



University of
Stavanger

FACULTY OF SCIENCE AND TECHNOLOGY

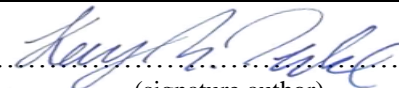
MASTER'S THESIS

Study program/specialization:
Information Technology –
Automation and Signal Processing

Spring semester, 2016

~~Open~~ / Confidential

Author: Kay Raymond Jessen Oskal


.....
(signature author)

Instructor: Prof. Trygve Eftestøl

Supervisor: Prof. Trygve Eftestøl

Title of Master's Thesis:

Myocardial damage during mountain bike race - an analysis of data from Nordsjørippet 2014
(NEEDED study)

Norwegian title:

Hjertemuskelsskade under terrengsykkelritt – en analyse av data fra Nordsjørippet 2014
(NEEDED studien)

ECTS: 30

Subject headings:

Nordsjørippet, NEEDED, Myocardial damage,
Troponin, Sports watch, Naïve Bayes,
Interpolation, Matthews correlation
coefficient, adjusdet Wald.

Pages: 74

+ attachments/other: 13 + embedded file.

Stavanger, 15th of June 2016

Date/year



University of
Stavanger

Myocardial damage during mountain bike race
- an analysis of data from Nordsjørippet 2014
(NEEDED study)

Kay Raymond Jenssen Oskal

June 2016

MASTER'S THESIS

Faculty of Science and Technology
Department of Electrical Engineering and Computer Science
University of Stavanger

Supervisor: Professor Trygve Eftestøl

Abstract

During Nordsjørippet 2014, blood samples, blood pressure and ECG were systematically gathered from over 1000 contestants. With this gathering of data, there is an opportunity to study the effects an endurance bike race has on presumably healthy individuals. In addition to this data, there are complete data from sports watches from a third of these contestants. When the medical analysis was done, 25 contestants were diagnosed with myocardial damage. Eight of these were from those with sports watches.

The motivation for this thesis is to test if the data logged with the sports watches can be used to predict myocardial damage. Or whether this data can be used to separate the contestants with high or low troponin I values.

To achieve this, a classification system was designed to test if there are patterns in the sports watch data that can be connected to myocardial damage or troponin I values.

As a consequence of variable sampling, the sports watch data were interpolated and smoothed. After this pre-processing, several features were extracted from the sports watches, from a predefined segment in the race. Two different methods, *exhausted search* and *sequential forward selection*, were used to select the feature subset maximizing the prediction rate. Furthermore, a *Naive Bayes* classifier was trained and validated using *leave-one-out* cross-validation. To measure reliability of the results in the experiments, *Matthews correlation coefficient* were used.

With an achieved prediction rate of 86% and a correlation coefficient of 0.58, the results clearly indicates that the data logged in the sports watches can predict myocardial damage.

When testing if the sports watch data can separate the contestants with high or low Troponin I, the results were inconclusive. Therefore, it is uncertain if this is achievable with the current data.

Preface

This thesis was written at the Department of Electrical Engineering and Computer Science, University of Stavanger. I would like to thank my supervisor Prof. Trygve Eftestøl for all advice and valuable feedback throughout this project. Additionally, I would like to thank Stein Ørn, Magnus Friestad Bjørkavoll-Bergseth and Øyvind Skadberg at Stavanger University Hospital for the data material, their medical insight and ideas. Finally, many thanks to all lecturers and my fellow students for five exiting years of study.

Contents

1	Introduction	1
1.1	Nordsjørittet	2
1.2	NEEDED	2
1.3	Previous work	3
1.4	Thesis outline	4
2	Theory	7
2.1	Interpolation	7
2.1.1	Piecewise Polynomial Interpolation	7
2.1.2	Hermite Cubic and Spline Interpolation	8
2.1.3	Piecewise Cubic Hermite Interpolation in MATLAB	8
2.2	Locally Weighted Scatter plot Smooth (<i>lowess</i>)	10
2.2.1	Robust <i>lowess</i>	12
2.3	Classification	13
2.3.1	Naive Bayes	14
2.4	Binomial proportion confidence interval	15
2.4.1	Adjusted Wald	15
2.5	Matthews correlation coefficient	16
3	Materials and Methods	17
3.1	Data material	17
3.2	Interpolation	19
3.2.1	Comparison of data	22
3.3	Smoothing of velocity	23
3.4	Split time detection	24
3.5	Race segments	26
3.6	Classifying data from the pulse watches	27
3.6.1	leave-one-out cross-validation	28
3.7	Feature extraction	29
3.7.1	Global features	29

3.7.2	Segment specific features	30
3.8	Class definitions	33
3.8.1	Original	34
3.8.2	CT	34
3.8.3	Troponin	35
3.8.4	Race Time	36
3.9	Feature selection	37
3.9.1	Exhaustive search	37
3.9.2	Sequential forward selection	38
4	Experiments and results	39
4.1	Prediction of myocardial damage	40
4.1.1	Original class definition	40
4.1.2	CT class definition	42
4.2	Prediction of raised hsTnI values	43
4.2.1	Troponin class definition	43
4.2.2	Race time class definition	45
4.3	Validating the results with a different hill	46
5	Discussion	49
5.1	Data material	49
5.2	Methods	50
5.3	Results	51
6	Conclusion	53
6.1	Future work	53
A	Matlab-code	59
B	Data sets	63
C	Additional figures	65
D	Additional tables	71

List of Figures

1.1	Course profile Nordsjørittet	2
1.2	A short summary of the data removed in previous work by Martin Nygård.	4
2.1	Comparison between Spline and Hermite interpolation.	8
2.2	Slopes for <i>pchip</i>	10
2.3	Local regression weight function.	12
2.4	<i>lowess</i> smoothing.	13
2.5	Confusion matrix	16
3.1	Data set before and after conversion between <i>cell array</i> and <i>struct</i>	18
3.2	Example of data found in <i>NeededData</i>	19
3.3	Interpolation of signals from the sports watches.	19
3.4	Comparison between <i>spline</i> and <i>pchip</i> when interpolating a heartbeat signal.	22
3.5	Comparison of original irregularly sampled signal versus interpolated signal.	22
3.6	Smoothing of the velocity signal.	23
3.7	Smoothing the velocity data.	23
3.8	Example of dataset after interpolation and smoothing.	24
3.9	Detect locations for the split times with use of the velocity signal.	24
3.10	Comparison of the official split distances and how far the contestants have reached at the official split times.	25
3.11	Detection of split times	26
3.12	Race segment from Tinghaug hill between the red dotted line.	27
3.13	Flow chart for classification with leave-one-out cross-validation	28
3.14	Feature extraction.	29
3.15	Example of heart rate and velocity signals from Tinghaug hill. The signal vectors are between the dotted lines.	31
3.16	Illustration of average incline with the variables used for calculating force.	33
3.17	Density functions for the original class definition with three classes.	35
3.18	Density functions for the original class definition with two classes.	35
3.19	Density functions for the CT class definition.	36
3.20	Density functions for the troponin class definition.	36

3.21	Density functions for the race time class definition.	37
4.1	Two exaples of data excluded from experiments.	39
4.2	ROC graph with the euclidean distance between classification result and perfect classification	44
4.3	Test results when validation classifier with a different hill.	47
5.1	Example of watch believed to have paused while contestant is standing still.	49
5.2	Example of mean velocity before and after smoothing	50
C.1	Visually inspection of correct POI from a selection of the contestants	65
C.2	Comparison of original heart rate signal versus interpolated signal from a selection of contestants.	66
C.3	Confusion matrices with the best results from the original class definition. .	67
C.4	Confusion matrices with the best results from CT class definition. Both with misclassification cost equal 5.	67
C.5	Confusion matrices with the best results from the hsTnI class definition. . .	68
C.6	Confusion matrices with the best results derived from the race time class definition. Both with misclassification cost equal 2.	69

List of Tables

2.1	Rule of thumb for interpreting the size of the correlation coefficient, Hinkle et al. [19]	16
3.1	Example of data found in <i>SplitTimes</i> .	19
4.1	Classification results from original classes - exhaustive search feature selection	41
4.2	Classification results from original classes - sequential forward feature selection	42
4.3	Classification results from CT classes - Exhaustive search feature selection	42
4.4	Classification results from CT classes - sequential forward feature selection	43
4.5	Classification results from troponin classes - Exhaustive search feature selection	44
4.6	Classification results from hsTnI classes with threshold at 100 [ng/l]	45
4.7	Classification results from hsTnI classes with threshold at 150 [ng/l]	45
4.8	Classification results from TnI classes with threshold at 200 [ng/l]	45
4.9	Classification results from race time classes - exhaustive search feature selection.	46
4.10	Classification results from race time classes - sequential forward feature selection.	46
D.1	Comparison of different classifiers. Sequential forward features selection and misclassification cost = 15	71
D.2	Comparison of different classifiers. Sequential forward features selection and misclassification cost = 5	71
D.3	Comparison of different classifiers. Exhaustive search features selection and misclassification cost = 4	72
D.4	Comparison of different classifiers. Sequential forward features selection and misclassification cost = 2	72

List of Abbreviations

ACC	Accuracy
BMI	Body Mass Index
CT	Computed Tomography
FN	False Negative
FP	False Positive
hsTnI	High Sensitive Troponin I
lowess	Locally Weighted Scatter plot Smooth
MAD	Median Absolute Deviation
MCC	Matthews correlation coefficient
NEEDED	<u>N</u> orth <u>S</u> ea <u>R</u> ace <u>E</u> ndurance <u>S</u> tudy
pchip	Piecewise Cubic Hermite Interpolating Polynomial
PDF	Probability Density Function
POI	Point Of Interest
TNR	True Negative Rate
TN	True Negative
TPR	True Positive Rate
TP	True Positive

1. Introduction

Recreational active people will in average live seven years longer than those who are inactive. With reduced stamina the inactive have a much greater risk of developing cardiovascular disease and death. But with with some positive change in oxygen intake during exercise this risk is reduced. From the Harmonised European Time Survey (HETUS) it is shown that Norway is second to last when it comes to overall daily physical activity related to transport, trips, training and sports [4]. In contrast, there has been an increase in physical activity during leisure time. An important motivational factor for organized training are the many competitions such as Nordsjørittet, Birkebeiner ski/bike race and Birkebeiner run [2].

Intense physical training does not come without some risk, although the probability for sudden death during these competitions are very small. In 100 000 marathon hours there is an estimate of one death. During Vasaloppet (90 km ski race), there has been thirteen deaths among 698.102 competitors from 1970 to 2005 [15]. In comparison to the health gain, this risk seems modest.

With data systematically gathered by the North Sea Race Endurance Study (NEEDED) in Nordsjørittet 2014, there is an opportunity to study the effects an endurance bike race has on presumably healthy individuals. In addition to blood samples, blood pressure and electrocardiography (ECG) from over 1000 contestants, there are complete sports watch data from about one third of these. This data and one of the biomarkers (troponin I¹) derived from the blood samples, are studied in detail in this thesis. The main goal in the end is to test if there are any relations between the data from the sports watches and high troponin I values or whether the contestant may have a serious heart condition.

¹Troponin I is a protein specific to the cardiac muscle. Cardiac troponins are released into circulation in response to myocardial necrosis. As such, cardiac troponins are one of the preferred biomarkers for the detection of cardiac injury [8]

1.1 Nordsjørippet

Nordsjørippet is a 91 km long mountain bike race between Egersund and Sandnes. When the race was arranged for the first time in 1998, only 184 contestants participated. But during the first five years the number of contestants increased fivefold. The increase continued and in the years 2012 - 2014 this number was 12.500 which is the upper limit decided by the organizers. With these numbers Nordsjørippet is the second largest mountain bike race in Norway [30]. The varied course takes you through forest, mud, dirt roads and some asphalt, along the famous North Sea Route and the Kings Road [7].

From the course profile, figure 1.1, one can see that the beginning of the course is somewhat hilly before the first real hill climb up to Vandavatnet at 33 km. Thereafter awaits a decline to about sea level, where it continues to be flat for a while. After around 70 km awaits the steepest climb of the race, Tinghaug hill, which is the last real hurdle before the contestants can ride to the finish line in relatively easy terrain. The hills up to Vandavatnet and Tinghaug will be the primary focus, to test if there are any connections between the logged data from the sports watches and myocardial damage or high troponin I values.

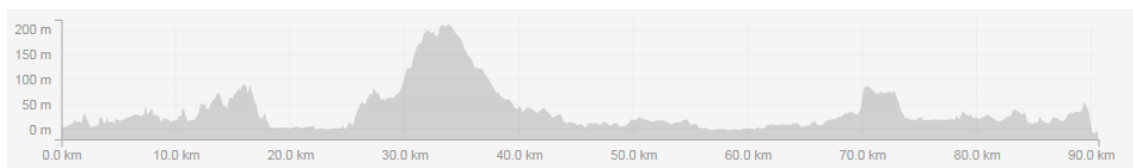


Figure 1.1: Course profile Nordsjørippet [3].

1.2 NEEDED

NEEDED is the biggest study of biomarkers on people competing in exercise competition in history. A group of scientists from *Helse Stavanger HF* are the initiators for this study, and one of the goals is to increase the knowledge about what happens inside the body on presumably healthy individuals during these types of competitions. The study will place particular emphasis on changes in troponin I values measured as high sensitive troponin I (hsTnI¹). As a marker hsTnI is very specific for myocardial injury, and is obligatory in the diagnosis of acute coronary syndromes. Regardless of reason, release of hsTnI into the blood stream is connected to unfavourable prognosis in multiple cardiac disorders. In contradiction to this, multiple studies have shown that there can be release of hsTnI

¹New method that is regarded as high sensitive if it, with high precision, can prove measurable values of troponin in 75-80% of healthy individuals [10].

during endurance training and competitions [27] [9]. The clinical significance of this for presumably healthy trained individuals is uncertain.

During the pilot study in 2013, which had 97 completing contestants, they found after medical examination that three of the four with highest hsTnI values had undiagnosed heart condition. The following year during the main study, a over 1000 contestants participated. Blood samples, blood pressure and ECG were systematically gathered 24 hours before, 3 and 24 hours after the race. As with the pilot study the contestants over a hsTnI threshold ¹ were called in for medical examination. These were examined either with computed tomography (CT)-coronary angiography ² or regular coronary angiography. When the medical examination was done, 25 contestants were diagnosed with myocardial damage. Eight of these were from those with sports watches.

All personal data is confidential and all research data are identified with their personal identification number to make allowance for gathering additional data in the future. The data are saved on a closed research server at Stavanger university hospital and has the identification key stored separate from this. In this thesis all data are depersonalised.

1.3 Previous work

This section is a short summary of Martin Nygård's previous work on this project. His first task was to gather the data files from the sports watches used during the race and to save these in a database for later work. The data were from both new and old watches from Polar and Garmin. Because of this, there are several different types of data files, and the information logged in these watches varies. To complement this data, additional information were provided by the NEEDED project group. This is information about the contestants age, total race time, the three hsTnI measurements and some medical remarks from those who had findings during CT-scan. A class indicator was given, dividing the contestants in to three classes. Two groups are over a hsTnI threshold value decided by the NEEDED project group, where the potential findings from CT-scan is their difference. The third class has low hsTnI values. In total there were registered 292 contestants with sports watches. After deleting corrupted, badly sampled or incomplete there are 182 contestants remaining. How and why these files were removed, can be read in Martin Nygård's master's thesis [25]. A flowchart of the removed data can be seen in figure 1.2. This is a reprint from the above mentioned thesis, why there is missing one contestant here is not known.

¹The hsTnI used in the main study was 195 [ng/l]

²CT coronary angiography is a non-invasive examination done with contrast agent and low radiation < 1 mSv.

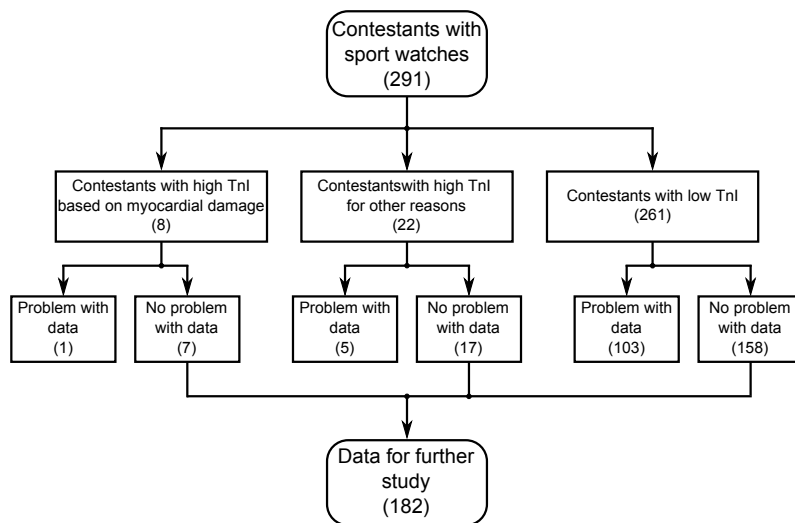


Figure 1.2: A short summary of the data removed in previous work by Martin Nygård. Reprint of the flowchart used in his thesis [25].

Furthermore, Martin tried with statistical analysis to test if there are any connections between those with myocardial damage and their heart rate data. The results can be studied in his thesis, but in short he concluded that there seems to be some connections worth mentioning. Those with heart problems had lower heart rate than the other contestants. It is not the case throughout the race, since the sick contestants heart rate declines slower than those who are healthy. This is more noticeable in the parts of the race that are less intense.

1.4 Thesis outline

This thesis is organized as follows.

Chapter 2 - Theory:

This chapter will in short describe the theory behind some of the methods used in Materials and Methods.

Chapter 3 - Materials and Methods:

This chapter starts with a presentation of data material and the pre-processing of this data. After this follows a description of the various methods used to interpolate, smooth, extract features and how to select a subset from these features. Additionally, four class definitions are presented.

Chapter 4 - Experiments and results:

This chapter describes the different experiments followed by the results from these.

Chapter 5 - Discussion:

This chapter discusses the data material, methods and results.

Chapter 6 - Conclusion:

The conclusion of the thesis and possible future work is found in this chapter.

Appendix A - Matlab-code:

A short presentation of the various MATLAB-files used during this work.

Appendix B - Data sets:

A short presentation of the various data sets that are generated and used during the experiments.

Appendix C - Additional figures:

Figures for visual inspection of data. And some figures to complement the results.

Appendix D - Additional tables:

Tables with comparison of the different classifiers initially tested.

2. Theory

This chapter presents the theoretical background needed for some of the methods used later on in this thesis. In addition, the theory behind how to measure the reliability and quality of the results from the experiments.

2.1 Interpolation

There are several reasons why it is desirable to interpolate the data. Since the signals from the sports watches in this thesis are sampled with different rates (some irregularly), interpolating it makes it possible to “read between” these samples. An important property when choosing an interpolating function is that it has to preserve the shape of the original signal. The simplest way is with the use of polynomials, but extremely large polynomials are generally needed to preserve the shape of the original signal. These are computationally expensive to decide and evaluate. They can also have issues with oscillation and non-convergence.

2.1.1 Piecewise Polynomial Interpolation

Piecewise polynomial interpolation is an alternative to the theoretical and practical issues with polynomial interpolating of higher degree. This method makes it possible to interpolate larger quantities of data with lower degree polynomials, which is an advantage. Interpolation of a given signal with the samples (t_i, y_i) , a different polynomial would be used for each subinterval $[t_i, t_{i+1}]$. The point t_i , where the interpolant changes from one polynomial to another has different names, which are *knots*, *breakpoints* or *control points*. Piecewise linear interpolation is the simplest use of this method. Here the successive samples at the *knots* are connected with straight lines. Even though this method removes the issues with oscillation and non-convergence, it achieves this by sacrificing the smoothness of the interpolating function.

2.1.2 Hermite Cubic and Spline Interpolation

With *Hermite* interpolation the derivative will in addition to the interpolation function, be specified for each interval. A *Hermite cubic* interpolant is a piecewise polynomial interpolant that has a continuous first derivative. An alternative is cubic *spline*, which is piecewise cubic polynomial which has a continuous second derivative and will with this be more suited if smoothness is critical. But if the preservation of the monotonicity of the original signal is essential, an *Hermite cubic* interpolant would be more appropriate [18]. Example in figure 2.1.

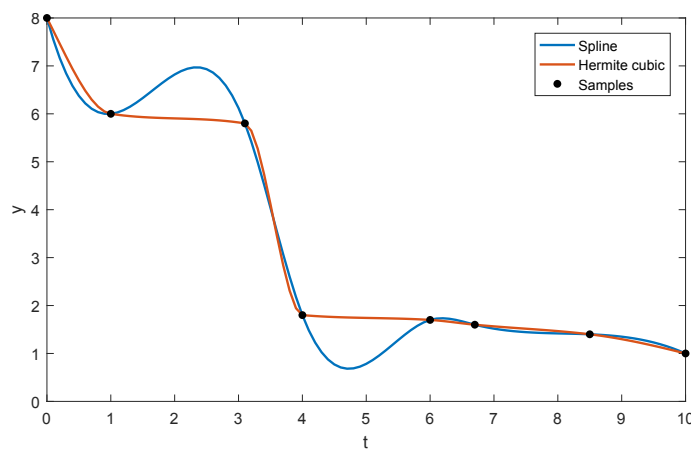


Figure 2.1: Comparison between Spline and Hermite interpolation.

2.1.3 Piecewise Cubic Hermite Interpolation in Matlab

The following section is how piecewise cubic Hermite interpolation is implemented in MATLAB, and is derived from *Numerical Computing with Matlab* [23].

Let h_i specify the length of the i th subinterval:

$$h_i = t_{i+1} - t_i \quad (2.1)$$

The first divided difference, δ_i , is derived from

$$\delta_i = \frac{y_{i+1} - y_i}{h_i} \quad (2.2)$$

Let d_i be the slope of the interpolant at t_i :

$$d_i = P'(t_i) \tag{2.3}$$

For the piecewise linear interpolant, $d_i = \delta_{i-1}$ or δ_i , but for higher order interpolants this is not necessary true. Regard the following function in the interval $t_i < t < t_{i+1}$, given in terms of the local variables $s = t - t_i$ and $h = h_i$:

$$P(t) = \frac{3hs^2 - 2s^3}{h^3}y_{i+1} + \frac{h^3 - 3hs^2 + 2s^3}{h^3}y_i + \frac{s^2(s-h)}{h^2}d_{i+1} + \frac{s(s-h)^2}{h^2}d_i \tag{2.4}$$

This cubic polynomial in s , and thus in t , satisfies the two conditions on function values and the two possible unknown derivative values:

$$P(t_i) = y_i, P(t_{i+1}) = y_{i+1} \tag{2.5}$$

$$P'(t_i) = d_i, P'(t_{i+1}) = d_{i+1} \tag{2.6}$$

If the function values *and* the derivative to a set of samples are known, piecewise Hermite interpolation will be able to reproduce this signal. But if the derivative is unknown the slopes d_i has to be defined. One of the methods to define these will be described here and is called *pchip*.

In MATLAB, this is a shape-reserving, visually pleasing interpolant. It is based on an old Fortran program by Fritsch and Carlson [14] and is described by Kahaner, Moler and Nash [20]. The idea is to decide the slopes d_i so the function values wont overshoot the data, at least not locally.

- If the signs of δ_i and δ_{i-1} are opposite or if one of them are zero, t_i would either be a local maximum or minimum. Then

$$d_i = 0$$

This is illustrated in the left part of figure 2.2. The blue solid line is the piecewise linear interpolant. Its slope on either side of the *knot* have opposite signs. Ergo, the slope of the dashed line is zero. The green solid line is the shape-preserving interpolant formed by two different cubics. These two interpolate the center value and both their derivative are zero here.

- If the signs of δ_i and δ_{i-1} are the same and their intervals are equal in length, then d_i is the harmonic mean of the two slopes.

$$\frac{1}{d_i} = \frac{1}{2} \left(\frac{1}{\delta_{i-1}} + \frac{1}{\delta_i} \right) \quad (2.7)$$

I.e. at the *knots*, the reciprocal slope of the Hermite interpolant is the average of the slopes from the piecewise linear interpolant. An illustration of this is shown in the right part of figure 2.2. If the slopes of the linear interpolant (blue line) changes from 1 to 5 at the *knot*, the slope of the dashed line is 3, the average of 1 and 5. The shape-preserving interpolant (green line) is then formed by two cubics that interpolate the center value which have a slope of $1/3$.

- If the signs of δ_i and δ_{i-1} are the same, but the intervals are of unequal length, then d_i is a weighted harmonic mean. These weights are decided by the intervals lengths.

$$\frac{\omega_1 + \omega_2}{d_i} = \frac{\omega_1}{\delta_{i-1}} + \frac{\omega_2}{\delta_i} \quad (2.8)$$

where

$$\omega_1 = 2h_i + h_{i-1}, \quad \omega_2 = h_i + 2h_{i-1} \quad (2.9)$$

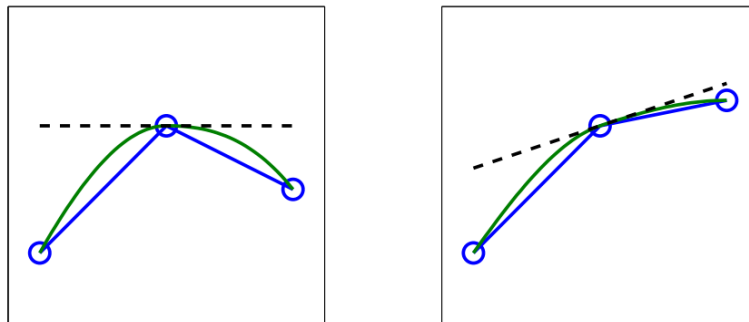


Figure 2.2: Slopes for *pchip* [23].

2.2 Locally Weighted Scatter plot Smooth (*lowess*)

The following sections on smoothing with *lowess* and *rlowess* is derived from *Filtering and Smoothing Data* [1]. This smoothing method uses weighted linear regression. The process is regarded as local since, as moving average, each smoothed value is determined by neighbouring data points within a give span. It is weighted since a regression weight is

defined for all data points within the span. In addition to this weight, a more robust weight is available (*rlowess*), which makes the smoothing process more resistant to outliers.

For each sample will *lowess* follow these steps:

1. The regression weights are computed for each sample within the span. These are derived from the tricube function below.

$$W_i = \left(1 - \left|\frac{x - x_i}{d(x)}\right|^3\right)^3 \quad (2.10)$$

Where x is the predictor value identified with the response value to be smoothed, x_i are neighbouring samples of x defined by the span. $d(x)$ is the distance from x to outermost predictor in the span along the abscissa¹. The weights have the following characteristics:

- The sample to be smoothed has the largest weight. Hence, it has most influence.
 - Outside the span the samples has zero weight and therefore no influence.
2. A weighted linear least-square regression is preformed with a first degree polynomial.
 3. The smoothed value is given by the weighted regression at the predictor value of interest.

If the calculation involves an equal number of neighbouring samples on either side of the smoothed sample, there is a symmetric weight function. But, if there are unequal number of neighbouring samples on either side, the weight function will be asymmetric. Unlike smoothing with moving average, the span wont change its size. If one were to smooth the start or end sample in a signal, the shape of the weight function is truncated by half. An example of this is shown in figure 2.3.

A *lowess* example with the smoothed values and corresponding regressions for the first four samples in a signal are shown in figure 2.4.

Observe that the span does not change during this example. But, depending on the number of neighbours on either side, the regression weight function are not always symmetric around the sample to be smoothed. In the plots (a) and (b) there is a asymmetric weight function, whereas in (c) and (d) the weight function is symmetric.

¹The distance of a point from the y-axis on a graph in the Cartesian coordinate system. It is measured parallel to the x-axis. For example, a point having coordinates (2,3) has 2 as its abscissa. <http://www.dictionary.com/browse/abscissa>

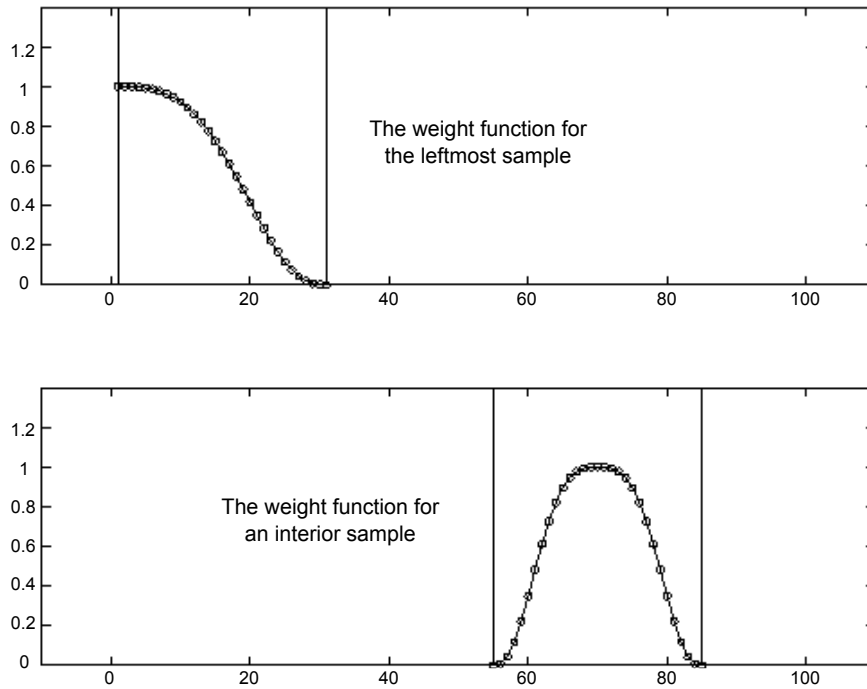


Figure 2.3: Local regression weight function. Figure from Mathworks [1]

2.2.1 Robust *lowess*

If the signal to be smoothed contains outliers, the values might be misrepresented, and not reflect the majority of neighbouring samples. To avoid this problem, a more robust method of *lowess* is available that won't be influenced by outliers.

The robust smoothing procedure follows these steps:

1. Calculate the residuals from the smoothing procedure described in the previous section.
2. Compare the *robust weights* for each sample in the span. The weights are given by the bisquare function

$$W_i = \begin{cases} (1 - (r_i/6MAD)^2)^2, & |r_i| < 6MAD, \\ 0, & |r_i| \geq 6MAD \end{cases} \quad (2.11)$$

where r_i is the residual of the i th sample produced by the smoothing procedure, and MAD is the median absolute deviation of the residuals,

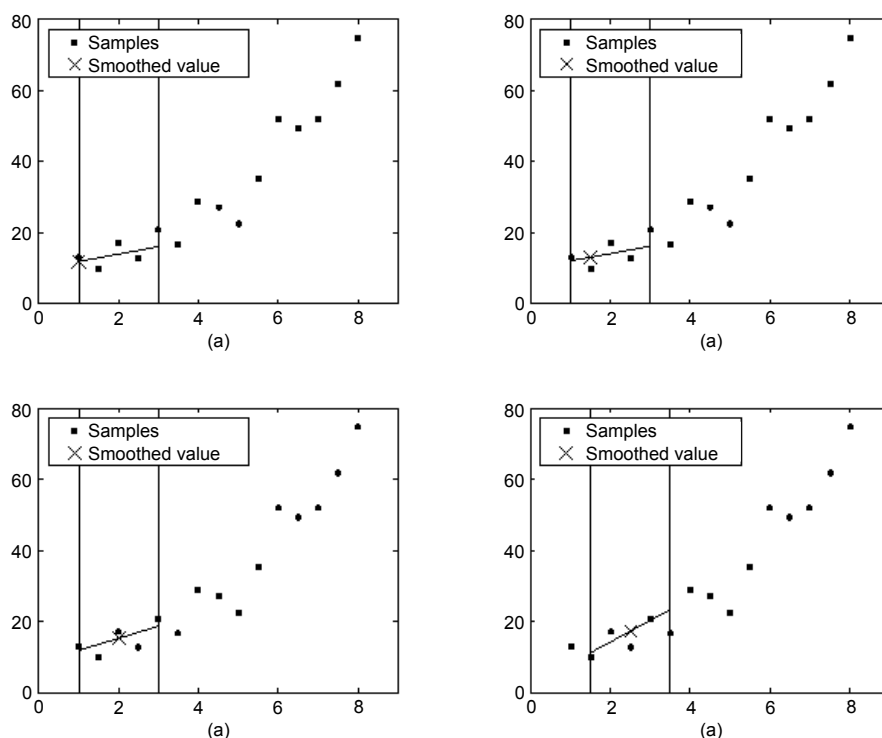


Figure 2.4: *lowess* smoothing. Figure from Mathworks [1].

$$MAD = \text{median}(|r|) \quad (2.12)$$

The median absolute deviation is a measure of how spread out the residuals are. If r_i is small compared to $6MAD$, then the robust weight is close to 1. If r_i is greater than $6MAD$, the robust weight is 0 and the associated sample is excluded from the smooth calculation.

3. Smooth the signal again using the robust weights. The final smoothed value is calculated using both the local regression weight and the robust weight.
4. Repeat the previous two steps for a total of five iterations.

2.3 Classification

A classifier can refer to a mathematical function, which implements a classification algorithm that maps an observation to a category. This is supported by a set of training data containing observations which is labelled with the correct category, and is known as supervised learning. These observations can be described by a vector of features, which

may be a physical description of an object (size, form, colour) or a description of the signal properties (mean, variance, etc.). These feature vectors can be regarded as points in a multi-dimensional space.

2.3.1 Naive Bayes

This classification algorithm has gone under several different names. Some of these are *idiot's Bayes*, *simple Bayes* and *independent Bayes*. It is especially appropriate if the dimension of the feature space is increasing, making density estimation demanding [17]. If one need N training data points to obtain sufficiently accurate estimation for a probability density function (PDF) in a one dimensional space. The number of points needed for a higher dimensional space will increase exponentially, referred to as the *curse of dimensionality*. So for a l -dimensional space one would need N^l data points. But if one assume that the individual features are statistically independent the PDF can be written as:

$$p(X|\omega_k) = \prod_{j=1}^P p(X_j|\omega_k) \quad (2.13)$$

Now one will only need to estimate l one-dimensional PDFs and lN data points is sufficient to get good estimates instead of N^l [29].

This assumption of independence is for most problems unrealistic and generally not true, but practical examples show that it often competes well and can even be more effective than more sophisticated classifiers [16].

The *Naive Bayes* may often be a better classifier than more powerful alternatives when the sample size is small, even in domains where this learning model might not be the most appropriate [13]. Since it also has favourable properties like simplicity, learning speed, classification speed, etc., it will be used in the experiments in chapter 4.

The *Naive Bayes* algorithm, as presented in the MATLAB documentation [5].

1. Estimate the densities of the predictors within each class.
2. Models posterior probabilities according to Bayes rule. That is, for all $k = 1, \dots, K$,

$$\hat{P}(\omega_k|X_1, \dots, X_p) = \frac{P(\omega_k) \prod_{j=1}^P p(X_j|\omega_k)}{\sum_{k=1}^K P(\omega_k) \prod_{j=1}^P p(X_j|\omega_k)} \quad (2.14)$$

where:

- X_1, \dots, X_P are the random predictors of an observation.
 - $P(\omega_k)$ is the prior probability that a class is k .
3. Classifies the observation by estimating the posterior probability for each class, and then assigns the observation to the class yielding the maximum posterior probability.

2.4 Binomial proportion confidence interval

When working with sampled data from a larger population it is interesting to know how reliable the observations are without further sampling. A confidence interval can be seen as a prediction of what would happen if the same experiments were to be repeated, and since each observation are seen as a sample of the population, one can predict the range of the true values in the population.

There are several formulas for calculating a binomial confidence interval, and in the experiments that follow the *adjusted Wald* approach is found to be suitable. One reason for this is that it behaves well for very small sample sizes. Another reason is that, in comparison to other methods, it provides shorter intervals with actual coverage probability usually nearer the nominal confidence level. This means that when forming a 95% confidence interval other methods achieves at least 95% and typically coverage probabilities up to 98% or 99% where *Adjusted Wald* may be less than 95% but it is usually quite close [11].

2.4.1 Adjusted Wald

Another name for this binomial confidence interval is Agresti-Coull Interval [11].

For X successes in n trials, one can define

$$\tilde{n} = n + z^2 \tag{2.15}$$

and

$$\tilde{p} = \frac{1}{\tilde{n}} \left(X + \frac{1}{2}z^2 \right) \tag{2.16}$$

A confidence interval for p can then be given as

$$\tilde{p} \pm z \sqrt{\frac{1}{\tilde{n}} \tilde{p}(1 - \tilde{p})} \tag{2.17}$$

Where z is the $1 - \frac{1}{2}\alpha$ quantile of a standard normal distribution. For a 95% confidence interval, $\alpha = 0.05$ which gives $z = 1.96$.

2.5 Matthews correlation coefficient

As a measurement of the quality of binary classifications, Matthews correlation coefficient (MCC) was introduced by biochemist Brian W. Matthews in 1975 [22]. It is generally regarded as a balanced measure between true and false positives and negatives, but can be used even if the classes are unbalanced. In essence the MCC is the correlation between the observed and predicted binary classification and will return a value between -1 and +1. A perfect prediction will give a coefficient of +1, 0 is not better than random guess, while -1 shows total disunity between prediction and observation. For the values between these extremities, use table 2.1 as a rule of thumb.

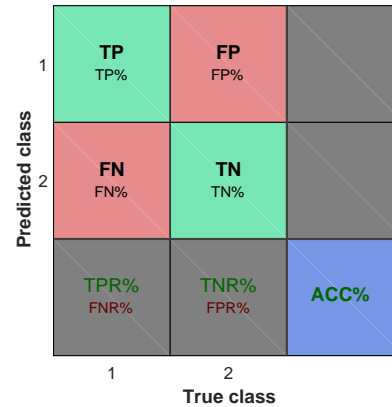


Figure 2.5: Confusion matrix

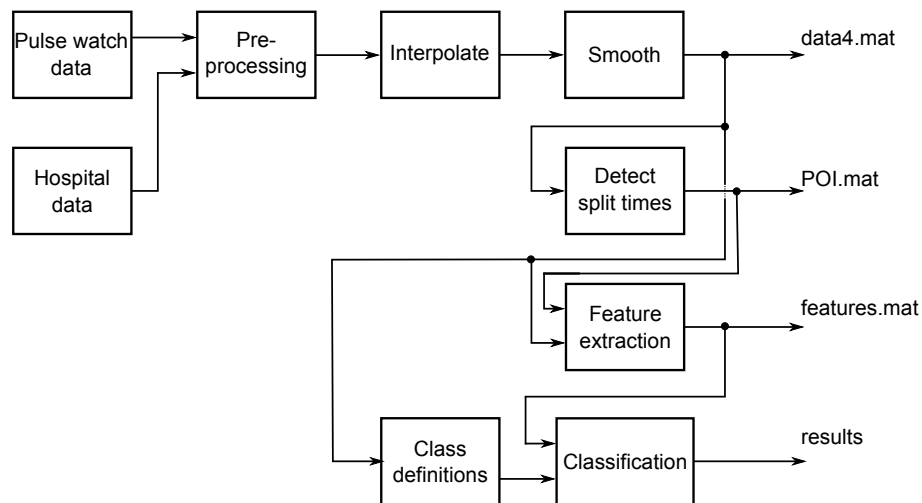
<i>Size of Correlation</i>	<i>Interpretation</i>
0.90 to 1.00 (-0.90 to -1.00)	Very high positive (negative) correlation
0.70 to 0.90 (-0.70 to -0.90)	High positive (negative) correlation
0.50 to 0.70 (-0.50 to -0.70)	Moderate high positive (negative) correlation
0.30 to 0.50 (-0.30 to -0.50)	Low positive (negative) correlation
0.00 to 0.30 (-0.00 to -0.30)	Little if any correlation

Table 2.1: Rule of thumb for interpreting the size of the correlation coefficient, Hinkle et al. [19]

There is no optimal way of describing a confusion matrix of true and false positives and negatives with a single digit. But MCC is generally seen as one of the best for said measures [26] and can be calculated directly from a confusion matrix, figure 2.5, as follows:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (2.18)$$

3. Materials and methods



This chapter starts with a presentation of the data material and how it is extended and restructured, to simplify the use of the data. Then, the pre-processing of the signals and method for interpolation are described, followed by an example on how and why the velocity signals are smoothed.

Next, there is a description of the method used to detect split times in the race. From these split times, race segments are defined so they are equal for all contestants. Furthermore, there is a presentation of the features extracted from these segments. This is followed by two methods to select subsets from these features. Additionally, four class definitions will be described along with their associated goals.

The MATLAB scripts/functions for the above mentioned methods is found in appendix A.

3.1 Data material

As mentioned in the introduction previous earlier work, the data used in this thesis are based on the work of Martin Nygård. To make the data more user friendly, the whole

dataset is changed from a *cell array*¹ to *struct*², example seen in figure 3.1. The main advantage of this restructuring, is that it will be easier to work with, it will also make it easier to read for future work. It is now more intuitive to locate the desired data just by reading the headlines in the *struct*. As an example, earlier one had to write `filer1{i,2}(:,4)` to access the heart rate signal from contestant number i . With *struct*, one can now write `Data(i).WatchData(:,4)` to access the same signal.

(a) Original data set as *cell array*

(b) Data set after conversion to *struct*

Figure 3.1: Data set before and after conversion between *cell array* and *struct*.

Since the *WatchData* contains several data vectors, ideally this *cell* should also be changed to *struct*, but it makes the file size to big. Causing the loading of data to take several minutes, which is unwanted since the data will be loaded often. The columns in *WatchData* contains the following signal vectors:

1. Time in seconds, $t(n)$.
2. Velocity in m/s or km/t, $v(n)$.
3. Distance in meter, $d(n)$.
4. Heart rate in bpm³, $hr(n)$.
5. Longitude (if any).
6. Latitude (if any).
7. Elevation (if any), $elev(n)$.

With additional information provided by the hospital the *NeededData*-field is now extended with height, weight and body mass index (BMI) for all contestants. This field is complemented with headlines so this information is found more easily, see figure 3.2.

¹Arrays that can contain data of varying types and data.

²Arrays with named fields that can contain data of varying types and data.

³beats per minute

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	'ID'	'RaceTime'	'S_HSTN1'	'S_HSTN2'	'S_HSTN3'	'Pulsfile'	'Sex'	'RemarksCT'	'Measures'	'Age'	'Weight'	'BMI'	'Height'
2	50226	3.1675	11.0900	277.4300	85.4800	'1'	'M'	'Misstanke ...	'Angio 7/8 ...	37.7000	102.3000	28.0400	191

Figure 3.2: Example of data found in *NeededData*.

This new information from the hospital also contains the official split time for the contestants throughout the race. In total there are five official split times, which are at the following locations:

1. 21 km: Brusand camping.
2. 32 km: Vandavatnet.
3. 55 km: Hå gamle prestegård (Haa old vicarage).
4. 66 km: Bottom of Tinghaug hill.
5. 67 km: Top of Tinghaug hill.

This extra information is appended to the data as a new *struct* element, *SplitTimes*. Since the time format for these are in hours, they are converted to seconds. Thus making it easier to use these with the rest of the data set, which are sampled in seconds. To preserve all information, the split times are saved both in hours and in seconds, example in table 3.1. The data is saved in a *.mat* file, *Data.mat*.

21 km	32 km	55 km	66 km	67 km	Unit
01:00:13	01:43:54	02:30:48	03:03:22	03:08:57	hour
3613	6234	9048	11002	11337	sec

Table 3.1: Example of data found in *SplitTimes*.

3.2 Interpolation

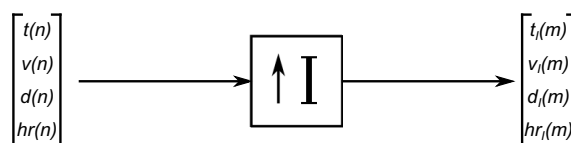


Figure 3.3: Interpolation of signals from the sports watches.

The fact that the recorded data are from several types of sports watches, means that there are some differences that needs to be addressed. There are a few issues, but the

main problem is that many of the signals are sampled irregularly. Where some of the regularly sampled signals have logged one sample per second, these might only have three to four samples for every ten seconds. To make it worse, the time between these samples are often irregular. It is imperative to have equal sampling rate when comparing the signals. Some of the data sets have signals which are already sampled once per second. Due to this, the remainder of them are also interpolated up to this rate. But, before the signals are interpolated, some additional pre-processing are needed.

The data set *Data.mat* generated in the previous section is the starting point for this pre-processing. The first issue to be addressed, is to locate the correct start and end time/sample for all contestants. This is because some of the contestants start their watches before the start signal of the race, some forget to stop it after they have crossed the finish line and some even do both. The reason why this needs to be changed is when interpolating, issues with start/end time can provoke some unwanted spikes. A method is created to handle this, see algorithm 1.

For all contestants, the method starts with the first sample and iterates forward until a criterion is met, this is determined as the new start sample for the data. Next, to determine a new end sample, the method iterates backwards from the last sample until it locates a sample that fulfils the end criterion. The criteria to decide where to set the new start and end samples are found through experimentation. It should approximate the real start and end samples better than it has before.

Algorithm 1: Pre-processing part 1: Adjustment of start/end sample.

```

for all contestants do
  Initialize  $t_s(n) = t(start)$ 
  while  $t_s(n + 8) - t_s(n) > 50$  do
    |  $n = n + 1$ 
  end
  start =  $t_s(n)$ 
  Initialize  $t_e(n) = t_e(end)$ 
  while  $t_e(n) - t_e(n - 10) > 60$  or  $t_e(n) == t_e(n - 1)$  do
    |  $n = n - 1$ 
  end
  end =  $t_e(n)$ 
  New Data = Data(start:end)
end

```

Furthermore, in some of the data sets, the heart rate signals has some or more samples which are zero. Since interpolation of the signal is the end goal of this process, these samples are deleted so they wont cause any unwanted artefacts. Another issue is that some samples, mainly at the end, has the same sample time two or more samples in a row. This causes the interpolation function in MATLAB to break down, which is obviously

unwanted. Hence, the samples that has the same time as the previous sample are deleted. Lastly, some samples in the distance signal are recorded with shorter distance than the previous, which is incorrect. These are estimated as the average of the distance sample before and the one after.

$$d(n) = \frac{1}{2} [d(n-1) + d(n+1)] \quad (3.1)$$

A summary of these pre-processing steps can be seen in algorithm 2.

Algorithm 2: Pre-processing part 2: Adjustment of corrupted data.

```

for all contestants do
  | for all  $n$  do
  | | if  $hr(n) == 0$  then
  | | | Delete sample  $n$  in all columns
  | | end
  | | if  $t(n) == t(n-1)$  then
  | | | Delete sample  $n$  in all columns
  | | end
  | | if  $d(n) < d(n-1)$  then
  | | | Estimate  $d(n) = \frac{1}{2} [d(n-1) + d(n+1)]$ 
  | | end
  | end
end

```

When the pre-processing of the data set is finished, the signals are ready for interpolation. This is done with a MATLAB function, *interp1* which has several different methods of interpolation available. To preserve the shape of the original signals, the methods *pchip* and *spline* described in section 2.1, are tested on all data sets. An issue with the signals from the sports watches is that it may at times have holes. This may happen if the heart rate monitor misses some or several beats after each other. If *spline* interpolation is used to fit these signals, it will cause an unwanted overshoot in some of the data sets. *pchip* on the other hand, will not have this issue. It will not be as smooth as *spline*, but visually it will represent the original signal better. An example of this, taken from the interpolation of the heart rate signal from two of the contestants are shown in figure 3.4.

When the interpolation is done, some final adjustments of the data sets are required. Some of the contestants does not have any logged velocity data. These are estimated as the derivative of the distance.

$$v_{I,est}(m) = d_I(m+1) - d_I(m) \quad \text{for all } m \quad (3.2)$$

Another issue with the velocity is that there is no consistency as to which unit is used. This is corrected by first calculating the mean velocity for everyone throughout the race.

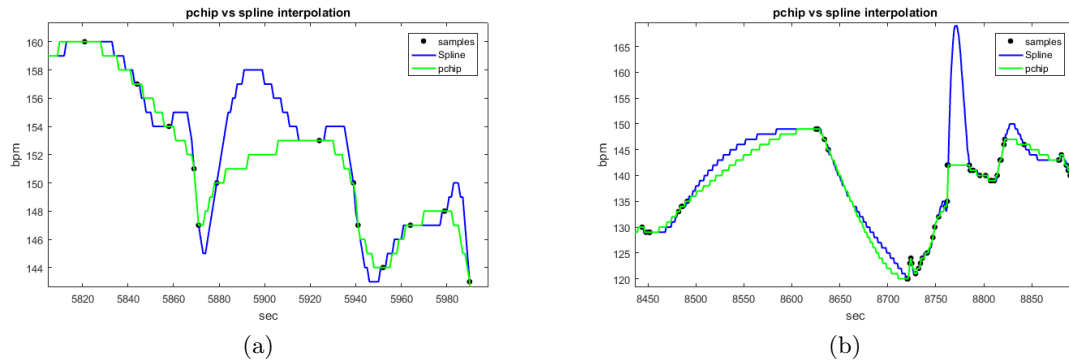


Figure 3.4: Comparison between *spline* and *pchip* when interpolating a heart-beat signal.

If this is lower than 10, the unit is in meters per seconds and is corrected by multiplying it with 3.6. The logic behind this threshold, is that the fastest contestant had less than 10 m/s as official average velocity and the slowest had over 10 km/t. The data set is saved as *Data2.mat*.

3.2.1 Comparison of data

To compare if the interpolated signals are preserved and equal in shape as the original, all contestants are plotted with both old and new data for visually inspection. An example with both original data, which in this case is irregularly sampled, and interpolated data is shown in figure 3.5.

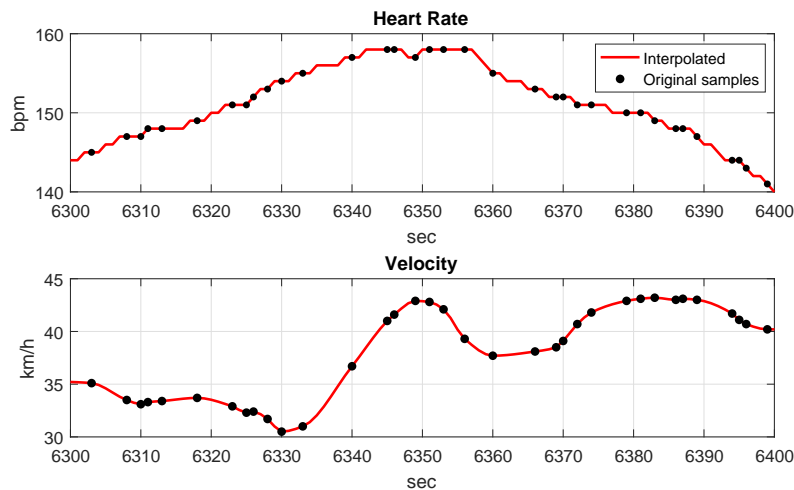


Figure 3.5: Comparison of original irregularly sampled signal versus interpolated signal.

3.3 Smoothing of velocity

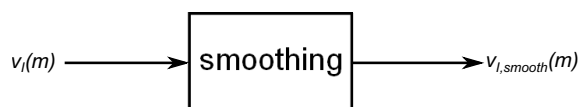


Figure 3.6: Smoothing of the velocity signal.

Since velocity will play an important role in the next parts of the project, such as locating good estimation of the split times and feature extraction, additional pre-processing is required. The velocity signals for many of the contestants are very chaotic which makes it hard to observe any comparable trends in these signals. In addition, the sports watches that did not log the velocity and was estimated as the derivative of distance (eq. 3.2), has some issues. The distance travelled between each logged sample are in some cases extremely uneven. This results in a very spiky velocity, which is even harder to read, figure 3.7 (b). To make this data useful, it has to be smoothed quite a bit. A robust version of locally weighted scatter plot smooth, described in section 2.2, is used for this. The main reason for using this method is that it handles outliers efficiently, which removes some of the unwanted spikes in the signal. A span of 150 samples is used for the smoothing. This is found through experimentation and established to be adequate for all data sets. An issue with a span this big, is that the velocity in the beginning and in the end of the race will be inaccurate. But since these two parts wont have any influence on the work further, the issue is found irrelevant.

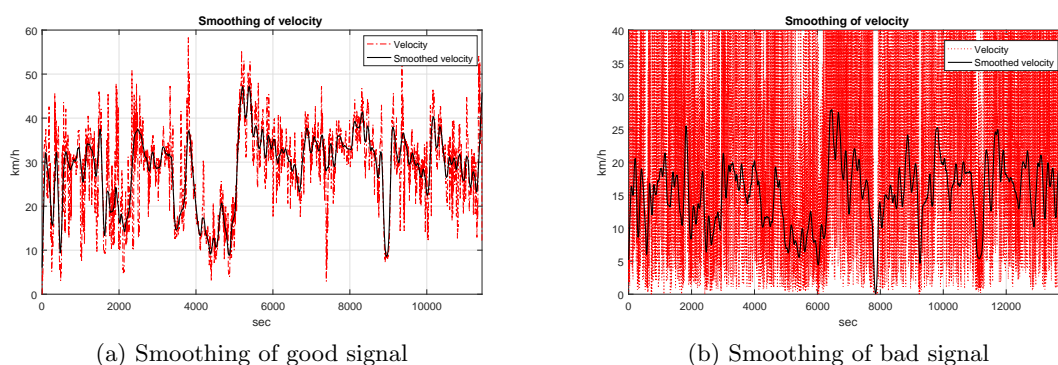


Figure 3.7: Smoothing the velocity signal to locate a comparable trend. (a) is a data set with a relatively good signal where the trend is visible in the original signal, but is more clear after smoothing. (b) is a data set where it is impossible to read any trends before smoothing. After smoothing the trend is prominent.

As illustrated in figure 3.8, the data set is now increased with a new *struct*-element called *Resampled*, where all signals now has a 1 Hz sample rate, and contains the following signal

vectors:

1. Interpolated time, $t_I(m)$
2. Interpolated distance, $d_I(m)$
3. Interpolated heart rate, $hr_I(m)$
4. Interpolated normalized heart rate, $hr_{I,norm}(m)$ (not in use)
5. Interpolated velocity, $v_I(m)$
6. Interpolated and smoothed velocity, $v_{I,smooth}(m)$
7. Interpolated elevation, $elev_I(m)$ (if any)

Additionally, the number of samples discarded during the pre-processing are saved under *Discarded*.

Fields	abc	Id	WatchData	NeededData	Int_puls	GroupIndicator	Norm_puls	SplitTimes	Resampled	Discarded
1		'50226'	4126x7 double	2x13 cell	4126x1 dou...	0	4126x1 double	3x6 cell	11437x7 double	1x1 struct
2		'50399'	4698x7 double	2x13 cell	4698x1 dou...	0	4698x1 double	3x6 cell	14810x7 double	1x1 struct
3		'50409'	5855x7 double	2x13 cell	5855x1 dou...	0	5855x1 double	3x6 cell	12600x7 double	1x1 struct
4		'50480'	4350x7 double	2x13 cell	4350x1 dou...	0	4350x1 double	3x6 cell	14379x7 double	1x1 struct
5		'50556'	6630x7 double	2x13 cell	6630x1 dou...	0	6630x1 double	3x6 cell	14039x7 double	1x1 struct
6		'50564'	6319x7 double	2x13 cell	6319x1 dou...	0	6319x1 double	3x6 cell	12087x7 double	1x1 struct
7		'50638'	12686x7 double	2x13 cell	12686x1 do...	0	12686x1 double	3x6 cell	12686x7 double	1x1 struct
8		'50062'	5907x7 double	2x13 cell	5907x1 dou...	1	5907x1 double	3x6 cell	13251x7 double	1x1 struct
9		'50106'	7106x7 double	2x13 cell	7106x1 dou...	1	7106x1 double	3x6 cell	12989x7 double	1x1 struct
10		'50153'	5805x7 double	2x13 cell	5805x1 dou...	1	5805x1 double	3x6 cell	11188x7 double	1x1 struct
11		'50187'	3720x7 double	2x13 cell	3720x1 dou...	1	3720x1 double	3x6 cell	13525x7 double	1x1 struct
12		'50200'	4139x7 double	2x13 cell	4139x1 dou...	1	4139x1 double	3x6 cell	12238x7 double	1x1 struct
13		'50286'	2408x7 double	2x13 cell	2408x1 dou...	1	2408x1 double	3x6 cell	14693x7 double	1x1 struct

Figure 3.8: Example of dataset after interpolation and smoothing.

3.4 Split time detection

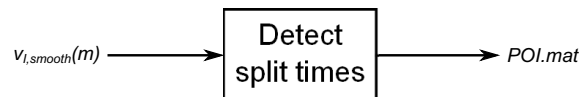


Figure 3.9: Detect locations for the split times with use of the velocity signal.

To be able to compare the contestants data correctly, it is imperative to use the same part of the signals for all the contestants, i.e. they need to be in the same part of the

race. The first idea was to use the official split times, to extract the same race segment for everyone from these. But when comparing these times with the sports watch data, it was not an approach that would suffice. There are mainly two reasons for this. As figure 3.10 clearly shows, there is a deviation between the distance at where the official split times are supposed to be and the distance reached for the majority of the contestants according to their watches. There are two probable causes for this. Either the majority of the watches or the official distances are wrong, the latter seems more probable.

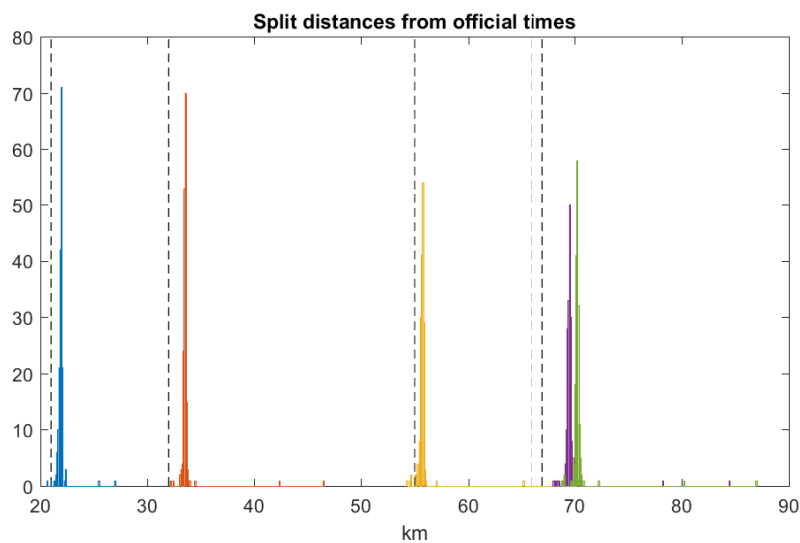


Figure 3.10: The histograms are the contestants and how far they have reached at the official split times according to their watch. The dotted lines are at which distance the official split times are registered.

Hence, another solution is needed. When studying the contestants velocity signals, especially after it has been smoothed, a common trend can be observed. In light of this, a new idea is to use these trends to pinpoint where in the race the contestants are, independent of the time. The most distinguished trend is around where split time 4 and 5 are and it corresponds to the Tinghaug hill. The velocity will reach a minimum up this hill, which can be found with relative ease. Both these points are located, for every contestant, between 68 and 71 km. Hence, a method is devised to automatically locate these points, see algorithm 3. An example of the desired result from this method is shown in figure 3.11.

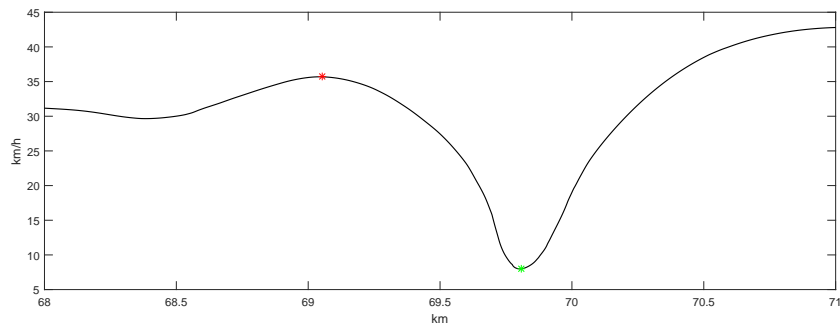
Algorithm 3: Detection of split times

```

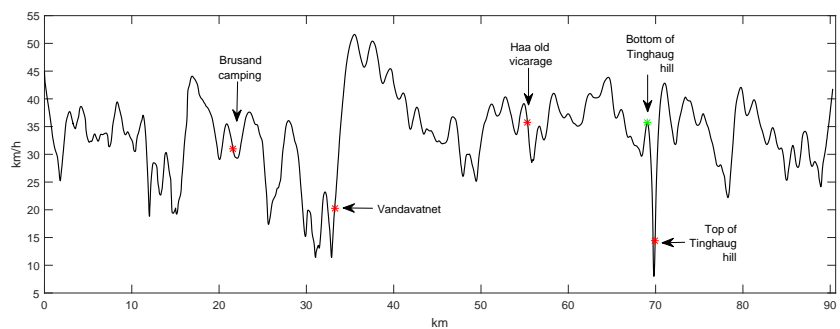
for all contestants do
    Pinpoint the lowest velocity between 68 and 71 km
    Locate the last local maximum before this and set it as split time 4
    Distribute the remaining split times according to the official split times
end

```

One problem is that this method wont locate the correct local maximum in all contestants velocity. To correct this, all signals are plotted and visually inspected to manually adjust those with incorrect local maximum. An example of this visual inspection can be seen in figure C.1 in appendix C. When all correct points are located, the data is saved is a .mat file *POI.mat* for the tasks ahead.



(a) Split time at Tinghaug hill



(b) Split times throughout the race

Figure 3.11: In (a) the green asterisk (*) is the point between 68 and 71 km with the lowest velocity from on of the contestants. When this point is found, the velocity peak before this point is the bottom of Tinghaug hill, shown as the red asterisk (*). This is approximately where split time 4 is. Thereafter in (b), the difference between the official registered split times are used to approximate where the other split times * are compared to split time 4 *

3.5 Race segments

To classify with the data from the contestants sports watches, one need to locate the sections of the race that will provoke a clear increase in the contestants heart rate. The hills in the race are the first segments that comes to mind and these will probably be the parts where the increase in heart rate will be most prominent. Of the hills in the race, one of them will be studied in detail and another hill will be tested as a validation hill. These

two are Tinghaug hill, which is between split time 4 and split time 5, the other is from one of the last inclines up to Vandavatnet at split time 2. Why Tinghaug hill is chosen as the one to be studied in detail has several reasons. Mainly it is because the split times 4 and 5 are close, which makes it possible to choose a segment that is equal for all contestants with high accuracy. Another important reason to choose this particular hill is that the incline is quite steep. Hence, the majority of the contestants will probably come close to their maximum heart rate during this incline. An illustration of Tinghaug race segment is shown in figure 3.12.

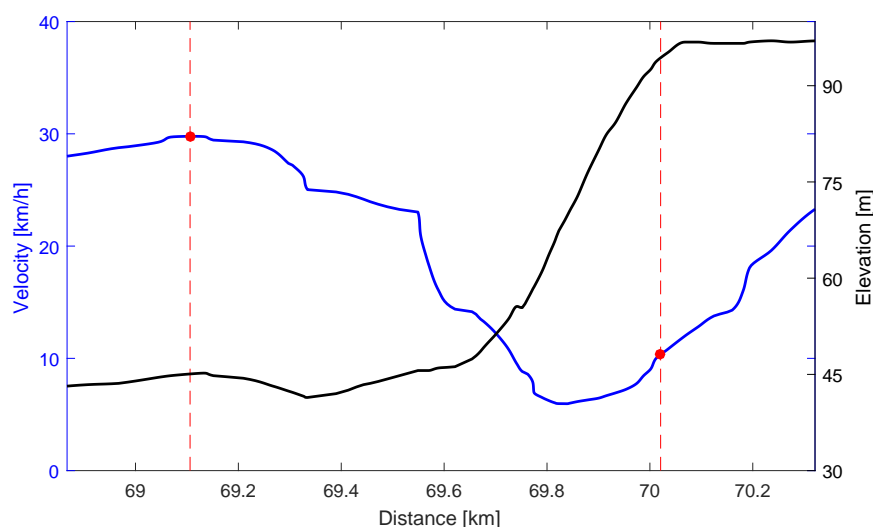


Figure 3.12: Race segment from Tinghaug hill between the red dotted line.

The second hill of interest is found in a similar way as split time 4 as described in section 3.4. But, as with split time 4, the method to locate local maximum struggles to locate the correct one in several of the velocity signals. To fit all the contestants correctly to this segment, a visual inspection is also needed in this case. From this, some will require some manual adjustment. Due to this, there are probably some more inaccuracies in this segment than the main one.

3.6 Classifying data from the pulse watches

After the race segments are chosen and adjusted for all contestants, a number of features will be extracted from their heart rate signal, velocity signal or a combination of these. This will be described in detail in section 3.7. With a subset from these features, a *Naive Bayes*-classifier (described in section 2.3.1) will be trained. Two features selection methods, described in section 3.9, will be tested to select the best subset of the available

features. Finally, all subsets selected with these two methods will be validated with *leave-one-out* cross-validation.

Algorithm 4: Classification with leave-one-out cross-validation.

```

for all contestants do
  Set contestant as validation set
  Remove contestant from training set
  Normalize the training set
  Normalize the validation data with training set
  Train classifier model
  Predict validation data with model
  Add result to predicted class vector
end
Compare predicted class with true class
Calculate accuracy, true positive rate and true negative rate.

```

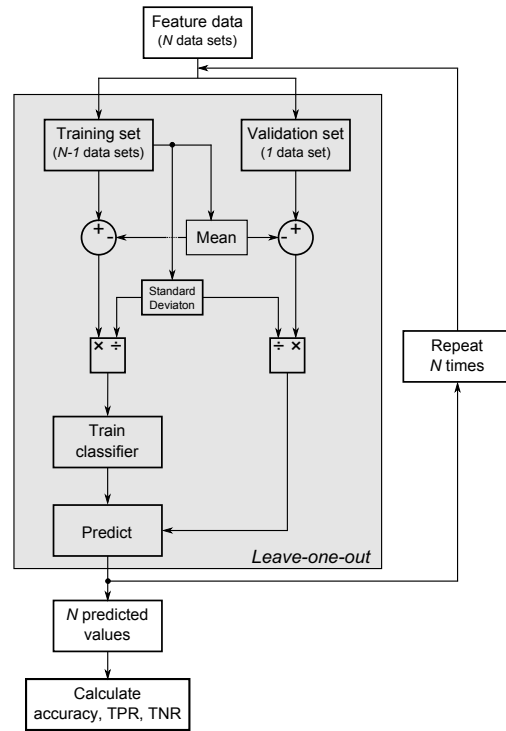


Figure 3.13: Flow chart for classification with leave-one-out cross-validation

3.6.1 Leave-one-out cross-validation

Since the amount of data to train and validate the classifier is finite, a method is needed to use the same data for both. There are several different methods available for this problem. In this thesis *leave-one-out* cross-validation is chosen, which is an N -fold cross-validation where N is the number of training data sets available. The training of the classifier is done using $N-1$ data sets and then validated with the excluded set of data. This is repeated N times with a different set of data excluded each time. Since the learning and validation is done N times, it comes with high computational cost. Nevertheless, this method makes the most of the small amount of available data. Also, since the data is very unbalanced this is thought to be the best approach so all classes are represented in every validation. Algorithm 4 and figure 3.13 illustrates this process.

3.7 Feature extraction

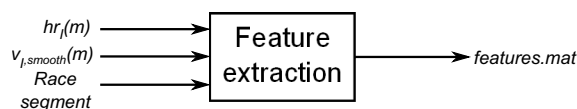


Figure 3.14: Feature extraction.

In this section there is a presentation of the various features that will be used when trying to classify the data from the sports watches. Some of these features are global, i.e. they are not dependent on any specific parts of the race. The remainder of the features are specific to the race segments mentioned in section 3.5.

3.7.1 Global features

These are features that describes the contestants physical properties, it can be an indicator on the health or how hard they have pushed themselves during the race. All these features may affect the heart rate in some way and may prove to be important when trying to classify the data.

Age

Studies [28] [24] show that age is an important factor on maximum heart rate. From these there are several different ways to predict this:

- $HR_{max} = 220 - age$
- $HR_{max} = 211 - 0.64 \cdot age$
- $HR_{max} = 208 - 0.7 \cdot age$

Since these are different methods of predicting maximum heart rate, the heart rate in this thesis will not be normalized as in previous work [25]:

$$HR_{norm} = \frac{HR}{HR_{max}} \quad (3.3)$$

Instead age will be used as a feature independent from heart rate.

Feature name: *Age*

Race time

Total race time as a feature will show how hard the contestants have pushed themselves and/or how physically fit they are. Alone it will probably not be useful as a feature, but when using it in combination with other features it might be.

Feature name: *RaceTime*

Body Mass Index

Body Mass Index (BMI) is an indicator of the contestants physical properties.

$$BMI = \frac{weight [kg]}{height^2 [m^2]} \quad (3.4)$$

According to *Norsk helseinformatikk* [6] one is overweight if the BMI exceeds 25 and one starts to be obese if the BMI exceeds 30. Since obesity is linked to several cardiovascular changes [31] it is interesting to test if this might be useful as a feature. One problem is that BMI does not differentiate between muscle mass and fat mass. Hence, an athletic person can be obese according to BMI, but in reality have a low fat percentage.

Feature name: *BMI*

hsTnI

As mentioned, there has been measurements of hsTnI for each contestant approximately 24 hours before, 3 and 24 hours after the race. High values is an indicator for myocardial damage, but some of the contestants with high hsTnI values did not have any damages to the heart. These three measurements will be used as features both when classifying myocardial damage and classifying race time (the class definitions will be described in the next section). When classifying myocardial damage one problem is that contestants both with and without any heart problems may have elevated hsTnI values. But hopefully with some testing one can show that one or more of the hsTnI features in combination with other features might prove to be useful.

Feature names: *hsTnI1*, *hsTnI2*, *hsTnI3*

3.7.2 Segment specific features

Whereas the global features are independent from the race segments, the following features are extracted directly from these. Since the same features are extracted from both sections, they are only described once. All the following features are mainly based on two signals,

heart rate and velocity, see figure 3.15. The only exception is the calculation of force which is only dependent on the contestants weight and the incline in the given segment.

Heart rate signal

$$hr_I(m) \quad (1\text{Hz sampling rate, length } M) \quad (3.5)$$

Velocity signal

$$v_{I,smooth}(m) \quad (1\text{Hz sampling rate and smoothed, length } M) \quad (3.6)$$

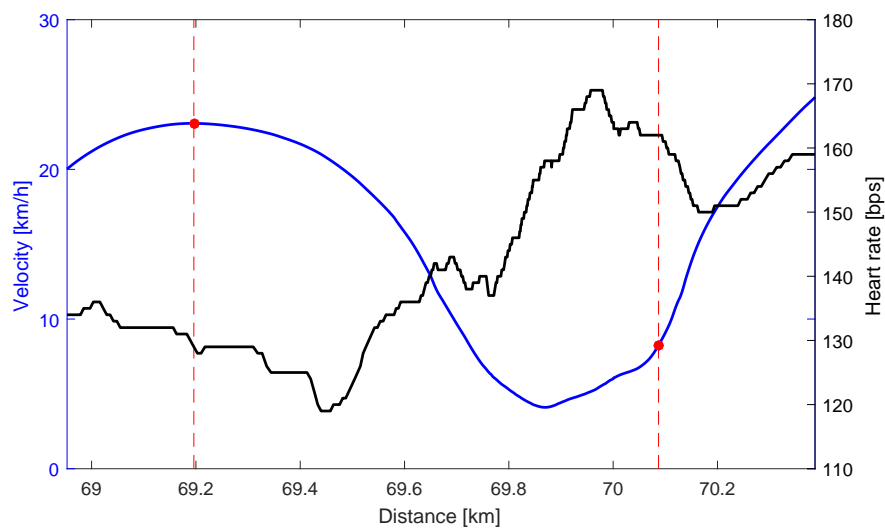


Figure 3.15: Example of heart rate and velocity signals from Tinghaug hill. The signal vectors are between the dotted lines.

From the signals in equation 3.5 and 3.6 the following features are extracted with their feature name in parenthesis.

Mean heart rate (*meanHR*)

$$\mu_{\text{HR}} = \frac{1}{M} \sum_{m=1}^M hr_I(m) \quad (3.7)$$

Variance heart rate (*varHR*)

$$\sigma_{\text{HR}}^2 = \frac{1}{M} \sum_{m=1}^M (hr_I(m) - \mu_{\text{HR}})^2 \quad (3.8)$$

Maximum heart rate (*maxHR*)

$$\max(hr_I(m)) \quad (3.9)$$

Median heart rate (*medianHR*)

Difference heart rate (*diffHR*)

$$\max(hr_I(m)) - \min(hr_I(m)) \quad (3.10)$$

Mean velocity (*meanSpeed*)

$$\mu_v = \frac{1}{M} \sum_{m=1}^M v_{I,smooth}(m) \quad (3.11)$$

Variance velocity (*varSpeed*)

$$\sigma_v^2 = \frac{1}{M} \sum_{m=1}^M (v_{I,smooth}(m) - \mu_v)^2 \quad (3.12)$$

Median velocity (*medianSpeed*)

Regression coefficient velocity (*RegCoeff*)

$$\Delta_v = \frac{v_{I,smooth}(M) - v_{I,smooth}(1)}{M} \quad (3.13)$$

Correlation between heart rate and velocity (*CorrHRspeed*)

$$\rho = \frac{\text{cov}(hr_I, v_{I,smooth})}{\sigma_{HR} \sigma_v} \quad (3.14)$$

where

$$\text{cov}(hr_I, v_{I,smooth}) = \frac{1}{M} \sum_{m=1}^M ((hr_I(m) - \mu_{HR})(v_{I,smooth}(m) - \mu_v)) \quad (3.15)$$

Force (*Force*)

To simplify this calculation, friction and air drag are assumed to be zero. In addition, mass is approximated as the persons measured weight plus 10 kg for the bike and other equipment. Since few of the contestants have any elevation data logged, the angle of the incline, α , is approximated as an average from these few, and used for all contestants. An illustration of the forces at play when calculating the average force, \vec{F} , needed to move up this incline are shown in figure 3.16

$$\vec{F} = \vec{F}_g \text{Sin}(\alpha) = mg \text{Sin}(\alpha) [N] \quad (3.16)$$

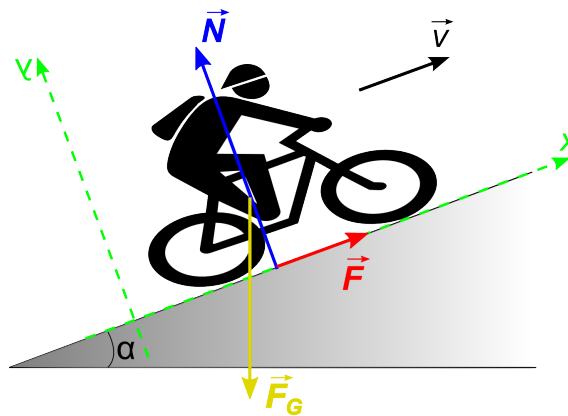


Figure 3.16: Illustration of average incline with the variables used for calculating force.

Work (*Work*)

$$W = \vec{F}s \text{ [KJ]} \quad (3.17)$$

Power (*Power*)

$$P = \vec{F}v \text{ [W]} \quad (3.18)$$

Recovery heart rate is the heart's ability to return to normal levels after physical activity [12]. A heart that is healthy will recover more quickly than an unhealthy or one that is not accustomed to exercise. In light of this four more features are extracted from the Tinghaug segment. These are:

Heart rate one minute after top (*HRoneMinAfterTop*)

Heart rate two minutes after top (*HRtwoMinAfterTop*)

Distance cycled one minute after top (*DistOneMinAfterTop*)

Distance cycled two minutes after top (*DistTwoMinAfterTop*)

Since most of the riders probably will not slow down after they have reached the top of this incline, it is uncertain if this can be regarded as recovery time. Nonetheless, these are features that need to be included in the experiments, to explore if there are any connections of value.

3.8 Class definitions

The main goal for this thesis is to test if it is possible to classify myocardial damage or high hsTnI values with the data from the sports watches. To do this, one needs to define

some classes that is going to be classified. In this section, four will be defined.

- Original - the original class definitions from previous work, which were a three class problem. These are redefined to a two class problem, which is with or without myocardial damage. (181 contestants in total).
- CT - as with *original* this is also with or without myocardial damage. The main difference here is that the whole *without damage* class have undergone a CT-scan and therefore it is certain that these do not have any myocardial damage. (53 contestants in total).
- Troponin - two class problem where one is over and the other is under a hsTnI threshold. (181 contestants in total).
- Race time - two class problem, where one is over and the other is under mean race time. (181 contestants in total).

3.8.1 Original

From earlier work the contestants were split in three different classes, labelled with a class indicator (CI-X). CI-0 are contestants with high hsTnI values and *with* findings during CT-scan (7), CI-1 are those with high hsTnI values *without* any findings during CT-scan (17) and finally CI-2 are those with low hsTnI values (157). The number in the parenthesis are number of contestants in either class. Probability density functions (PDFs) for a selection of the features can be seen in figure 3.17.

But since one of the goals are to test if there are any connection between the sports watch data and myocardial damage, this definition is redefined. Based on the original class definition with three classes, CI-1 and CI-2 are combined. Hence, there are now two classes of which one have heart damage (7) and one does not (174). The PDFs for this definition is seen in figure 3.18.

3.8.2 CT

A control group containing 40 contestants with low hsTnI had to undergo a CT-scan. Based on the scan, these were confirmed healthy. CI-1 will therefore consist of the contestants over the hsTnI threshold *without* findings during CT-scan and the proven healthy with low hsTnI. CI-0 is equal as the *original* class definition. Hence, there are two classes (CI-0 and CI-1) which can be defined with certainty as with (7) or without (46) myocardial damage. The PDFs for a selection of features can be seen in figure 3.19.

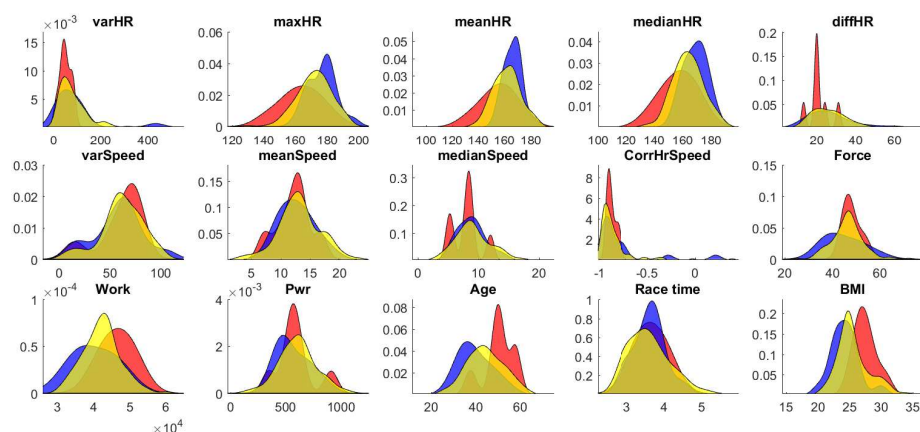


Figure 3.17: Density functions for the original class definition with three classes. **CI-0**: High hsTnI values with heart damage, **CI-1**: High hsTnI values with heart damage and **CI-3**: Low hsTnI values.

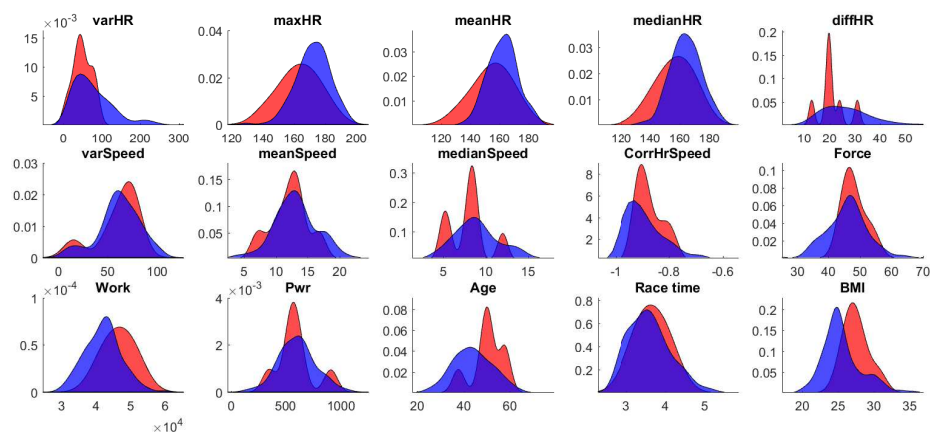


Figure 3.18: Density functions for the original class definition with two classes. **CI-0** - with myocardial damage. **CI-1** and **CI-2** combined - without myocardial damage.

3.8.3 Troponin

The goal for this class definition is to see if there are any connections between the data from the sports watches and the hsTnI values three hours after the race. Here one will try to classify two groups as low (CI-0) or high (CI-1) hsTnI value. As a threshold to divide these two classes, the mean hsTnI for all contestants are used at first. Several other threshold values are later tested, to see if there are thresholds which are easier classify than other. Because of this, the number of contestants in either class will vary with the

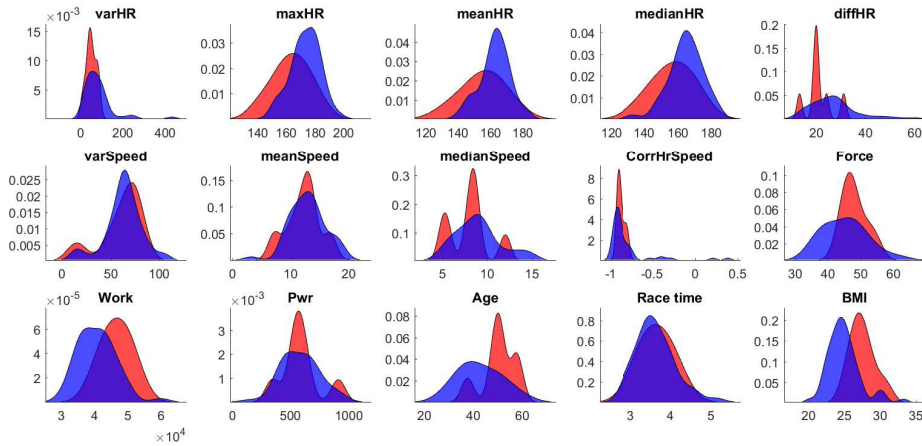


Figure 3.19: Density functions for the CT class definition. **CI-0** - with myocardial damage. **CI-1** - without myocardial damage.

chosen threshold. PDFs for a selection of features from this class definition, with mean hsTnI as threshold value, can be seen in figure 3.20.

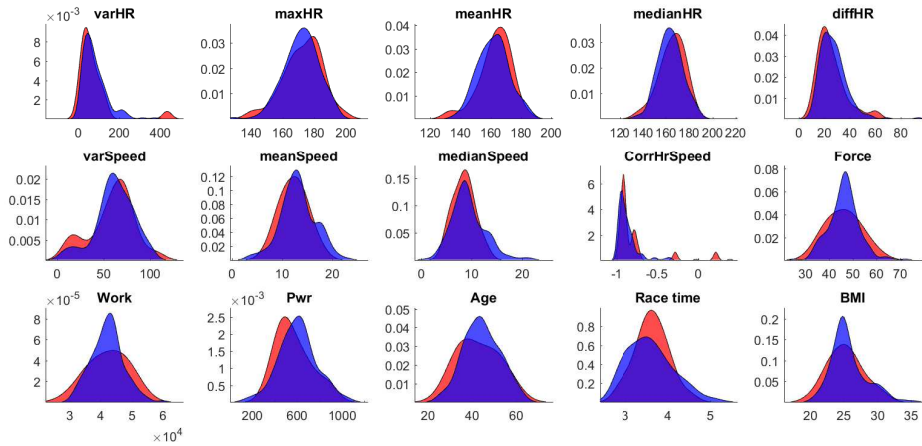


Figure 3.20: Density functions for the troponin class definition. **CI-0** - Under mean hsTnI. **CI-1** - over mean hsTnI.

3.8.4 Race time

This last class definition is based on the total race time for the contestants. The whole group is divided in two, whereas class CI-0 is above the average race time and class CI-1 is below. With this class definition the goal is to test if the hsTnI features are relevant. If so, there may be a connection between the hsTnI values and how hard the contestants

have pushed themselves or something about their level of fitness. PDFs for a selection of features from this class definition can be seen in figure 3.21.

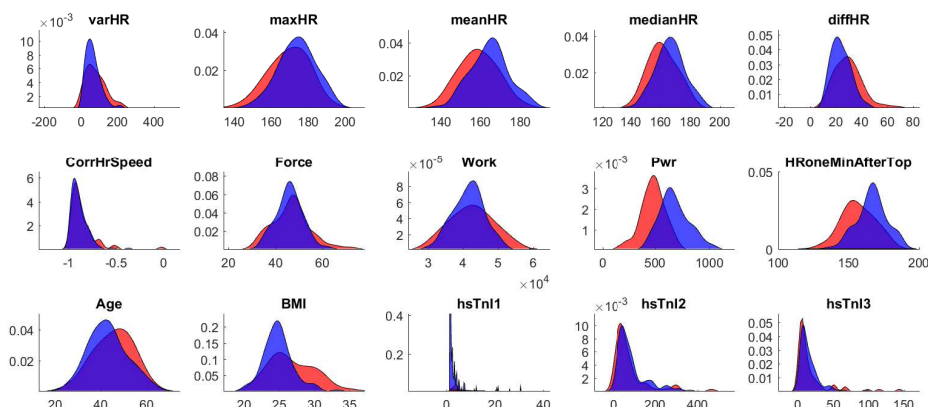


Figure 3.21: Density functions for the race time class definition. **GI 0** - above average race time. **GI 1** - below average race time.

3.9 Feature selection

In this study the number of features available to design a classification system is quite large. Even though *Naive Bayes* classifier will be used and the *curse of dimensionality* won't matter much, there are other reasons to reduce the number of features. The most obvious one is to reduce the computational complexity, but a more important reason is to remove the features that are redundant and/or irrelevant. As seen in section 3.8, the PDFs for the different class definitions are not a visually good indicator on how good the individual features will perform. Therefore, two different approaches will be used to select a subset of l features from n available which will create the best l -dimensional feature set.

3.9.1 Exhaustive search

Since the number l is unknown, all combinations of features are tested for all $l = 1, 2, \dots, n$. As an example, to select the best subset of the following feature vectors x_1, x_2, x_3 , which is $n = 3$ different features. With this method, a classification is done with each of the following feature subsets:

$$\begin{array}{l} [x_1]^T \quad [x_1, x_2]^T \quad [x_1, x_2, x_3]^T \\ [x_2]^T \quad [x_1, x_3]^T \\ [x_3]^T \quad [x_2, x_3]^T \end{array}$$

In total this is seven runs, which might not be much. But when the number of different features n increases, the number of classifications needed for an exhaustive search will increase exponentially, $2^n - 1$. With $n = 19$ different features, which is the most tested in the experiments, this is now 524 287 classifications and will be quite challenging computationally. The best subset of features are chosen by a criterion which will differ between the different class definitions. These criteria are described more in detail when the experiments are presented in chapter 4.

3.9.2 Sequential forward selection

This method is based on a method with the same name in *Pattern Recognition* [29]. Compared to exhaustive search which will select the best available feature subset, this a suboptimal method. But it is considerably less complex computationally. As an example, from the available features x_1, x_2, x_3 the subset is found as follows:

1. Classify the training data with all the one-dimensional features, $[x_1]^T, [x_2]^T, [x_3]^T$, separately and select the best based on a criterion.
2. Then classify all two-dimensional feature combinations with the best feature from the previous step, $[x_1, x_2]^T$ and $[x_2, x_3]^T$. Now select the best of these.
3. Finally, for this example with three features, form a tree-dimensional feature vector $[x_1, x_2, x_3]^T$, and classify the training data with these.

If there are more features available than the example above, keep selecting the best combination from the previous step and increase dimensionality until the dimension equals number of features. Since the dimension of the best of these suboptimal feature sets are not known, all dimensions are tested. In total this will require $n^2 - \frac{n(n-1)}{2}$ classifications.

In contrast to exhaustive search, this method will with $n = 19$ only need 190 classifications.

4. Experiments and results

This chapter presents the various experiments done and the results from these. There is an explanation on the test set up, which methods and parameters used. Also the criteria used for the determination of the best feature subset are described.

After visual inspection of all velocity/heart rate signals in the Tinghaug race segment, ten of the contestants were excluded. There are different reasons for this, but mainly it was due to poor data sets or that the race segment itself was too inaccurate. As these are all from the large group of presumably healthy contestants, this should have had minimal effect on the result. Because of this, the number of contestants in the following experiments will differ compared to the numbers presented in section 3.8.

Two examples of the excluded data sets can be seen in figure 4.1. The left figure shows a sudden drop in hear rate, the reason for this is not known. The right figure shows the heart rate signal from a contestant that has seemingly stopped. In addition, it seems like the logging of data stops when the contestant is standing still.

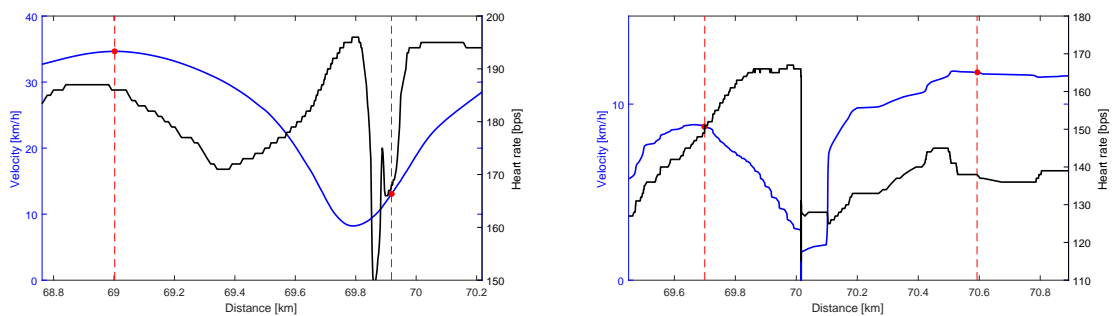


Figure 4.1: Two examples of data excluded from experiments.

There are four different ways the classes were defined, and with all these class definitions the two methods described in section 3.9.1 and 3.9.2 were used similarly to select feature subsets. The only difference was that the *troponin* and *race time* classes used a different criteria than *original* and *CT* to select the best feature subset.

In short, *exhaustive search* iterated through all possible feature combinations from a chosen feature set. Each feature combination was then classified individually, before the optimal combination was selected based on a given criteria. Opposed to *exhaustive search*, the *sequential forward selection* would only classify a small portion of the possible feature combination and select a suboptimal but hopefully decent subset of features. The classification process itself was equal in all experiments, described in section 3.6.

All results from the experiments in this chapter are presented in tables with accuracy (ACC), true positive rate (TPR) and true negative rate (TNR). Furthermore, since these results are derived from a sample of a larger population, a 95% confidence interval is calculated, described in section 2.4. This is a measurement on how reliable the results are, and is shown in the square brackets behind each results. Finally, Matthews correlation coefficient (MCC) is used as a second measurement on the results reliability, described in section 2.5.

At the bottom of each table, the subset of features that produced the results are presented. In addition to these results, confusion matrices for all the best achieved results can be found in appendix C.

4.1 Prediction of myocardial damage

The goal for the experiments in this section, was to test if there are any information logged in the sports watched that could predict myocardial damage.

4.1.1 Original class definition

The training data were divided in two classes as described in section 3.8.1

TP - True positive class with contestants that has high values of hsTnI and proven myocardial damage. Contains 7 test subjects.

TN - True negative class with contestants that has high values of hsTnI without myocardial damage and those with low hsTnI values (presumably without any heart issues). Contains 164 test subjects.

Exhaustive search

This experiment tested all possible combinations of the available features, as described in section 3.9.1. In this class definition, TP was the most important class to classify cor-

rect. Since the data set is quite unbalanced, these tests were also performed with several different misclassification costs. This was a time consuming process and it took, with 19 different features, around 24 hours running in parallel with 80 processing cores. These calculations were done on a server dedicated for these types of demanding operations. To do this experiment on a single processing core, it will use approximately 80-times more time to complete these calculations. Afterwards, when exhaustive search had iterated through all possible combinations, the best subset was chosen based on criteria 1.

Criteria 1: Selection of feature subset when classifying original/CT class definitions.

1. Select all subsets with the highest TP prediction rate
 2. Select the subset with highest TN prediction rate from step one.
-

With exhaustive features selection and feature subset chosen by these criteria, the classification results are presented in table 4.1.

	No cost		Cost 10		Cost 20	
ACC ¹	0.92	[0.87 0.96]	0.81	[0.75 0.86]	0.73	[0.66 0.79]
TPR ²	0.00	[0.00 0.40]	0.86	[0.47 0.99]	0.86	[0.47 0.99]
TNR ³	0.96	[0.92 0.98]	0.81	[0.74 0.86]	0.72	[0.65 0.79]
MCC ⁴	-0.04		0.32		0.25	
Features	<i>Age</i> <i>medianHR</i>	<i>Force</i> <i>medianSpeed</i>	<i>CorrHrSpeed</i>			

Table 4.1: Classification results from original classes - exhaustive search feature selection

Sequential forward selection

As described in section 3.9.2, this is a suboptimal method for finding a good subset of feature. But the fact that it is considerably more efficient in terms of computational speed, it could test a lot more features than exhaustive search. Compared to the previous method, these calculations could with ease be done on a regular computer and each test were completed in matter of minutes. This made it possible to run more tests in a shorter amount of time. With this method and the use of criteria 1 to select a subset of features, the results are shown in table 4.2.

¹Accuracy

²True Positive Rate (also known as *sensitivity*)

³True Negative Rate (also known as *specificity*)

⁴Matthews correlation coefficient

	No cost		Cost 15		Cost 30	
ACC	0.92	[0.87 0.96]	0.79	[0.73 0.85]	0.74	[0.67 0.80]
TPR	0.14	[0.01 0.53]	0.86	[0.47 0.99]	0.86	[0.47 0.99]
TNR	0.96	[0.91 0.98]	0.79	[0.72 0.85]	0.73	[0.66 0.79]
MCC	0.08		0.30		0.26	
Features	<i>Age</i>	<i>Force</i>	<i>CorrHrSpeed</i>	<i>BMI</i>		
	<i>maxHR</i>	<i>meanHR</i>	<i>medianSpeed</i>			

Table 4.2: Classification results from original classes - sequential forward feature selection

4.1.2 CT class definition

The training data were divided in two classes as described in section 3.8.2

TP - Contestants with high values of hsTnI and proven myocardial damage. Contains 7 test subjects.

TN - Contestants with high values of hsTnI without any myocardial damage and the control group with low hsTnI (proven without any heart issues). Contains 41 test subjects.

The methods used to select the optimal feature subset with exhaustive search, and to select a good suboptimal subset with sequential forward selection, were exactly the same as with the *original* class definition. The only difference was that TN in this experiment contained fewer contestants.

Exhaustive search

	No cost		Cost 5		Cost 10	
ACC	0.78	[0.64 0.87]	0.86	[0.73 0.93]	0.84	[0.71 0.92]
TPR	0.29	[0.08 0.65]	0.86	[0.47 0.99]	0.86	[0.47 0.99]
TNR	0.86	[0.72 0.94]	0.86	[0.72 0.94]	0.83	[0.69 0.92]
MCC	0.14		0.58		0.55	
Features	<i>Age</i>	<i>Force</i>	<i>CorrHrSpeed</i>	<i>BMI</i>		
	<i>maxHR</i>	<i>meanHR</i>	<i>medianHR</i>			
	<i>diffHR</i>	<i>RaceTime</i>	<i>medianSpeed</i>			

Table 4.3: Classification results from CT classes - Exhaustive search feature selection

Sequential forward selection

	No cost		Cost 5		Cost 10	
ACC	0.80	[0.87 0.96]	0.84	[0.71 0.92]	0.80	[0.66 0.89]
TPR	0.43	[0.16 0.75]	0.86	[0.47 0.99]	0.86	[0.47 0.99]
TNR	0.86	[0.72 0.94]	0.83	[0.69 0.92]	0.79	[0.64 0.89]
MCC	0.26		0.55		0.49	
Features	<i>Age</i>		<i>Force</i>	<i>CorrHrSpeed</i>		<i>BMI</i>
	<i>RaceTime</i>		<i>meanHR</i>	<i>maxHR</i>		

Table 4.4: Classification results from CT classes - sequential forward feature selection

4.2 Prediction of raised hsTnI values

The goal for the experiments in this section, was to test if there are any information logged in the sports watches that is correlated to the contestants hsTnI values. With the two class definitions in these experiments, the goal were tried achieved using two separate approaches.

4.2.1 Troponin class definition

The training data are divided in two classes as described in section 3.8.3

TP - Contestants with hsTnI values over a given threshold. Number of test subjects will vary with the threshold.

TN - Contestants with hsTnI values under a given threshold. Number of test subjects will vary with the threshold.

With this class definition, the goal was to test if there are any connections between the hsTnI values and the available features. As the contestants with high hsTnI values due to proven heart disease could do this difficult, these were excluded from the data prior to the experiment. Furthermore, the features *hsTnI1*, *hsTnI2* and *hsTnI3* were removed from the feature set for obvious reasons.

Both classes were evenly important in this class definition. Based on this, the criteria used to select subsets when predicting predict myocardial damage would not suffice. To weigh in the importance of both classes, criteria 2 were used to select feature subset in this experiment.

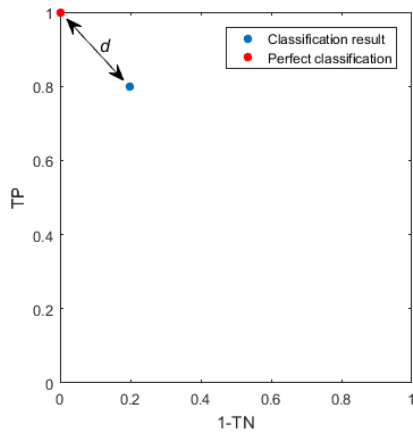


Figure 4.2: ROC ¹graph with the euclidean distance between classification result and perfect classification.

Criteria 2: Selecting best feature subset when classifying hsTnI or race time.

for all classifications do
 | Calculate euclidean distance (d i figure 4.2)
 | to perfect classification.

end

Select feature subset with minimum distance.

Exhaustive search

When using this method to search for the optimal subset of features, the hsTnI threshold value was the contestants combined mean, which was 120 [ng/l].

	No cost		Cost 4		Cost 8	
ACC	0.80	[0.73 0.85]	0.64	[0.57 0.71]	0.39	[0.32 0.46]
TPR	0.03	[0.00 0.17]	0.63	[0.45 0.77]	0.84	[0.68 0.94]
TNR	0.98	[0.94 1.00]	0.65	[0.56 0.73]	0.27	[0.21 0.36]
MCC	0.05		0.22		0.11	
Features	<i>Age</i>	<i>Force</i>	<i>HRtwoMinAfterTop</i>	<i>DistTwoMinAfterTop</i>		

Table 4.5: Classification results from troponin classes - Exhaustive search feature selection

Sequential forward selection

A few different thresholds were tested when searching for a subset of features with this method, these were 100, 150 and 200 [ng/l]. Furthermore, the selected features were testes with several misclassification costs for all thresholds.

¹Receiver operating characteristic graph, sometimes called sensitivity vs (1-specificity) plot. It depicts the relative trade-off between TP and false positive (FP).

	No cost	Cost 4	Cost 8
ACC	0.74 [0.67 0.80]	0.64 [0.57 0.71]	0.40 [0.33 0.48]
TPR	0.06 [0.01 0.20]	0.60 [0.44 0.74]	0.71 [0.55 0.84]
TNR	0.93 [0.87 0.96]	0.66 [0.57 0.73]	0.31 [0.24 0.40]
MCC	-0.02	0.21	0.02
Features	<i>Age</i> <i>diffHR</i>	<i>Work</i> <i>varHR</i>	<i>medianSpeed</i> <i>HRoneMinAfterTop</i>

Table 4.6: Classification results from hsTnI classes with threshold at 100 [ng/l]

	No cost	Cost 4	Cost 8
ACC	0.82 [0.76 0.87]	0.66 [0.59 0.73]	0.51 [0.43 0.58]
TPR	0.00 [0.00 0.15]	0.46 [0.29 0.65]	0.77 [0.58 0.89]
TNR	0.98 [0.93 1.00]	0.70 [0.62 0.77]	0.46 [0.38 0.54]
MCC	-0.06	0.13	0.17
Features	<i>Age</i> <i>DistOneMinAfterTop</i>	<i>diffHR</i>	<i>HRtwoMinAfterTop</i> <i>DistTwoMinAfterTop</i>

Table 4.7: Classification results from hsTnI classes with threshold at 150 [ng/l]

	No cost	Cost 4	Cost 8
ACC	0.90 [0.84 0.93]	0.74 [0.67 0.80]	0.61 [0.53 0.68]
TPR	0.00 [0.00 0.25]	0.64 [0.39 0.84]	0.64 [0.39 0.84]
TNR	0.98 [0.94 1.00]	0.75 [0.68 0.81]	0.60 [0.52 0.68]
MCC	-0.04	0.25	0.14
Features	<i>Age</i> <i>RaceTime</i>	<i>maxHR</i> <i>diffHR</i>	<i>meanHR</i> <i>HRtwoMinAfterTop</i>

Table 4.8: Classification results from TnI classes with threshold at 200 [ng/l]

4.2.2 Race time class definition

For this last class definition the training data were divided in two classes as described in section 3.8.4

TP - Contestants slower than mean race time. Contains 61 test subjects.

TN - Contestants faster than mean race time. Contains 102 test subjects.

The goal for this class definition were to test if there are any connection between the contestants race time and their hsTnI values. There could be issues with the result since

some had high values due to their heart issues. Hence, these contestants were excluded from the data so only the presumably healthy individuals remained. Furthermore, all features related to velocity were excluded since these are directly connected to the race time and would be preferred over the other features with both feature selection methods. As with the *troponin* class definition these experiments also used criteria 2 to select the subset of features.

Exhaustive search

	No cost		Cost 2		Cost 4	
ACC	0.79	[0.72 0.85]	0.74	[0.66 0.80]	0.71	[0.63 0.77]
TPR	0.67	[0.55 0.78]	0.69	[0.56 0.79]	0.70	[0.58 0.81]
TNR	0.86	[0.78 0.92]	0.76	[0.67 0.84]	0.71	[0.61 0.79]
MCC	0.55		0.45		0.40	
Features	<i>Age</i>		<i>BMI</i>	<i>meanHR</i>	<i>hsTnI3</i>	
	<i>diffHR</i>		<i>HRoneMinAfterTop</i>	<i>HRtwoMinAfterTop</i>		

Table 4.9: Classification results from race time classes - exhaustive search feature selection.

Sequential forward selection

	No cost		Cost 2		Cost 4	
ACC	0.74	[0.67 0.80]	0.68	[0.61 0.75]	0.61	[0.54 0.68]
TPR	0.62	[0.50 0.73]	0.67	[0.55 0.78]	0.74	[0.61 0.83]
TNR	0.81	[0.73 0.88]	0.69	[0.59 0.77]	0.54	[0.44 0.63]
MCC	0.44		0.35		0.27	
Features	<i>Age</i>		<i>BMI</i>	<i>meanHR</i>	<i>hsTnI2</i>	
	<i>maxHR</i>	<i>diffHr</i>	<i>medianHR</i>			

Table 4.10: Classification results from race time classes - sequential forward feature selection.

4.3 Validating the results with a different hill

A final experiment was done where the goal was to test the possibility to train the classifier with one hill (Tinghaug) and validate it with another (Vandavatnet). The subset of features that produced the best results in the experiments with the *original* and *CT* class definitions were tested. Moreover, the misclassification cost was the same as with the previous experiments. The test results are presented as confusion matrices, figure 4.3.

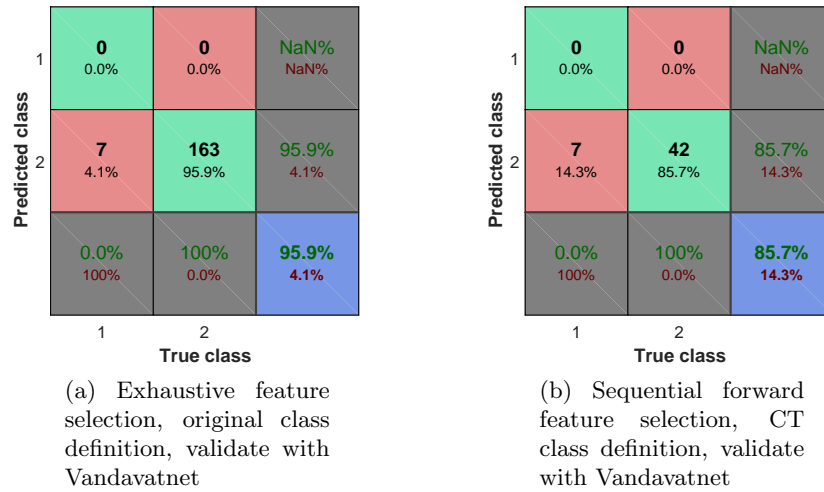


Figure 4.3

5. Discussion

5.1 Data material

Thoroughly pointed out in previous work [25], there are several issues with the data collected from the sports watches. The main problem for the work in this thesis was that there were no consistency as to which rate the signals were sampled. This problem was overcome by interpolating all data sets to the point that all sampling rates were equal. But the fact remains that the data are collected from various watch brands, which may not use equal methods for logging the data.

Another issue worth mentioning, is that some of the sports watches seemingly stops logging data when the contestants are standing still. Since this is a feature available in many sports watches, it may be activated on several of the contestants watches. An illustration of this can be seen in figure 5.1. If this issue was detected and it was within the race segment used for features extraction, these contestants were excluded prior to the experiments.

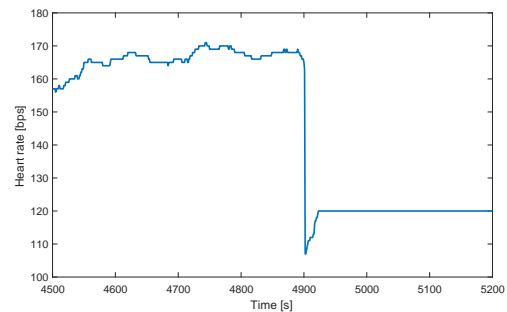


Figure 5.1: Example of watch believed to have paused while contestant is standing still.

As mentioned in section 3.3, there were some issues with a few of the sports watches that did not log the velocity signal. These were estimated as the derivative of the distance. But as pointed out earlier, this could produce a very spiky signal. When smoothing these signals, another issue emerged. Figure 5.2 shows the mean velocity before and after smoothing from a selection of the contestants. Two of these, marked with red boxes, has a sports watch that did not log the velocity. It clearly shows that when smoothing these contestants velocity, the mean velocity are notably changed compared to the rest. Since some of these are from the contestants with myocardial damage, the data were not excluded.

28.5154	21.7977	25.4888	22.6727	23.3126	26.7176	25.8034	24.5555	24.7933	29.1710	23.7149
28.6469	21.6157	25.5094	15.1349	23.2264	27.0442	25.9399	24.4340	25.0334	29.6143	15.4946

Figure 5.2: Example of some contestants mean velocity before and after smoothing. The two contestants within the red boxes did not have velocity data logged, and is estimated as the derivative of distance before smoothing.

5.2 Methods

Since there are issues with the velocity signals from several of the sports watches, other methods might be more suitable for the smoothing. One idea might be to use some form of adaptive smoothing algorithm, where the troublesome signals are detected and smoothed different than the rest.

When it comes to detection of the real split times in the race, some improvements may be needed. Firstly, since the detection of split time 4 is based on the velocity, the issue with smoothing might introduce inaccuracies. These inaccuracies are tried corrected after visual inspection, but a more sophisticated and less time consuming method for this might be convenient. Secondly, since there are suspicion that some watches might pause when the contestants are standing still, another problem arises. The other four split times are placed based on the time difference in the official split times and distributed from split time 4 according to each contestants watch data. If a watch pauses due to a contestant sanding still, this distribution will not be correct. This might cause issues when selecting the race segments, since these are based on where the split times are detected.

The main race segment, which is between split time 4 and split time 5, should be quite accurate. This segment is short and through visual inspection some (but few) contestants were excluded prior to the experiments. The other segment, near split time 2, used a method to locate some local maxima and with these define the segment. Due to some issues with locating these maxima, this segment is not as accurate as the main segment.

After the segments were defined and the features had been extracted from these, a *Naive Bayes* classifier was trained. Several other classifiers, such as *Support Vector Machine* and *Decision tree* were initially tested. But the *Naive Bayes* preformed better in every test, and therefore it was chosen as the preferred classifier. A comparison of the different classifiers are shown in appendix D. The subset of features used when comparing these are equal. Since the optimal subset for *Naive Bayes* may not be the best for the other classifiers, a new search for subsets could give a more favourable results for these.

Since the data to train the classifier is limited, the same data is used for validation. The *leave-one-out* cross-validation method is used in the experiments in this thesis. 10-fold cross-validation is proven [21] better, but is hard to realize with data as unbalanced as the available data is.

5.3 Results

The first goal were to test if any connections can be found between the data from the sports watches and myocardial damage. The results presented in tables 4.1- 4.4 shows that it should be possible to use features derived from sports watches to predict myocardial damage. There is also a clear trend of which features these experiments selected. With the exception of exhaustive search in the original class definition, all the other experiments had the following features among the selected: *Age*, *Force*, *CorrHrSpeed*, *BMI*, *meanHR* and *maxHR*. Why the best from exhaustive search in the original class definition only had three of these features is uncertain, but it might be a case of overfitting. When analysing the best results more in detail, it shows that the classifier correctly predicts six out of seven with myocardial damage, which is a TPR of 0.86. But, since there are very few samples in this class, the 95% confidence interval becomes wide: [0.47 0.99]. The TNR from the best results varies from 0.79 to 0.86.

When measuring the quality of the classification, the CT class definition gains a MCC between 0.5 and 0.6. This is interpreted as a fairly good prediction according to table 2.1. The original class definition has nearly as good results in terms of TPR and TNR, but the MCC is barely over 0.3, which is not so good. This is believed to be a result of a more unbalanced data set. Lastly, the one TP that is misclassified in all experiments, is the same individual. When studying the information provided by the hospital, this individual has “suspicion of myocardial damage” whereas the other six have confirmed findings as remarks from CT-scan.

The second goal were to test if any connections can be found between the sports watch data and the contestants hsTnI values. Both approaches to test this had relatively poor results. When using the watch data to predict if the contestants were over or under a hsTnI threshold, majority of the results are barely better than random guess. The best results were achieved with a hsTnI threshold value of 200 [ng/l], with both TPR and TNR over 0.6. But in despite of this prediction rate, the MCC indicates quite poor results.

The other method used to test the connection between hsTnI and watch data, was to search for features that could predict the race time. If one or more of the hsTnI features were one of these, it could indicate that there is a connection between the hsTnI values and how fast the contestants completed the race. Both feature selection methods used in this experiment gave hsTnI as one of the features. But since the TPR, TNR and MCC scores are not very good, it is uncertain if the goal is achievable with this method.

The final experiment, where the classifier is trained and validated with different hills, predicted all data to belong to TN. Hence, the TNR was 100% and TPR is 0%. The features from these hill are identically extracted, but as mentioned, the Vandavandet race

segment might not be as accurate as the Tinghaug segment. Another possibility is the fact that the hill at Vandavandet are not as steep as Tinghaug, and therefore not as hard. A third possibility is that Vandavandet is earlier in the race, which is an important factor on how tired the contestant are. To test if the classifier can be used to classify other hills, a more similar and more accurately placed hill might be needed to get any conclusive results.

6. Conclusion

Of the two goals for the work in this thesis, one achieved favourable results. For the first goal, to test if there were any information logged in the sports watches that could be connected to myocardial damage, the results indicated a clear connection. It was especially clear when using the *CT* class definition where all contestants had been diagnosed with CT-scan. Hence, their true medical condition were confirmed. To summarize, the best result in each of these experiment was the correct prediction of six of the seven who had myocardial damage. At the same time, up to 86% of the healthy contestant achieved a correct prediction.

As for the second goal, to find any relations between data from the sports watches and the contestants hsTnI values. Both approaches tested had less conclusive results. The results showed that when using the data from the sports watches, to predict whether the contestants were over or under a predefined hsTnI threshold. None of the tested feature sets were able to do this with satisfying results. When using the hsTnI values to predict whether the contestant were over or under the average race time, the results were somewhat better. All in all, it is uncertain if either of these methods will be able to achieve this goal.

Finally, for the last experiment, to test if it were possible to train the classifier with one hill and validate it with another. These results were inconclusive with current data and race segments.

6.1 Future work

As discussed, there are some issues with the current data set. For similar studies in the future, an idea is to provide the contestants with identical sports watches. The collected data from these will be more suitable, and might improve the results in this thesis. If lucky, one might even find relations between this new data and the hsTnI values.

If working further with current data, the velocity might need another method for smooth-

ing. As discussed, one solution might be to smooth the signals from the different contestants with different methods. Or one could simply just exclude the data sets where the velocity signals require too much smoothing. This will remove data from every class, which might not be an advantageous approach. Another improvement might be to design a method that increases the accuracy when defining the race segments.

Based on initial testing, the work in this thesis mainly used the *Naive Bayes* classifier in the experiments. It is possible that other classifiers might outperform it, or be more suitable for the second goal which remains unfulfilled.

Bibliography

- [1] Filtering and Smoothing Data. https://se.mathworks.com/help/curvefit/smoothing-data.html#bq_6ys3-3.
- [2] Helse Stavanger - NEEDED. <http://www.helse-stavanger.no/no/FagOgSamarbeid/ForskningOgUtviking/Sider/needed.aspx>.
- [3] Løypeprofil Nordsjørittet. <http://www.nordsjorittet.no/rittet/vis/loypekart>.
- [4] Myten om spreke nordmenn står for fall. <https://www.ssb.no/kultur-og-fritid/artikler-og-publikasjoner/myten-om-spreke-nordmenn-staar-for-fall>.
- [5] Naive bayes classifier. <https://se.mathworks.com/help/stats/templatenaivebayes.html>.
- [6] Norsk helseinformatikk - bmi. <http://nhi.no/forside/skjema-og-kalkulatorer/kalkulatorer/kmi-kroppsmasseindeks-3263.html>.
- [7] Om Nordsjørittet. <http://www.nordsjorittet.no/om-oss/>.
- [8] Siemens Healthcare - Cardiac troponin. <http://www.healthcare.siemens.com/clinical-specialities/cardiovascular/laboratory-expert/cvd-diagnostics-portfolio/portfolio-products/cardiac-troponin>.
- [9] Tidsskrift for Den norske legeforening nr. 22, 2011. <http://tidsskriftet.no/pdf/pdf2011/2226.pdf>.
- [10] AAKRE, K. M., RØRAAS, T., PETERSEN, P. H., SVARSTAD, E., SELLEVOLL, H., SKADBERG, Ø., SÆLE, K., AND SANDBERG, S. Weekly and 90-minute biological variations in cardiac troponin t and cardiac troponin i in hemodialysis patients and healthy controls. *Clinical chemistry* 60, 6 (2014), 838–847.
- [11] AGRESTI, A., AND COULL, B. A. Approximate is better than “exact” for interval estimation of binomial proportions. *The American Statistician* 52, 2 (05 1998), 119–126.

-
- [12] COLE, CHRISTOPHER R., M., BLACKSTONE, EUGENE H., M., PASHKOW, FREDRIC J., M., SNADER, CLAIRE E., M., AND LAUER, MICHAEL S., M. Heart-rate recovery immediately after exercise as a predictor of mortality. *The New England journal of medicine* 341, 18 (Oct 28 1999), 1351–1357.
- [13] DOMINGOS, P., AND PAZZANI, M. On the optimality of the simple bayesian classifier under zero-one loss. *Mach. Learn.* 29, 2-3 (Nov. 1997), 103–130.
- [14] F. N. FRITSCH, R. E. C. Monotone piecewise cubic interpolation. *SIAM Journal on Numerical Analysis* 17, 2 (1980), 238–246.
- [15] FARAHMAND, B., HÅLLMARKER, U., BROBERT, G. P., AND AHLBOM, A. Acute mortality during long-distance ski races (Vasaloppet). *Scandinavian Journal of Medicine & Science in Sports* 17, 4 (2007), 356–361.
- [16] HAND, D. J., AND YU, K. Idiot’s bayes—not so stupid after all? *International Statistical Review* 69, 3 (2001), 385–398.
- [17] HASTIE, T., TIBSHIRANI, R., AND FRIEDMAN, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics)*, 2nd ed. 2009. corr. 7th printing 2013 ed. Springer, 4 2011.
- [18] HEATH, M. T. *Scientific Computing: An Introductory Survey*, 2nd ed. McGraw-Hill Higher Education, 1996.
- [19] HINKLE, D. E., WIERSMA, W., AND JURIS, S. G. *Applied Statistics for the Behavioral Sciences*, 5 ed. Wadsworth Publishing, 10 2002.
- [20] KAHANER, D., MOLER, C., NASH, S., AND FORSYTHE, G. *Numerical methods and software*. Prentice-Hall series in computational mathematics. Prentice Hall, 1989.
- [21] KOHAVI, R., ET AL. A study of cross-validation and bootstrap for accuracy estimation and model selection. *International joint conference on artificial intelligence, 1995* 14, 2 (1995), 1137–1145.
- [22] MATTHEWS, B. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure* 405, 2 (1975), 442 – 445.
- [23] MOLER, C. B. *Numerical Computing with Matlab*. Society for Industrial and Applied Mathematics, 2004, ch. 3. Interpolation, pp. 93–116.
- [24] NES, B. M., JANSZKY, I., WISLØFF, U., STØYLEN, A., AND KARLSEN, T. Age-predicted maximal heart rate in healthy subjects: The hunt fitness study. *Scandinavian Journal of Medicine & Science in Sports* 23, 6 (2013), 697–704.

-
- [25] NYGÅRD, M. A. An analysis of pulse watches within the needed project. Master's thesis, University of Stavanger, 2015.
- [26] POWERS, D. M. W. Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation. *Journal of Machine Learning Technologies* 2, 1 (2011), 37 – 63.
- [27] SHAVE, R., BAGGISH, A., GEORGE, K., WOOD, M., SCHARHAG, J., WHYTE, G., GAZE, D., AND THOMPSON, P. D. Exercise-induced cardiac troponin elevationevidence, mechanisms, and implications. *Journal of the American College of Cardiology* 56, 3 (2010), 169–176.
- [28] TANAKA, H., MONAHAN, K. D., AND SEALS, D. R. Age-predicted maximal heart rate revisited. *Journal of the American College of Cardiology* 37, 1 (2001), 153 – 156.
- [29] THEODORIDIS, S., AND KOUTROUMBAS, K. *Pattern Recognition, Fourth Edition*, 4 ed. Academic Press, 11 2008.
- [30] TRUDE HÅLAND, A. L. *Nordsjørittet. Opplevelser. Mestring. Fellesskap*. Jæren Forlag, 2014.
- [31] VASAN, R. S. Cardiac function and obesity. *Heart* 89, 10 (10 2003), 1127.

A. Matlab-code

The following MATLAB-files are embedded as *matlab.7z*

restructureData.m

This script changes the original data set *filer.mat* from *cell* to *struct*-array. Furthermore, the script adds some additional information to the data set. This process is described more in detail in section 3.1.

Input: < *filer1.mat* > < *1502061_v2.xlsx* > < *151102mellomtider.xlsx* >

Output: < *Data.mat* >

resapleData.m

In this script there is some preprocessing of *Data.mat* before the whole data set is interpolated. See section 3.2 for further details.

Input: < *Data.mat* >

Output: < *Data2.mat* >

resampleTest.m

This script plots the original data and the interpolated data for visual inspection. Example in figure C.2.

Input: < *Data2.mat* >

smoothData.m

As described in section 3.3, this script is used to smooth the velocity signal.

Input: < *Data2.mat* >

Output: < *Data4.mat* >

findPOI.m

Finds the time and distance for the official split times in the contenders data. This script also calls the function *findDistance.m*

Input: < *Data4.mat* >

Output: < *POI.mat* >

findDistance.m

Function that returns the sample index for all contestants corresponding to a desired distance.

Input: < *Data4.mat* > < *desireddistance* >

Output: < *sample indexes* >

testPOI.m

Plots the point found in *POI.mat* for all contestants either relative to distance or time. This is used for visual inspection and then manually adjustment of the points.

Input: < *timeOrDistance* > < *Data4.mat* > < *POI.mat* >

tinghaug.m

Script that extracts the features from heart rate and velocity signals from the Tinghaug race segment. Additionally there is a possibility to plot these two signals for all contestants. The global features are also found in this script.

Input: < *Data4.mat* > < *POI.mat* >

Output: < *features.mat* >

vandavatnet.m

This script extends *features.mat* with features from the race segment at Vandavatnet. There is also a possibility to plot the heart rate and velocity signal in this segment with this script.

Input: < *Data4.mat* > < *POI.mat* > < *features.mat* >

Output: < *featuresAll.mat* >

CTclasses.m

Script that creates data set and features set for the CT class definition.

Input: < *Data4.mat* > < *CT.mat* > < *featuresAll.mat* >

Output: < *DataCT.mat* > < *featuresCT.mat* >

RaceTimeClasses.m

Function that decides the class indicator in data set *DataRT.mat* based on a chosen threshold.

Input: < *Data4.mat* > < *Threshold* >

Output: < *DataRT.mat* >

TroponinClasses.m

Function that decides the class indicator in data set *DataTnI.mat* based on a chosen threshold.

Input: < *Data4.mat* > < *Threshold* >

Output: < *DataTnI.mat* >

featuresDensity.m

Script that plots the PDFs for the original class definition. Either for two or three classes.

Input: < *Data4.mat* > < *featuresAll.mat* >

featuresDensityCT.m

Script that plots the PDFs for the CT class definition. It also has the option for a histogram plot over the PDFs.

Input: < *DataCT.mat* > < *featuresCT.mat* >

featuresDensityTnI.m

Script that plots the PDFs for the troponin class definition.

Input: < *DataTnI.mat* > < *featuresAll.mat* >

featuresDensityRT.m

Script that plots the PDFs for the race time class definition. It also has the option for a histogram plot over the PDFs.

Input: < *DataRT.mat* > < *featuresAll.mat* >

ClassifyFcn.m

Function that use a chosen < *classifier* > to classify the sports watch data with a selected < *featureSet* > and < *classDefinition* >. It will return the information displayed in the tables from the results. If used without outputs, the function will plot the confusion matrix. A more detailed explanation on the inputs and outputs for this function can be found in the beginning of the code itself.

Input: < *classDef* > < *featureSet* > < *classifier* > < *cost* > < *Val* > < *confInt* >

Output: < *conf* > < *Acc* > < *TPR* > < *TNR* > < *MCC* > < *ID on the misclassified* >

ClassifyCtUX.m

Example script for *exhaustive feature selection* with CT class definition

Input: < *DataCT.mat* > < *featuresCT.mat* >

Output: < *classifyResults_CT.mat* >

ClassifyCtUX_BottomUp.m

Example script for *sequential forward feature selection* with CT class definition

Input: < *DataCT.mat* > < *featuresCT.mat* >

Output: < *classifyResult_CT_BottomUp.mat* >

B. Data sets

The following data sets are not available to the public. If needed, contact Prof. Trygve Eftestøl at University of Stavanger for more information.

filer.mat

Data file from Martin Nygårds master thesis.

Data.mat

Data file after its been restructured and with added split times, weight, height and BMI..

Data2.mat

Data file where the interpolated signals are included.

Data4.mat

Data file after smoothing of the velocity data. This is the main data file for the original classes.

DataTnI.mat

Data file for the troponin class definition.

DataRT.mat

Data file for the race time class definition.

DataCT.mat

Data file for the CT class definition.

CT.mat

Vector with the ID numbers for the control group with low hsTnI that was diagnosed with CT-scan.

POI.mat

This file contains time and distance to the detected split times from the sport watch data.

featuresAll.mat

All features from the two hills Tinghaug and Vandavatnet and all the global features are collected in the file.

featuresCT.mat

Contains the same features as the one above, but only for the contestants in the CT class definition.

1502061_v2.xlsx

Excel sheet from the NEEDED study with various information. Example can be seen in figure 3.2.

151102mellomtider.xlsx

Excel sheet from the NEEDED study with the official split times for all contestants.

C. Additional figures

Visual inspection of POIs

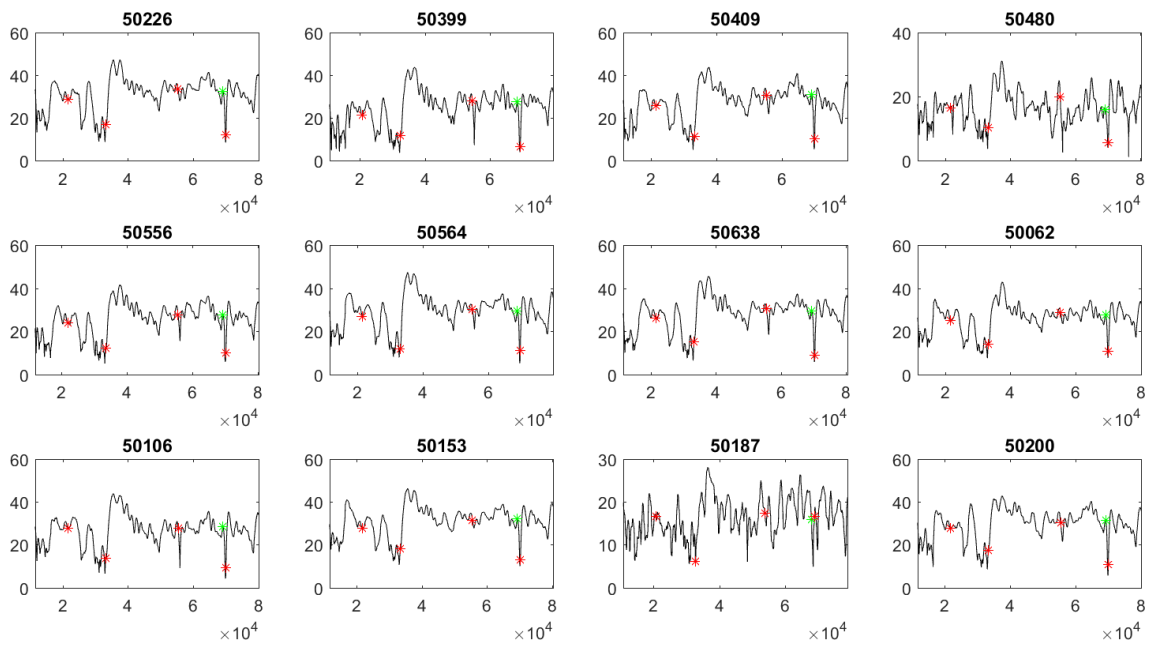


Figure C.1: Visually inspection of correct POI from a selection of the contestants

Visual inspection of interpolated singals

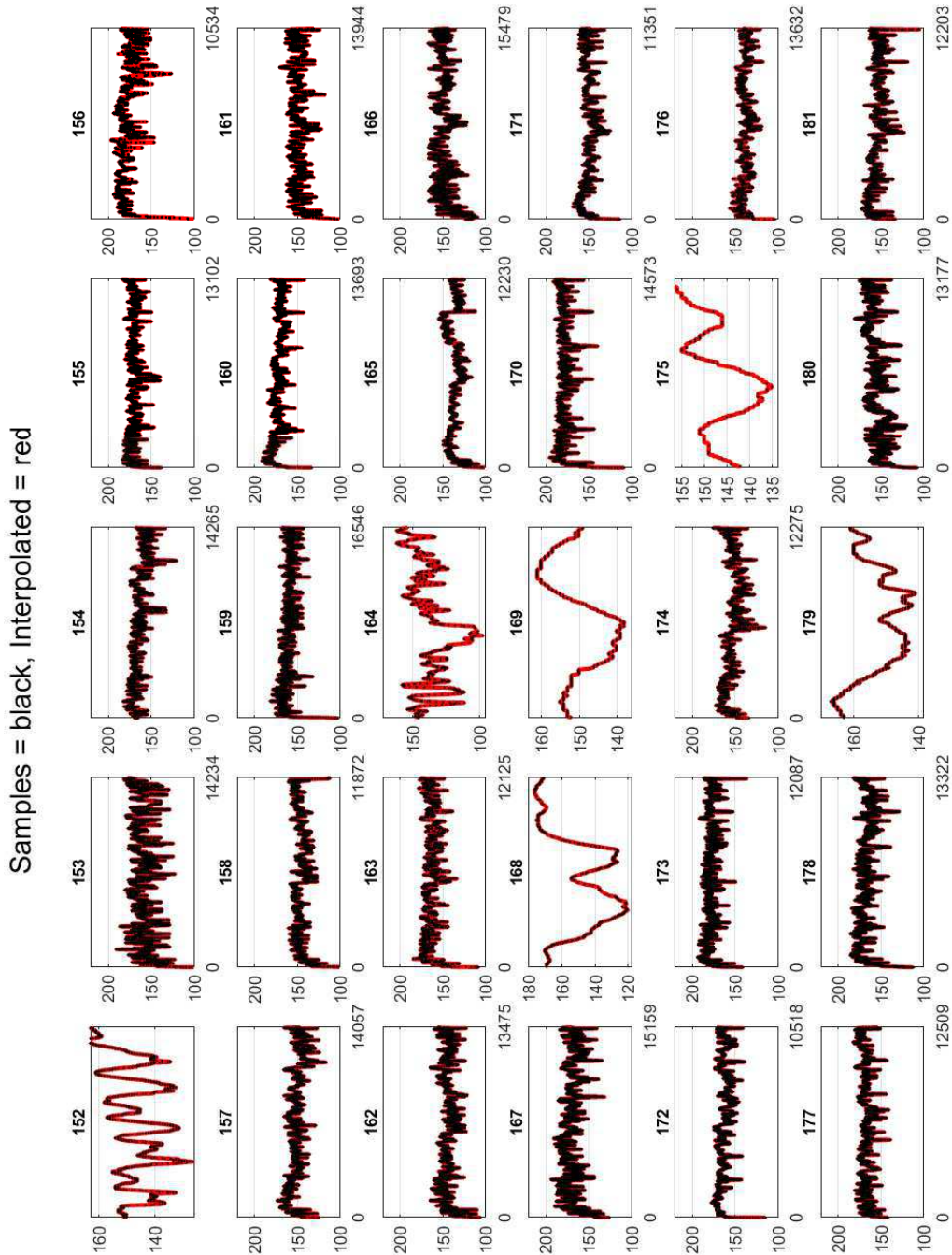


Figure C.2: Comparison of original heart rate signal versus interpolated signal from a selection of contestants. Some of the contestants in this plot are zoomed in for a more detailed inspection.

Original class definition

Predicted class	1	6 3.5%	31 18.2%	16.2% 83.8%
	2	1 0.6%	132 77.6%	99.2% 0.8%
		85.7% 14.3%	81.0% 19.0%	81.2% 18.8%
		1	2	
		True class		

(a) Exhaustive feature selection, Cost = 10

Predicted class	1	6 3.5%	34 20.0%	15.0% 85.0%
	2	1 0.6%	129 75.9%	99.2% 0.8%
		85.7% 14.3%	79.1% 20.9%	79.4% 20.6%
		1	2	
		True class		

(b) Sequential forward feature selection, Cost = 15

Figure C.3: Confusion matrices with the best results from the original class definition.

CT class definition

Predicted class	1	6 12.2%	6 12.2%	50.0% 50.0%
	2	1 2.0%	36 73.5%	97.3% 2.7%
		85.7% 14.3%	85.7% 14.3%	85.7% 14.3%
		1	2	
		True class		

(a) Exhaustive feature selection

Predicted class	1	6 12.2%	7 14.3%	46.2% 53.8%
	2	1 2.0%	35 71.4%	97.2% 2.8%
		85.7% 14.3%	83.3% 16.7%	83.7% 16.3%
		1	2	
		True class		

(b) Sequential forward feature selection

Figure C.4: Confusion matrices with the best results from CT class definition. Both with misclassification cost equal 5.

Troponin class definition

Predicted class	1	20 12.3%	46 28.2%	30.3% 69.7%
	2	12 7.4%	85 52.1%	87.6% 12.4%
		62.5% 37.5%	64.9% 35.1%	64.4% 35.6%
		1	2	
		True class		

(a) Exhaustive feature selection, Threshold = mean hsTnI, Cost = 4

Predicted class	1	21 12.9%	44 27.0%	32.3% 67.7%
	2	14 8.6%	84 51.5%	85.7% 14.3%
		60.0% 40.0%	65.6% 34.4%	64.4% 35.6%
		1	2	
		True class		

(b) Sequential forward feature selection, Threshold = 100, Cost = 4

Predicted class	1	18 11.0%	54 33.1%	25.0% 75.0%
	2	8 4.9%	83 50.9%	91.2% 8.8%
		69.2% 30.8%	60.6% 39.4%	62.0% 38.0%
		1	2	
		True class		

(c) Sequential forward feature selection, Threshold = 150, Cost = 5

Predicted class	1	9 5.5%	37 22.7%	19.6% 80.4%
	2	5 3.1%	112 68.7%	95.7% 4.3%
		64.3% 35.7%	75.2% 24.8%	74.2% 25.8%
		1	2	
		True class		

(d) Sequential forward feature selection, Threshold = 200, Cost = 4

Figure C.5: Confusion matrices with the best results from the hsTnI class definition.

Race time class definition

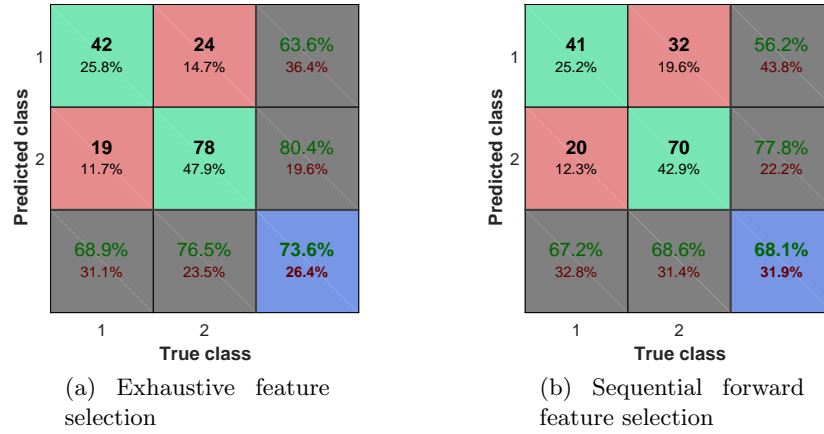


Figure C.6: Confusion matrices with the best results derived from the race time class definition. Both with misclassification cost equal 2.

D. Additional tables

Original class definition

	SVM ¹		NB ²		DT ³	
ACC	0.79	[0.72 0.84]	0.79	[0.73 0.85]	0.87	[0.81 0.91]
TPR	0.57	[0.25 0.84]	0.86	[0.47 0.99]	0.29	[0.08 0.65]
TNR	0.80	[0.73 0.85]	0.79	[0.72 0.85]	0.90	[0.84 0.93]
MCC	0.18		0.30		0.11	
Features	<i>Age</i>	<i>Force</i>	<i>CorrHrSpeed</i>	<i>BMI</i>		
	<i>maxHR</i>	<i>meanHR</i>	<i>medianSpeed</i>			

Table D.1: Comparison of different classifiers. Sequential forward features selection and misclassification cost = 15

CT class definition

	SVM		NB		DT	
ACC	0.69	[0.55 0.81]	0.84	[0.71 0.92]	0.80	[0.60 0.84]
TPR	0.57	[0.25 0.84]	0.86	[0.47 0.99]	0.86	[0.16 0.75]
TNR	0.71	[0.56 0.83]	0.83	[0.69 0.92]	0.79	[0.64 0.89]
MCC	0.21		0.55		0.17	
Features	<i>Age</i>	<i>Force</i>	<i>CorrHrSpeed</i>	<i>BMI</i>		
	<i>RaceTime</i>	<i>meanHR</i>	<i>maxHR</i>			

Table D.2: Comparison of different classifiers. Sequential forward features selection and misclassification cost = 5

¹Support Vector Machine

²Naive Bayes

³Decision tree

Troponin class definition

	SVM		NB		DT	
ACC	0.51	[0.51 0.66]	0.64	[0.57 0.71]	0.59	[0.51 0.66]
TPR	0.47	[0.31 0.64]	0.63	[0.45 0.77]	0.19	[0.09 0.36]
TNR	0.61	[0.53 0.69]	0.65	[0.56 0.73]	0.69	[0.60 0.76]
MCC	0.06		0.22		-0.11	
Features	<i>Age</i>	<i>Force</i>	<i>HRtwoMinAfterTop</i>	<i>DistTwoMinAfterTop</i>		

Table D.3: Comparison of different classifiers. Exhaustive search features selection and misclassification cost = 4

Race time class definition

	SVM		NB		DT	
ACC	0.65	[0.57 0.72]	0.68	[0.61 0.75]	0.50	[0.42 0.57]
TPR	0.69	[0.56 0.79]	0.67	[0.55 0.78]	0.39	[0.28 0.52]
TNR	0.63	[0.53 0.72]	0.69	[0.59 0.77]	0.59	[0.46 0.65]
MCC	0.31		0.35		0.-0.05	
Features	<i>Age</i>	<i>BMI</i>	<i>meanHR</i>	<i>hsTnI2</i>		
	<i>maxHR</i>	<i>diffHr</i>	<i>medianHR</i>			

Table D.4: Comparison of different classifiers. Sequential forward features selection and misclassification cost = 2