: 08878250. DOI: 10.1111/joss.12064.

Waterval, Mart (2017). "HANDBALVERBOND GROEIT DOOR HANDBALDAMES". In: *De Sportbestuurder.nl*. URL: https://desportbestuurder.nl/2017/07/04/handbalverbond-groeit-door-handbaldames/.

Wu, J. (2012). "Cluster Analysis and K-means Clustering: An Introduction". In: *Advances in K-means Clustering*. Berlin, Heidelberg: Springer. Chap. 1, pp. 1–16. ISBN: 978-3-642-29806-6. DOI: 10.1007/978-3-642-29807-3. URL: http://link.springer.com/10.1007/978-3-642-29807-3.

Zhang, Shichao et al. (2017). " Learning k for kNN Classification ". In: *ACM Transactions on Intelligent Systems and Technology* 8.3, pp. 1–19. ISSN: 21576904. DOI: 10.1145/2990508.