Can the 2MCA isomers improve vitamin B12 deficiency determination? a case for fuzzy modelling

Harrewijn, J.H.

Award date:
2017

Technische Universiteit Eindhoven University of Technology

Department of Industrial Engineering & Innovation Science

Information Systems Group

Can the 2MCA isomers improve vitamin B12 deficiency determination?

A case for fuzzy modelling

By:
J.H. (Jonan) Harrewijn
Student number: 0814621

Supervisors:

dr. A.M. (Anna) Wilbik	*TU/e IE&IS Information Systems Group*
prof. dr. ir. U. (Uzay) Kaymak	*TU/e IE&IS Information Systems Group*
dr. M. (Murat) Firat	*TU/e IE&IS Information Systems Group*
dr. A. (Arjen-Kars) Boer	*Catharina Hospital*

in partial fulfilment of the requirement for the degree of:

Master of Science in Operations, Management and Logistics

Eindhoven, Monday 20th November, 2017

TUE. School of Industrial Engineering
Series Master Thesis Operations Management and Logistics

Subject headings: Fuzzy inference systems, Vitamin B12 deficiency, 2-Methyl-Citric-Acid

Abstract

This report focusses on the improvement of determination of vitamin B12 deficiency. Vitamin B12 deficiency is prevalent in western countries, mainly in the elderly and the vegetarian populations. The determination is difficult since clinical symptoms become visible in a late stage of the disease. Biological indicators are often inconclusive or just plain out wrong. The most promising biological indicator Methyl Malonic Acid (MMA) is not just affected by vitamin B12 deficiency but also by renal function.

This study builds further on a previous fuzzy inference system which used plasma B12 and kidney function CKD-EPI as inputs to determine MMA levels. With that model it was possible to negate the effect of kidney function on MMA levels, though the performance left to be desired. In this study the two isomers of 2-Methyl-Citric acid (2MCA) and their ratio are explored as additional inputs to the model. Furthermore additional artificial intelligence methods are used with the goal of further performance improvement.

Though the relationships that the 2MCA isomers have with plasma B12, MMA and kidney function as described in literature were visible, it did not add any information which was not described by a model with vitamin B12 and CKD-EPI. This was visible in the fact that in the end none of the models with any form of 2-Methyl-Citric acid added were an improvement to the model with just plasma vitamin B12 and CKD-EPI as inputs.

The exploration into different artificial intelligence techniques did result in the creation and derivation of a weighted Gustafson-Kessel clustering function.

Executive summary

Vitamin B12 deficiency is still a common disease in the western countries. The main groups suffering from it are vegetarians, elderly and people who underwent bariatric surgery. About 20% of the elderly population has the disease. The effects being anemia and neurologic dysfunctions such as memory loss. These effects show late and damage to the neurological system can be irreversible up to a point. This strengthens the call for the search of biological markers of vitamin B12 deficiency, so that early detection can take place.

All of the biological markers have their own respective flaws. Using plasma vitamin B12 levels can lead to missing up to 50% of the deficient patients, while Methyl-Malonic Acid (MMA) would mark 45% of non-deficient patients as deficient. MMA is the most prominent marker, since it needs B12 to be processed in the body and therefore is directly related. It gives a good indication of cellular B12 levels, but it has the disadvantage of being influenced by kidney function. If a patients kidney function is worse, MMA will rise, just as if the patient is B12 deficient.

An earlier fuzzy inference model used plasma B12 and kidney function CKD-EPI as input to determine MMA levels of patients. The goal of the model was to be able to determine the normal MMA levels of a patient as if it had a good kidney function. Thereby discerning between a high MMA as a result of B12 deficiency and/or as a result of bad kidney function. The performance of the developed model still left a lot to be desired.

This rapport tries to improve that model performance by adding another biomarker called 2-Methyl-Citric Acid. Literature showed that this biomarker and its isomers are related to MMA and therefore to B12 deficiency. B12 deficient patients have higher 2MCA levels. It also is related to renal failure, being higher when people had worse performing kidneys. The individual isomers I and II, as well as their ratio I/II, were less well researched.

After the data was cleaned, the same relationships as described in the literature were seen. One important bias became visible, which is the fact that more measurements were performed when people had a B12 lower than 300. Initially no difference was visible between 2MCA1, 2MCA2 and 2MCAR, which resulted in the decision to compare all of the different variations when generating models which can be made by adding the 2MCA isomers.

Table 1: All of the different model inputs

Model name	**Model input**	**Model output**
Model 1	B12, CKD-EPI, 2MCA1, 2MCA2, 2MCAR	MMA
Model 2	B12, CKD-EPI, 2MCA1	MMA
Model 3	B12, CKD-EPI, 2MCA2	MMA
Model 4	B12, CKD-EPI, 2MCAR	MMA
Model 5	B12, CKD-EPI, 2MCA1, 2MCA2	MMA
Model 6	B12, CKD-EPI, 2MCA1, 2MCAR	MMA
Model 7	B12, CKD-EPI, 2MCA2, 2MCAR	MMA
Benchmark	B12, CKD-EPI	MMA

Model generation and results

The first model generated was a first order Takagi-Sugeno model generated by fuzzy c-means clustering. Errors were obtained by performing 10 times 10-fold cross validation. The benchmark model performed the best out of all of the models, so adding 2MCA in any way did not improve the models ability to predict MMA based on its inputs. The scatterplot of the errors show two problems which occur in all of the models and in all variations of model generation.

- Overestimating low MMA values and underestimating high MMA values.
- Having an output between 100 and 500.

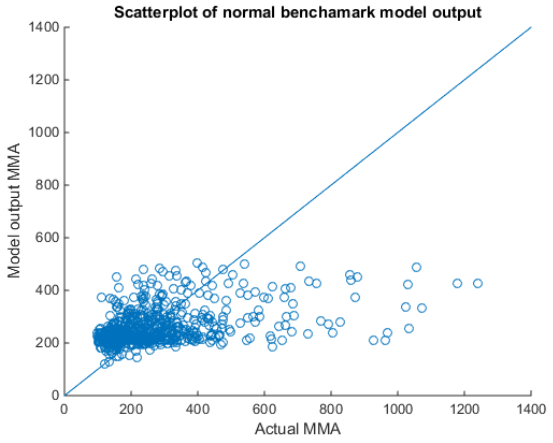

Figure 1: Scatterplot of errors, FCM first order Takagi-Sugeno model with 4 clusters, benchmark model

Changing from fuzzy c-means to Gustafson-Kessel clustering, or subtractive clustering did only slightly improve the output range to 600 for a handful of points. Different ways of partitioning the data in order to manually force the rules to spread out were also unsuccessful in solving the problems, while having a worse performance.

Table 2: Error values of the best two models for fuzzy c-means clustering

	Added input	**MAPE**	**MAE**	**MSE**	**R-squared**
Model 4	2MCAR	0,3667	95,34	21393	0,2177
Benchmark	NONE	0,3654	95,19	21371	0,2185

Balancing the data or adding weights to it did not manage to create model that stopped overestimating low MMA and underestimate high MMA. Though it did manage to get the output increased slightly to 800, but only for a handful of points. In most cases balancing and adding weights would impact the low MMA values significantly more than high MMA values. Causing high error rates, without fixing any of the aforementioned problems.

Adding other biomarkers to the FCM generated models did not increase performance either. Just as using neural networks and bootstrap aggregated decision trees did not significantly impact the outcome of the models.

Since none of the models with a variation of 2MCA isomers managed to improve compared to the benchmark models, it is clear that 2MCA does not add additional information to determine the elevation of MMA due to renal failure. Therefore it does not have any added value in determining B12 deficiency.

During the final stage of the project, it came to light that during the data collection of the 2MCA isomers their scale was changed. The data thus included 2MCA measurements with two different scales. Due to time limitations only the best performing model (fuzzy c-means) was redone with the data of the biggest set of the two.

The models which had 2MCA1 or 2MCA2 in their input showed an improved performance. The models with just 2MCAR added and the benchmark were unaffected. Though the performance of models with 2MCA1 and 2MCA2 improved, they still were not as good as the benchmark model, or the 2MCAR model and therefore insignificant to the final conclusion of the report.

Preface

This thesis is the result of six months of dedication and research, which marks my end as a student. In those six months a lot of people gave their support to me and this project, for which I want to take some space to thank them.

First off I want to thank Christ for giving me His daily blessings, knowledge and grace to sustain me. It is the foundation of everything I do and as such this report would not have been possible without Him.

I want to thank my university supervisors Anna and Uzay for their support and knowledge. Our meetings provided me with insights and your questions and remarks were crucial in bringing this thesis to a higher level. It was an honour getting the opportunity to learn from the years of experience you both have in this field. An experience I will take with me going further in life.

I also want to thank the company supervisor Arjen-Kars for giving me the data and the opportunity to use my talents to improve the real world. It gave me motivation knowing that my work might have an impact on the lives of people, which would not have been possible without your cooperation.

Next I would like to thank my parents and friends who always gave me mental support. An extra thank you goes to my friends from university who supported me when I was working, giving coffee, making jokes and giving me a light-hearted work environment. Further thanks go to my new friends in Eindhoven, who made me feel at home nearly instantly after moving to this city for my thesis.

Lastly I want to say that writing and making this thesis has been an incredible experience. I will take with me the experiences in research, artificial intelligence, but also the mistakes I made in every job I will have from now on.

I wish you much pleasure in reading this thesis,

Jonan Harrewijn

20th of November 2017

Table of Contents

1 Introduction 13

1.1 Research goals 13

1.2 Thesis structure 14

2 Background 15

2.1 B12 deficiency and its biomarkers 15

2.2 Methyl Malonic Acid and renal failure 15

2.2.1 Methyl Citric Acid 15

2.2.2 Additional biomarkers to add 16

2.3 Artificial Intelligence methods 16

2.4 Fuzzy Inference systems 17

2.4.1 Fuzzy partitions 17

2.4.2 Fuzzy inference systems 17

2.4.3 Selecting the right amount of clusters 20

2.4.4 Fuzzy C-means clustering 21

2.4.5 Gustafson Kessel 21

2.4.6 Subtractive clustering 22

2.4.7 Partitioning 23

2.4.8 Balancing and adapting the data 24

2.4.9 Global versus Local least squares minimization 25

2.5 Neural Networks 26

2.5.1 Neuron 26

2.5.2 Multilayer perceptron 26

2.5.3 Training the models 27

2.5.4 Disadvantage of neural networks 27

2.5.5 Fitnet 27

2.5.6 Cascadeforwardnet 28

2.6 Bootstrap aggregated decision trees 28

2.6.1 Sampling with replacement 28

2.6.2 Generate a tree from this sample 29

2.7 Methods for model evaluation 30

3 Methodology 31

3.1 Experimental setup 31

3.2 Data changes 31

3.3 Creating new values 32

3.4 Data cleaning 33

3.4.1 Removing wrong values 33
3.5 Getting to know the data 34
3.5.1 Histograms 34
3.5.2 Scatterplots 36
3.6 Artificial intelligence methods 37
3.7 Fuzzy Inference systems 37
3.7.1 Fuzzy c-means clustering 37
3.7.2 Gustafson Kessel clustering 37
3.7.3 Subtractive clustering 38
3.7.4 Partitioning 38
3.7.5 Adapting and balancing the data 40
3.7.6 Adding biomarkers 43
3.8 Neural networks 44
3.8.1 Fitnet 44
3.8.2 Cascadeforwardnet 44
3.8.3 Balanced Fitnet and Cascadeforwardnet 44
3.9 Bootstrap aggregated decision trees 44
3.9.1 With normal data 44
3.9.2 With balanced data 44
4 Results 45
4.1 Fuzzy inference systems 45
4.1.1 Individual relationships between 2MCA and the other data vectors and cluster selection 45
4.1.2 Fuzzy c-means clustering 47
4.1.3 Gustafson Kessel 48
4.1.4 Subtractive clustering 48
4.1.5 Partitioning 49
4.1.6 Balancing and adapting the data 52
4.1.7 Adding biomarkers 56
4.2 Neural Networks 59
4.2.1 Fitnet 59
4.2.2 Cascadeforwardnet 59
4.2.3 Balancing 60
4.3 Bootstrap aggregated decision trees 61
4.3.1 Tree size 61
4.3.2 Balancing 62

5 Conclusion 63
6 Bibliography 66
7 Appendix 69
7.1 Error values of models 69
7.2 Histograms of errors 79
7.3 Scatterplot of errors 89
7.4 Fuzzy c-means model with correct data 92

List of figures

Figure 1: Scatterplot of errors, FCM first order Takagi-Sugeno model with 4 clusters, benchmark model 5
Figure 2: Example of fuzzy sets of age (Agrawal, 2014) 17
Figure 3: Fuzzy inputs used to determine rule output in an Mamdani type fuzzy inference system (International Birch University, 2012) 18
Figure 4: Two outputs of a Mamdani fis system being combined to make an output distribution (Princeton, 2007) 19
Figure 5: Representation of a neuron as proposed by McCulloch and Pitts (Jain et al., 1996) 26
Figure 6: Visualization of a model with neurons sorted into layers (University of Nevada Reno, 2003) 27
Figure 7: A visualization of a neural network model with a tansig hidden layer and a linear output layer (Mathworks, 2017) 28
Figure 8: CascadeForward Neural Network architecture (Badde, Gupta, & Patki, 2009) 28
Figure 9: Histogram of 2MCA1 occurrences Figure 10: Histogram of 2MCA1 occurrences after removal 33
Figure 11: Histogram of 2MCAR before removal 34
Figure 12: Histogram of MMA occurrences 35
Figure 13: Histogram of 2MCA1 occurrences 35
Figure 14: Histogram of B12 occurrences 35
Figure 15: Histogram of CKD-EPI occurrences 36
Figure 16: Scatterplots of the data 36
Figure 17: Visualization of the membership functions generated by genfis1 39
Figure 18: The histogram of CKD-EPI compared to the scatterplot of the weights 40
Figure 19: The histogram of B12 compared to the scatterplot of the weights 40
Figure 20: Scatterplot of the errors of the benchmark model, generated by fuzzy c-means clustering 47
Figure 21: Histogram of the errors of the benchmark model, generated by fuzzy c-means clustering 47
Figure 22: Scatterplot of the errors of the benchmark model, generated by Gustafson-Kessel clustering 48
Figure 23: Scatterplot of the errors of model 4, generated by subtractive clustering 48
Figure 24: Scatterplot of the errors of the benchmark model, generated by subtractive clustering ... 48
Figure 25: Scatterplot of the errors of model 4, generated by maximum size partitioning 49
Figure 26: Scatterplot of the errors of model 4, generated by maximum size partitioning 50
Figure 27: Scatterplot of the errors of the benchmark model, generated by partitioning based on the biomarkers 51

Figure 28: Scatterplot of the errors of the benchmark model versus model 4, generated by equal distance partitioning 51
Figure 29: Scatterplot of the errors of the benchmark mode, generated by equal distance partitioning 52
Figure 30: Membership degrees of fcm versus weighted fcm 52
Figure 31: Membership functions of vitamin B12 from the benchmark model, non-weighted GK versus weighted GK 53
Figure 32: Scatterplot of the errors of the benchmark model, generated by weighted GK clustering 53
Figure 33: Scatterplot of the errors of model 4, generated by balanced subtractive clustering in training 54
Figure 34: Histogram of errors, balanced training subtractive clustering, benchmark model 54
Figure 35: Scatterplot of the errors of model 4, generated by balanced subtractive clustering in training and cluster selection 55
Figure 36: Scatterplot of errors of the benchmark model with age, generated by fuzzy c-means clustering 56
Figure 37: Scatterplot of errors of the benchmark model with Hcy, generated by fuzzy c-means clustering 57
Figure 38: Scatterplot of errors of model 4 with folate, generated by fuzzy c-means clustering 57
Figure 39: Scatterplot of errors of model 4 with all biomarkers, generated by fuzzy c-means clustering 58
Figure 40: Scatterplot of the errors of model 4, generated by 4 layered fitnet 59
Figure 41: Scatterplot of errors of model 1, generated by balanced fitnet 60
Figure 42: Histogram of errors, 4 layered fitnet, model 1 60
Figure 43: Scatterplot of errors of model 1, generated by bootstrap aggregated decision trees with 100 trees 61
Figure 44: Scatterplot of errors of model 1, generated by balanced bootstrap aggregated decision trees with 100 trees 62
Figure 45: Scatterplot of 2MCA1 versus MMA in which the relationship between them is visible..... 63
Figure 46: Scatterplot of errors, FCM first order Takagi-Sugeno model with 4 clusters, benchmark model 63
Figure 47: Histogram of errors, GK, Benchmark model 79
Figure 48: Histogram of errors, Subtractive clustering, Benchmark model 79
Figure 49: Histogram of errors, Maximum size partition without MMA, Model 4 80
Figure 50: Histogram of errors, Maximum size partitioning with MMA, Model 4 80
Figure 51: Histogram of errors, Biomarker partition, benchmark model 81
Figure 52: Histogram of errors, equal distance partition 3, Benchmark model 81
Figure 53: Histogram of errors, equal distance partition 5, Benchmark model 82
Figure 54: Histogram of erros, weighted fuzzy c-means, benchmark model 82
Figure 55: Histogram of errors, weighted GK, Benchmark model 83
Figure 56: Histogram of errors, totally balanced subtractive clustering, benchmark model 83
Figure 57: Histogram of errors, 0-Order Takagi-Sugeno weighted training, Benchmark model 84
Figure 58: Histogram of errors, FCM with age added, benchmark model 84
Figure 59: Histogram of errors, FCM with Hcy added, benchmark model 85
Figure 60: Histogram of errors, FCM with folate added, model 4 85
Figure 61: Histogram of errors, FCM with haemoglobin added, benchmark model 86
Figure 62: Histogram of errors, 0-order Takagi-Sugeno, Benchmark model 86
Figure 63: Histogram of errors, 4 layered fitnet, model 4 87
Figure 64: Histogram of errors, 4 layered cascadeforwardnet, model 6 87

Figure 65: Histogram of errors, bootstrap aggregated decision trees, 100 trees 88
Figure 66: Histogram of errors, bootrstrap aggregated decision trees, 1000 trees 88
Figure 67: Histogram of errors, bootstrap aggregated decision trees, 100 balanced trees 89
Figure 68: Scatterplot of errors, weighted fuzzy c-means, benchmark model 89
Figure 69: Scatterplot of errors of benchmark model, fuzzy c-means with haemoglobin 90
Figure 70: Scatterplot of errors of model 4, generated by 4 layered cascadeforwardnet 90
Figure 71: Scatterplot of errors, bootstrap aggregated decision trees, 1000 trees, model 1 91
Figure 72: Scatterplot of 2MCA1 measurements per date 92

List of tables

Table 1: All of the different model inputs 4
Table 2: Error values of the best two models for fuzzy c-means clustering 5
Table 3: Table showing how data is partitioned with 2 inputs 23
Table 4: Table showing the different models generated by selecting inputs 31
Table 5: Table showing the initial parameters for Gustafson Kessel 38
Table 6: Table showing the changed parameters of Gustafson Kessel 38
Table 7: Table showing the different partition of values based on human interpretation 39
Table 8: Table showing the initial and changed values of the parameters of weighted Gustafson Kessel 41
Table 9: Mean absolute percentage errors with 2MCA values as output with the amount of clusters in brackets 45
Table 10: Mean absolute percentage errors with 2MCA values as input with the amount of clusters in brackets 45
Table 11: Pearson cluster correlation validity index with the amount of clusters in brackets 46
Table 12: Spearman cluster correlation validity with the amount of clusters in brackets 46
Table 13: Error values for fuzzy c-means clustering 47
Table 14: Cluster centres of the best performing benchmark model 47
Table 15: Error values for Gustafson Kessel clustering 48
Table 16: Error values for subtractive clustering 48
Table 17: Cluster centres of model 4 49
Table 18: Cluster centres of the benchmark model 49
Table 19: Error values for maximum size partitioning without MMA 49
Table 20: Error values of maximum size partitioning with MMA 50
Table 21: Error values of partitions based on the biomarkers 51
Table 22: Error values of equal distance partitioning in 3 parts 51
Table 23: Error values of equal distance partitioning in 5 parts 52
Table 24: Error values of weighted fcm 52
Table 25: Error values of weighted Gustafson Kessel 53
Table 26: Error values of balanced subtractive clustering for training 54
Table 27: Error values of balanced subtractive clustering 55
Table 28: Consequent parameters of non-weighted and weighted equal distance partitioning 55
Table 29: Error values of equal distance partitioning with weight adapted training 55
Table 30: Error values of fuzzy c-means clustering with age added 56
Table 31: Error values of fuzzy c-means clustering with Homocysteine added 57
Table 32: Error values of fuzzy c-means clustering with folate added 57
Table 33: Error values of fuzzy c-means clustering with haemoglobin added 58
Table 34: Error values of fuzzy c-means clustering with all 4 additional biomarkers added 58

Table 35: Error values of the best performing fitnet neural networks 59
Table 36: Error values of 100 bagged decision trees 61
Table 37: Error values of 1000 bagged decision trees 61
Table 38: Error values of 100 balanced bagged decision trees 62
Table 39: Best error values of the models with corrected data 92

1 Introduction

1.1 Research goals

Vitamin B12 can cause severe and irreversible damage to the human body (Van Der Put, Van Straaten, Trijbels, & Blom, 2001). It cannot be synthesized by the human body itself and therefore we are completely dependent on B12 intake via diet. Furthermore it is only available in animal products, which make vegans likely to develop a deficiency when not taking vitamin B12 supplements. Often elderly people too have a problem of getting sufficient B12 intake via their diet and absorption. Due to these factors it is common and necessary to test for vitamin B12 deficiency either after bariatric surgery or when people show symptoms.

The biomarkers of vitamin B12 deficiency are ambiguous. Plasma B12 for example has shown to not be reflective of cellular B12 values (Solomon, 2005). Though MMA manages to detect all deficient people, it will often also falsely nominate about 50% of people without a deficiency (Palacios et al., 2013; Schwarz et al., 2015). Hannibal has made an up to date summary of the currently used biomarkers and their problems (Hannibal et al., 2016).

The goal of the research is to improve the classification of patients with a vitamin B12 deficiency. Nowadays in the Catharina Hospital in Eindhoven measures plasma vitamin B12 and MMA to determine whether or not patients have a vitamin B12 deficiency. MMA is elevated in patients with B12 deficiency, but can also be elevated due to renal failure. It has already been established that models which take into account CKD-EPI values for renal failure improve the accuracy. In this research we try to improve the accuracy even more, by using 2MCA as well.

2MCA can be measured in two isomers called 2MCA-I and 2MCA-II. When the relationship between 2MCA and something else is researched, we check the individual isomers and the ratio of isomer I divided by isomer II.

> Can 2MCA be used to improve the diagnostic capabilities of MMA in detecting B12 deficiency?
>
> a. Which relationship does 2MCA have to renal failure?
> b. Which relationship does 2MCA have to MMA?
> c. Which relationship does 2MCA have to vitamin B12 plasma?
> d. Can 2MCA determine the elevation of MMA values due to renal failure?

As can be viewed in the research question, the first part of the research consists of establishing if 2MCA has any relationship with renal failure, MMA and vitamin B12 plasma. That information can be used to determine which of the isomers to use in building a model. After having established these relationships, the work on generating a model that uses 2MCA, Renal failure and B12 levels to determine accurate MMA levels can begin.

Methodology

Generating a model will follow the Crisp-DM method. Which consists of the following phases:

Business understanding

Understanding the environment of the problem, the underlying causes and the scope. This is meant as a preparation for the actual work itself, making sure as much as possible that a solution will fit within the current situation and that it will fulfil all requirements.

Data understanding

Understanding the data means gathering knowledge about the way of data collection. It is important to understand who contributed to the data and who did not. This better understanding can help when drawing conclusions from it, or might be the cause for collecting additional or different data.

Data preparation

The preparation of the data consists of making the data suitable for its intended use. It can mean manually inserting the data into a system, removing wrong values or combining or changing values into required formats.

Modelling

Modelling is about making and generating a model with the data that has been prepared. Of course the model should be made with the scope and the environment in mind.

Evaluation

Evaluation is not just testing the model on its performance. It is mainly evaluating if the model solves the originally stated problems and whether or not it falls inside the scope and is a good fit for the place it needs to perform in.

Deployment

Deployment is about integrating the solution into the company or other place it needs to be deployed in. An important task is to see if the model keeps its performance and if it is being used in the intended manner. Furthermore the effects of the model should be evaluated and checked if they are intended and desired.

1.2 Thesis structure

Chapter 2 explains the background information of 2MCA, vitamin B12 deficiency and the artificial intelligence used. In chapter 3 the methods used to generate the models are described, which includes a derivation of optimization parameters for weighted Gustafson-Kessel. The results of the models are presented and evaluated in chapter 4. Chapter 5 gives a conclusion of the research results. It also shows the limitations of the research, directions for further research and recommendations to the Catharina Hospital.

2 Background

2.1 B12 deficiency and its biomarkers

Vitamin B12 deficiency is common among adults. In developed countries, about 20% of the elderly population has a B12 deficiency (Andrès et al., 2004). Mainly vegetarians and elderly people have a deficiency due to their nutritional habits. Vitamin B12 also called Cobalamin cannot be synthesized by the human body itself, so humans need to get it from animal products. A clinical diagnosis is difficult, since the clinical symptoms are subtle and appear late. Furthermore, when those symptoms appear, not all of the damage done can be reversed by treatment. These factors fuel the search for biological markers of vitamin B12 deficiency in an attempt to diagnose the deficiency in an early stage. An up to date summarization of the currently used biomarkers and their limitations was made by Hannibal (Hannibal et al., 2016).

All of the biomarkers have their own respective limitations and therefore more research has started to appear in an attempt to combine biomarkers in different ways to improve diagnostic capabilities. A common way is using multiple biomarkers during diagnosis, but that does not negate their individual limitations, since the markers can disagree with each other. Fedosov was one of the first that started combining the biomarkers into one parameter (Fedosov, 2010, 2013). It was successfully applied in diagnosing elderly patients (Risch et al., 2015). The parameter is defined as follows:

$$\mathrm{cB12} = \log10((\mathrm{holoTC} * \mathrm{B12})/(\mathrm{MMA} * \mathrm{tHcy})) - (\mathrm{age\ factor}) \quad (1)$$

Though this parameter is an improvement to the previous if -> then structures, it still relies on linear structures and was found to be inaccurate in patients with renal insufficiency.

2.2 Methyl Malonic Acid and renal failure

One of the commonly used biomarkers MMA has the limitation that it is not only affected by B12 deficiencies, but also by renal failure. It can nearly always identify B12 deficient patients, but has a lot of false positives (Palacios et al., 2013; Schwarz et al., 2015). It marked about 45% of non-deficient patients as deficient. In order to increase the diagnostic value of MMA and reduce its limitations an attempt was made at generating artificial intelligence models that could accurately compensate MMA values for renal deficiency (Wilbik, Loon, Boer, & Kaymak, 2016). This was done by generating a model which had MMA as output and both plasma B12 and CKD-EPI (an indicator of renal failure) as inputs. In this report an attempt is made to further improve the model by adding the isomers 2MCA1, 2MCA2 and the ratio of them 2MCA1/2MCA2 as inputs.

2.2.1 Methyl Citric Acid

2 Methyl citric acid is related to cobalamin deficiency via the methylmalonyl-CoA mutase. This is related to the biomarker MMA. It was found to be significantly increased in cobalamin deficient patients (Allen, Stabler, Savage, & Lindenbaum, 1993). Besides that 2MCA was lowered more in patients who got folate, vitamin B6 and vitamin B12 supplements, compared to placebos (Naurath et al., 1995).

Other research confirmed the connection between renal failure and 2MCA concentrations. Both in patients which received treatment for renal failure (Busch et al., 2004; Henning et al., 1999) as well in patients after a renal transplant (Stein, Muller, Busch, Fleck, & Sperschneider, 2001). More importantly, in patients with chronic renal failure, vitamin supplements failed to lower the elevated 2MCA (Henning, Tepel, Graefe, & Zidek, 2000).

Very little research has been done on the isomers of 2MCA. Allen showed in 1993 that the ratio of the isomers was related to renal failure, though with a sample of only 15 persons (Allen et al., 1993).

Later on the ratio was shown to have an inverse relationship to serum folate in renal patients (Stein et al., 2001). It has never been used as a biomarker for B12 deficiency.

2.2.2 Additional biomarkers to add

Besides 2MCA there were additional biomarkers available in the dataset. The additional biomarkers were added in an explorative fashion to see whether or not they would have any effect on the effects of adding 2MCA. The biomarkers are not chosen randomly though and have an already established connection to vitamin B12 deficiency.

Age

The first biomarker that is added is age. Even though age is a parameter in the kidney function, it might still be useful to have as a separate variable. B12 was found to be age sensitive by Fedosov (Fedosov, 2013) and later on his combined parameter improved its performance when age was added to help discern for renal failure (Risch et al., 2015).

Homocysteine

Homocysteine (Hcy) was used by Fedosov (Fedosov, 2010; Risch et al., 2015) in his combined parameter, but also by a lot of other people when looking for biological markers to determine a B12 deficiency (Langan & Zawistoski, 2011; Schwarz et al., 2015). Therefore it might be of good use helping to determine MMA levels.

Folate

Folate is often used to discern between a folate deficiency and a B12 deficiency (Palacios et al., 2013). Though MMA is not impacted by a folate deficiency, it could still give additional information about the health of a person.

Haemoglobin

Haemoglobin was used by Fedosov as a way to see the severity of a B12 deficiency (Fedosov, 2013). It showed the effects of a vitamin B12 deficiency and was used to search for biomarkers. Plasma B12 was found to be able to differentiate between normal and low haemoglobin values. While MMA could not.

2.3 Artificial Intelligence methods

In order to achieve the goals of this project, not just one artificial intelligence method was used, but multiple different methods were used to compare models and their performances. Furthermore to combat bias in data different techniques were used in cluster selection and training.

Three main modelling techniques were used:

- Fuzzy Inference Systems (FIS)
- Neural Networks
- Bootstrap aggregated decision trees

In the next sections the techniques which were used are explained. With the exception of weighted Gustafson-Kessel, which will be explained and derived in the methodology chapter.

2.4 Fuzzy Inference systems

2.4.1 Fuzzy partitions

Fuzzy inference systems are based on rules consisting of fuzzy partitions. The idea comes from the fact that sometimes crisp segregation into sets is not always applicable or good. In the case of gender it is certainly possible to split it into two sets, but when talking about age, any split would not be fair to the type of the data.

Fuzziness in this case means that an age can be part of multiple sets. If we divide age up into three different fuzzy sets called young, middle aged and old, then any age can belong to one, two or depending on the membership functions three categories. How much a point is part of a fuzzy set is called degree of membership. This degree of memberhip is calculated by a membership function.

There are different types of membership functions. Two general types can be discerned between functions who reach an absolute zero and functions that do not.

Each of the functions has it respective advantages and disadvantages. Some datasets will perform better with certain functions, though there is no strict set of rules regarding it.

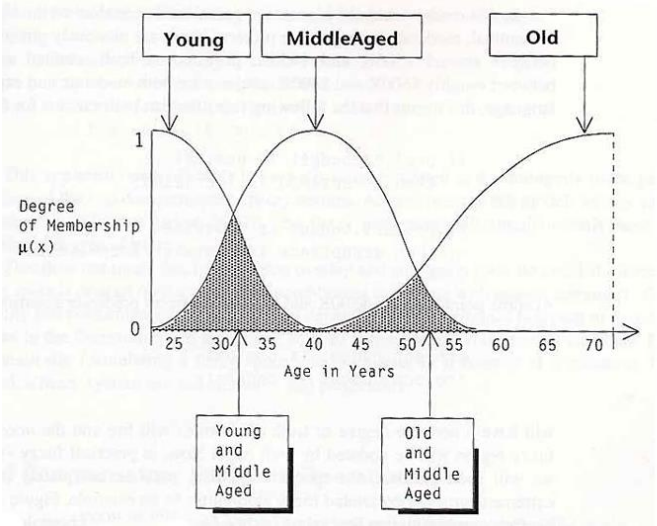

Figure 2: Example of fuzzy sets of age (Agrawal, 2014)

2.4.2 Fuzzy inference systems

Fuzzy inference systems make models out of those fuzzy sets by generating rules. A system generally consists of the following steps, though different types of fuzzy systems may have different steps (Princeton, 2007):

1. determining a set of fuzzy rules
2. fuzzifying the inputs using the input membership functions,
3. combining the fuzzified inputs according to the fuzzy rules to establish a rule strength,
4. finding the consequence of the rule by combining the rule strength and the output membership function,
5. combining the consequences to get an output distribution, and
6. defuzzifying the output distribution (this step is only if a crisp output (class) is needed).

Determining a set of fuzzy rules

The first step is integral in making a fuzzy inference model. Determining a set of fuzzy rules has a big impact on the way a model functions. A common method is fuzzy c-means clustering. Which is a method that determines cluster centres and the fuzzy memberships of the points of a dataset. It will be explained with more detail later on. Other methods of determining a set of fuzzy rules exist and will be used. Each of the clustering techniques has its own respective advantages and disadvantages.

Fuzzifying the inputs using the input membership functions

When a model is made, the first thing that happens in determining the outcome, is fuzzifying the input. This means that the input is put into the membership functions and the degrees of membership are established. These degrees of membership function as inputs for the rules of the fuzzy inference model. An input of age 32 in figure 1 might for example have the degrees of membership:

$$\mu_x(young) = 0.4, \qquad \mu_x(middle\ aged) = 0.6, \qquad \mu_x(old) = 0$$

Combining the fuzzy inputs according to fuzzy rules to establish a rule strength

A rule of a fuzzy inference model consist of one degree of membership of one or multiple input variables. The rules follow an IF- THEN structure, with the IF part being called the antecedent and the THEN part being called the consequent. The IF part gets added together to generate a rule strength which is used in the next step. The rule strength can be calculated in different ways and these calculations are called T-norms. There are a lot of different T-norms, ranging from the simple minimum (called 'and') and maximum (called 'or') to more complicated products.

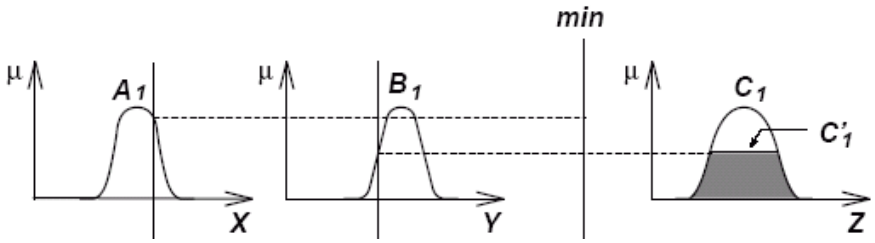

Figure 3: Fuzzy inputs used to determine rule output in an Mamdani type fuzzy inference system (International Birch University, 2012)

Finding the consequence of the rule by combining the rule strength and the output membership function

This is where the difference between Mamdani and Takagi-Sugeno systems come in. The consequence part of a Mamdani system is made with fuzzy membership functions called output membership functions. In Takagi-Sugeno the consequence part consists of linear formulas, or in case of a 0-order model of just a constant.

$$Z_1 = p_1 x + q_1 y + r_1 \tag{2}$$

The rule strength is called degree of applicability in Takagi-Sugeno systems and is used in the last step when determining the output of a model.

In Mamdani systems the right part of figure 2 is executed, the rule strength is used to determine the consequence of the rule in the output membership function.

Combining the consequences to get an output distribution

This is only applicable in Mamdani systems. The output membership functions are combined in a way to make an output distribution. The combining of functions can be done in multiple ways again, just like in the antecedents. Minimum, maximum and more complicated methods are used. This output is still in a fuzzy form and needs to be defuzzified before something meaningful can be derived from it.

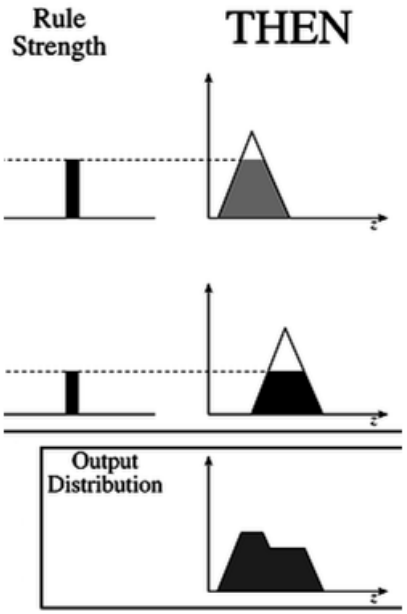

Figure 4: Two outputs of a Mamdani fis system being combined to make an output distribution (Princeton, 2007)

Defuzzifying the output distribution (this step is only if a crisp output (class) is needed).

Mamdani fuzzy inference systems

Mamdani fuzzy inference models can be defuzzified in a multitude of different ways (Princeton, 2007). For example the centre of mass:

$$z = \frac{\sum_{j=1}^{q} Z_j u_c(Z_j)}{\sum_{j=1}^{q} u_c(Z_j)}, \tag{3}$$

where: z is the centre of mass and u_c is the membership in class c at value Z_j.

Another option to use is the mean of the maximum:

$$z = \sum_{j=1}^{l} \frac{z_j}{l}, \tag{4}$$

where: z is the mean of the maximum, z_j is the point at which the membership function is maximum, and l is the number of times the output distribution reaches the maximum level.

Takagi-Sugeno fuzzy inference systems

Takagi-Sugeno fuzzy inference systems have outputs that consist of functions or just of constants. These functions are usually combined by weighted average to determine the output of the model. Weighted average uses the degree of applicability -sometimes just called weight- determined in step 3 and the rule outputs determined in step 4.

$$x^* = \frac{\sum_{r=1}^{R} w_r x_r}{\sum_{r=1}^{R} w_r}, \tag{5}$$

where x^* is the model output, R is the amount of rules, w_r is the degree of applicability of rule r and x_r is the output of rule r.

2.4.3 Selecting the right amount of clusters

To start fuzzy c-means clustering or Gustafson Kessel clustering you need to determine the amount of clusters prior to the algorithm. To see if there is an optimal amount of clusters you can use iVAT to generate an image in which clusters might become visible (Bezdek & Hathaway, 2016; Havens & Bezdek, 2012). Though this method requires an exponential amount of computing power when the amount of data is increased. For a large dataset like this one iVAT or a normal VAT image is not possible to make.

After having computed clusters we want to compare them to each other in order to select the best performing ones, but since the cost function cannot be compared between different data points and a different amount of clusters we need some other way of evaluation. There are two correlation cluster validity indices, Pearson and Spearman, which allow for cluster comparisons. Since both indices are better on different types of data, it is best to check both of the indices when deciding on the amount of clusters to use.

Pearson correlation cluster validity index

This index compares two dissimilarity matrices. The first dissimilarity matrix is the matrix of the initial data. The second one is the matrix of the clustered data. When the data was partitioned well the two matrices should be similar to each other. Pearsons test assumes that the data is normally distributed (Popescu et al., 2013; Popescu, Keller, Bezdek, & Havens, 2011).

$$V_{ccvp}\big(D, D(U)\big) = \frac{\langle A,B\rangle_2}{(\|A\|_2\|B\|_2)}, \tag{6}$$

where

$$a_{ij} = D_{ij} - \overline{D}_{ij} \tag{7}$$

$$b_{ij} = D_{ij}(U) - \overline{D}_{ij}(U) \tag{8}$$

Pearsons ccv is 1 when the matrices are linearly correlated in the same direction and -1 if the matrices are linearly correlated in the opposite direction.

Spearman correlation cluster validity index

This index uses the same formula as the Pearson index, but instead of the values themselves the comparison is done with the ranks of the values. The monotonic relationship is more general than the linear relationship being checked by the Pearson index (Popescu et al., 2013, 2011).

$$V_{ccvs}\big(D, D(U)\big) = \frac{\langle r,r^*\rangle_2}{(\|r\|_2\|r^*\|_2)} \tag{9}$$

Spearmans ccv checks if there is a relationship between the ranks of the two matrices. It returns 1 when the matrices are in monotonic agreement and -1 when the monotonic orders are reversed.

Errors of the model

A different way to compare the amount of clusters, is by generating a fuzzy inference model by clustering the data and evaluating it. This is the easiest way of selecting the right amount of clusters. It also gives shows directly how well the selected clusters are made into a fuzzy inference system.

2.4.4 Fuzzy C-means clustering

With fuzzy c-means clustering, each data point belongs to a certain degree to each cluster. Though, in some cases a point can virtually belong for 100% to one cluster, this is never the case for all of the data points when using fuzzy c-means clustering. The sum of all the degrees of membership of a single data point to the clusters always needs to be 1.

$$\sum_{i=1}^{c} u_{ij} = 1, \forall\, j = 1, \dots, n \tag{10}$$

The objective function for FCM is to minimize:

$$J(U, c_1, \dots, c_c) = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \sum_{j}^{n} u_{ij}^{m}\, d_{ij}^{2}\,, \tag{11}$$

where $0 < u_{ij} < 1$; c_i is a cluster centre of a fuzzy group I; $d_{ij} = \|c_i - x_j\|$; and $m \in [1, \infty)$ which is a weighting exponent (Jang, Sun, & Mizutani, 1997).

To find the necessary conditions for the minimization of the objective function, you can make a new objective function and differentiate it to get the following two conditions:

$$c_i = \frac{\sum_{j=1}^{n} u_{ij}^{m} x_j}{\sum_{j=1}^{n} u_{ij}^{m}} \tag{12}$$

and

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}}\right)^{2/(m-1)}} \tag{13}$$

The FCM algorithm

Now we have the necessary formula to make the FCM algorithm:

Step 1: Randomly initialize the membership functions of the data points, or initialize the cluster centres.
Step 2: Calculate the fuzzy cluster centres with the first of the necessary conditions
Step 3: Calculate the objective function (cost function). Stop if a pre requisite condition has been met.
Step 4: Compute the new u_{ij} values using the second necessary conditions and go back to step 2.

2.4.5 Gustafson Kessel

The GK algorithm is an extension of fuzzy c-means clustering. It allows to find different cluster shapes. It does this by allowing the clusters to adapt the distance norm to the local topological structure of the data (Balasko, Abonyi, & Feil, 2005). The algorithm described and used here is the normal Gustafson-Kessel (Gustafson & Kessel, 1979) algorithm and not the modified one (Babuška, Veen, & Kaymak, 2002; Balasko et al., 2005; Oliveira & Pedrycz, 2007).

Given the dataset Z, choose the standard parameters $M, \alpha, \varepsilon, p_i$ and the condition number threshold β. Initialize the partition matrix and compute the covariance matrix F_0 of the whole dataset.

In the functions w_{ij} is the membership of sample i of cluster j, x_i is a sample, M is a symmetric and positive-definite, α is a smoothing parameter and p_i is a volume constraint.

Repeat for l = 1,2,…

Step 1: Compute the prototypes (means)

$$v_j^{*} = \frac{\sum_{i=1}^{N} w_{ij}^{\alpha} x_i}{\sum_{i=1}^{N} w_{ij}^{\alpha}} \tag{14}$$

Step 2: Compute the cluster covariance matrices

$$P_{fj} = \frac{\sum_{i=1}^{N} w_{ij}^{\alpha}(x_i - m_{fj})\,(x_i - m_{fj})^T}{\sum_{i=1}^{N} w_{ij}^{\alpha}} \; ; \; \alpha > 1 \tag{15}$$

Step 3: Reconstruct the distances

$$d_{ij}(\theta_j) = (x_i - v_j)^T M_j (x_i - v_j), 1 \le j \le k \tag{16}$$

Where

$$M_j^{*-1} = \left(\frac{1}{\rho_j |P_{fj}|}\right)^{1/n} P_{fj} \tag{17}$$

Step 4: Update the partition matrix

For $1 \le i \le N$, if $d_{ij} > 0 \; for \; 1 \le j \le K$,

$$w_{ij}^{*} = \frac{1}{\sum_{l=1}^{j} (d_{ij}/d_{il})^{1/(\alpha-1)}} \tag{18}$$

Otherwise

$$w_{ij}^{*} = 0 \; \boldsymbol{if} \; d_{ij} > 0 \; \boldsymbol{and} \; w_{ij}^{*} \in [0,1] \, , \tag{19}$$

$$\textbf{else} \sum_{i=1}^{N} w_{ij} = 1$$

$$\textbf{Until} \left\| W^{l} - W^{(l-1)} \right\| < \varepsilon \tag{20}$$

2.4.6 Subtractive clustering

Fuzzy c-means clustering chooses the starting points of clusters through pure randomness. These random starting points have a big impact on the final results of the clusters. To circumvent this random selection we will use a method that selects cluster centres based on the density of points (Chiu, 1994).

Subtractive clustering selects data points themselves as cluster centres.

Step 1: Calculate the potential value of each of the points.

$$P_i = \sum_{j=1}^{n} e^{-\alpha \left\| x_i - x_j \right\|^2} \qquad \text{where } \alpha = \frac{4}{r_a^2} \tag{21}$$

Step 2: Select the point with the highest potential and take that point as the first cluster centre.
Step 3: Revise the potential of all of the data points with the following formula:

$$P_i = P_i - P_k^{*} * e^{-\beta \left\| x_i - x_j \right\|^2} \qquad \text{where } \beta = \frac{4}{r_b^2} \tag{22}$$

The value $R_b$is a constant that determines the effect of a chosen point on the suitability of the other points. When R_b is bigger, it will lower the potential of other points more. It was advised that R_b takes on the value of $R_b = 1.5 * R_a$.

Step 4: Set the potential of the selected data point to 0.
Step 5: Select the next highest potential.
Step 6: Check if the new data point should be accepted. We need two values for this:

$$\bar{\varepsilon} = The \; upper \; boundrary$$

$$\underline{\varepsilon} = The \; lower \; boundrary$$

- If the newly selected point $P_k^* > \overline{\varepsilon} * P_1^*$ then the point should be accepted as a cluster centre and the steps should be repeated from step 3 onwards.
- If the newly selected point $P_k^* < \underline{\varepsilon} * P_1^*$ then the data point should be rejected and the clustering process should be stopped.
- If the newly selected point is between the upper and lower limits another check should take place. This check determines if the distance of the new cluster centre is far enough away from existing selected cluster centres, but also compensates for the potential of the new cluster centre compared to the potential of the first cluster centre. So, if the potential low, but the distance to the existing cluster centres is high the cluster will be selected. The reverse is possible too. The formula is: $\frac{d_{min}}{R_a} + \frac{P_k^*}{P_1^*} \geq 1$ where d_{min} is the shortest distance of the selected data point to the existing cluster centres.
    - If the formula is true the data point should be accepted as a cluster centre and the steps should be repeated from step 3 onwards.
    - If the formula is not true the data point should be rejected as a cluster centre and given a potential of 0, then the steps should be repeated from step 5 onward.

It is clear that this method of cluster selection has no set amount of clusters to select. It will differ based on the values of R_a, R_b, $\overline{\varepsilon}$ and $\underline{\varepsilon}$. For all of these a lower value will mean a higher amount of clusters selected.

To change these cluster centres into a fuzzy inference system we consider each of the cluster centres to be a fuzzy rule. The degree of fulfilment of rule *i* by a point is defined as the formula:

$$\mu_{i=} e^{-\alpha \|y - y_i^*\|^2} \tag{23}$$

The formula for the output is a weighted average, multiplication is used for the AND operator.

2.4.7 Partitioning

Partitioning is a way of selecting clusters while giving the density of points less of an importance. The data gets partitioned in each of its input vectors. Each combination of partitions gets its own rule in the fuzzy inference system. The way the partitions are made, and the requirements for a partition to become a rule can differ in some ways.

This means that with X amount of input variables and a partitioning of Y parts, the total amount of subsets in the dataset will be:

$$Amount\ of\ subsets = Y^X \tag{24}$$

In case of two inputs being partitioned in 2 ways the four combinations are:

Table 3: Table showing how data is partitioned with 2 inputs

Partition combinations	Input 2	
Input 1	Low-Low	High-Low
	Low-High	High-High

2.4.7.1 Maximum size partitions

Partitioning without MMA

By partitioning the data into different parts and making membership functions for the parts that have enough data points we force the clusters to represent each partition separately. This way clusters are forced to be distributed and as such they are prevented from staying together in a biased area.

With maximum size partitioning the data is divided into equal lengths based on their maximum values. If the maximum value of any input is 1500 and we partition the data into two, the partition will be from 0-750 and 751-1500. Data is partitioned in all of the input variables.

Partitioning with MMA

By just partitioning the input values we are potentially missing out on vital information which is in the output value MMA. By also partitioning the data into the direction of the output values the clusters will better represent both high and low MMA.

2.4.7.2 Partition based on sources of the biomarkers

Using partitions purely based on maximum values might not reflect reality, since the distributions of values might not coincide with what is seen as low, normal or high values by human interpretation. By selecting the partitions ourselves the membership functions will make more biological sense and might select better clusters as a result.

2.4.7.3 Equal distance partitions

The partitions of the previous parts were self-made and gaussian combination functions were used. In equal distance partitioning, the Matlab function genfis1 is used. This function automatically partitions the data into a set partitions, so we do not need to set a minimum amount of points to filter out small partitions. All of the partitions therefore are made into a rule. The cluster centres are of equal distance removed from each other. The difference between the equal distance partitioning and the maximum size partitioning is in the location of the membership functions as well as the type of the membership functions.

2.4.8 Balancing and adapting the data

Why use different weights in model generation and evaluation?

When a model gets generated, there often is a point in which the model tries to improve its parameters, through different ways. Most commonly these parameter improvements are evaluated by error values, such as the MSE or MAE. When the new parameters have lowered the error values they are better than the previous ones. In some models it can be useful to increase or decrease the weight of certain data points. The effect of these increases and decreases are a bigger or lesser focus of the model generation on those respective data points when evaluating their parameters. This can also be done by copying data points which need more focus.

2.4.8.1 Fuzzy c-means clustering with weights

The fuzzy c-means clustering was adapted for weights by Kaymak (Kaymak, 2003). These weights came in two ways. One for clusters (τ_i) and one for the data points (σ_j). Since weights are used to combat bias in the data collection which is only part of the data points the cluster weights will not be used.

The new objective function to minimize is:

$$J_w = \sum_{i=1}^{C} \sum_{j=1}^{N} u_{ij}^{m} \sigma_j \tau_i d_{ij}^2 \quad (25)$$

The update equations then change to:

$$v_i = \frac{\sum_{j=1}^{N} u_{ij}^{m} \sigma_j x_j}{\sum_{j=1}^{N} u_{ij}^{m} \sigma_j} \tag{26}$$

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} (\frac{\tau_i d_{ij}^2}{\tau_k d_{kj}^2})^{\frac{1}{m-1}}} \tag{27}$$

2.4.8.2 Gustafson-Kessel clustering with weights

Weights can be used during cluster selection to increase the reward for the algorithm when points with high weights are closer to a cluster centre than compared to points with a low weight. Weights therefore can be used to focus the "attention" of the Gustafson-Kessel algorithm to points which are in a less dense area. Kaymak did this already for the fuzzy c-means algorithm and the adjustments will be similar (Kaymak, 2003; Kaymak & Van Berg, 2004).

In the FCM example, weights are added for both data points as well as clusters. But in this report we will not use cluster weights, since the weights are used to combat bias in the data points.

2.4.8.3 Balanced subtractive clustering for training

To force the models to put more 'focus' onto the higher MMA values, we wanted to rebalance the data. This means that you divide the data into subsets and copy the data points which need more focus. To be able to still compare the results from these model to the results of the other models we only balance the training data. The testing data remains unbalanced.

After selecting the clusters we need to decide on the data attribute and cut off point to balance the data. The output attribute was chosen and the cut-off point was 300. So the data was divided into two sets, one with MMA values below 300 and one with MMA values above 300. Then the smallest dataset, which is the one with values above 300 was copied fully as much as possible without getting bigger than the other dataset. The last data points to equal the sizes are chosen randomly.

By using this balanced data when training the consequent values will be trained with more high MMA values.

2.4.8.4 Balanced subtractive clustering for training and cluster selection

In order to select better clusters, the cluster selection was also done with the balanced data. This forces subtractive clustering to select clusters which better represent higher MMA values. This, when combined with the balanced training is expected to put even more focus onto high MMA values and therefore diminishing the negative aspects of bias in the data.

2.4.8.5 0 order Takagi-Sugeno equal distance partition with weight adapted training

When adding weights in the training of a 0-order Takagi-Sugeno model we can easily see the differences that the weights have on the consequent values. The antecedent values will remain the same for all of the models. By changing the consequent values the expectation is that the model will perform differently.

2.4.9 Global versus Local least squares minimization

Now some of the models generated are just models with set antecedent parameters, for which the consequent operators have been randomly determined and still can be improved with local and global least squares estimation. Global least squares estimation changes all of the parameters at the same time, while local estimation varies only one parameter in each step. The advantages of both over the other are (Ortigueira & Tribolet, 1984):

Local minimization over global minimization:

- It is easier to implement
- It computes the RC's in a sequential way, which gives some insights into the correct order and so it may avoid the problem of an ill-conditioned matrix.
- It allows to control of the stability of the prediction error filter

Global minimization over local minimization:

- Truly minimizes the error power
- Gives more reliable estimates

In this report we will therefore use global minimization.

2.5 Neural Networks

2.5.1 Neuron

Neural networks are networks of individual neurons connected to each other. These neurons are based off of biological networks. Though we now know that biological networks do not work in the same way, the inspiration can be seen. The building block of a neural network is a neuron. The representation of such a neuron as first proposed by McCulloch and Pitts is shown in figure 1. This neuron computes a weighted sum of its input signals. If that is above a certain threshold it will fire an output.

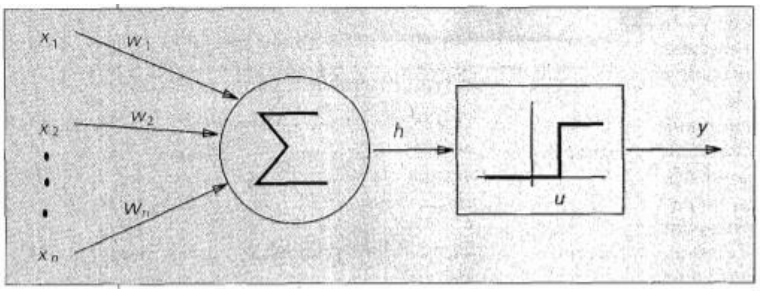

Figure 5: Representation of a neuron as proposed by McCulloch and Pitts (Jain et al., 1996)

Mathematically it can be represented as (Jain, Mao, & Mohiuddin, 1996):

$$y = \theta\left(\sum_{j=1}^{n} w_j x_j - u\right) \qquad (28)$$

Where θ is a unit step function if the result is positive. This is the most basic version of a neuron, and other activation functions can be used too. From different step-wise functions to more general piecewise linear or gaussian functions.

2.5.2 Multilayer perceptron

The most common way of organizing neurons in a model is by putting them into layers. This is called multilayer perceptron. In these types of neural networks the neurons are connected in a one directional way. All of the neurons from the input layer send their output signal to all of the neurons of the second layer. The neurons of the second layer then use the outputs of the first layer as input and send their output to the next layer again. Generally speaking these layers have 3 different names.

Input layer

The input layer is the first layer in the data and has the input of the model as input in the neurons. Though the layers can differ in method, usually each input has a separate neuron.

Output layer

The output layer is the last layer of a neural network. It's goal is to translate the state of the neural network into the respective output. In the case of pattern recognition it might use step functions for example to determine the difference between two classes. In the case of function approximation other functions will be used, that are able to produce continuous values.

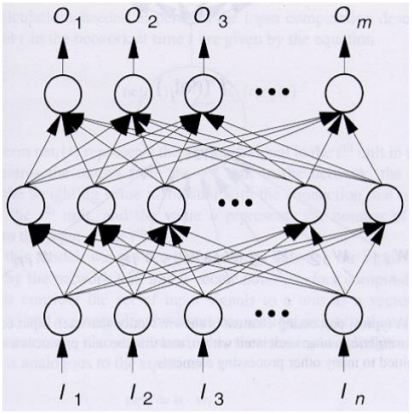

Figure 6: Visualization of a model with neurons sorted into layers (University of Nevada Reno, 2003)

Hidden layer

The hidden layers are the layers in-between the input and output layer. These layers do the actual computation of the models. More hidden layers means that the model will do more complicated calculations, but it is also more prone to overfitting. Usually all of the neurons in a layer have the same type of activation functions. Though when you have multiple hidden layers you can vary the activation function in each layer. Furthermore the amount of neurons can be varied for better or worse results.

2.5.3 Training the models

Training of a neural network model is difficult due to the complicated relationships between a neuron in a hidden layer, and the resulting output of the model. Backpropagation is used to train neural networks when supervised learning can be applied. Backpropagation consists of two stages (Byoung-Tak, 2001):

- Forward stage: Calculate outputs given input X
- Backward stage: Update weights by calculating delta

There is a lot of mathematics to the backward stage, but it is not in the scope of this report to explain and explore those in depth.

2.5.4 Disadvantage of neural networks

Neural networks can find the most complicated relationships, but the disadvantage is that it does not clearly disclose the relationships themselves. Even though visualization techniques for the activation of nodes have been improved, neural networks still remain black boxes. It is really good in predicting outputs, but not good in making apparent the relationships between input and output.

2.5.5 Fitnet

The function fitnet creates a standard network that tries to fit a function between the input and output data. The Levenberg-Marquardt function is used for training. Mean Squared Error is used as performance function during training.

The nodes in the output layer have purelin functions. These functions are used to generate continuous outputs like our MMA values. The hidden layers use tansig functions for which the ouput is between -1 and 1.

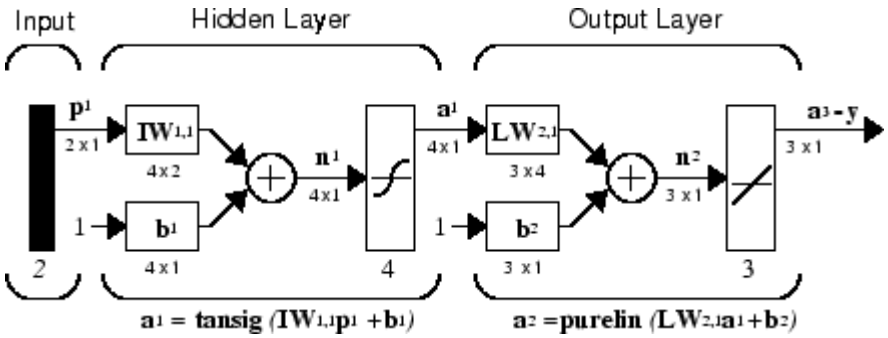

Figure 7: A visualization of a neural network model with a tansig hidden layer and a linear output layer (Mathworks, 2017)

2.5.6 Cascadeforwardnet

A cascadeforwardnet uses the same parameters as the fitnet function. The one difference is that it includes a connection from the input layer and every previous layer to all of the following layers. This way not just individual information is carried on from neuron to neuron, but also from layer to layer.

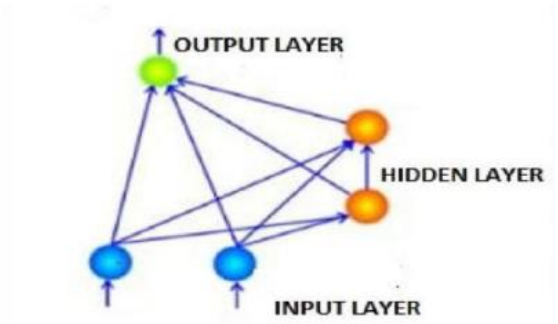

Figure 8: CascadeForward Neural Network architecture (Badde, Gupta, & Patki, 2009)

2.6 Bootstrap aggregated decision trees

The basis for bootstrap aggregation artificial intelligence are tree predictors. Tree predictors work with decision nodes. Usually a decision node consists of 2 options. After a decision has been made another node can be found with yet another decision. This continues for some time until the end of the tree has been reached. In classification the end of a tree will give a certain class. In the event of regression the end has an output value. The specialty of bootstrap aggregation is the combination of trees. Multiple trees are made in the following way:

Step 1: Sample with replacement until you have n samples
Step 2: Generate a tree from this sample
Step 3: Repeat step 1 and 2 until you have a predetermined amount of trees
Step 4: Combine the trees by averaging their results

2.6.1 Sampling with replacement

For bootstrap aggregation the standard way to sample with replacement is until you have just as much data points in the set as in the set you are sampling from. With replacement means that you do not remove the samples from the original set once you have sampled them. You keep sampling from

the same set. This means that the sample can contain a data point multiple times. On average 68.2% of the sample will be original points.

This sampling without replacement is important for bootstrap aggregation. Because of the different samples, different trees will be produced.

2.6.2 Generate a tree from this sample

Step 1: Split the data on the node on a variable

The first step is to generate multiple binary splits on one variable of the data which are to be compared to each other.

Step 2: Apply a goodness of split to each of the splits and evaluate the reduction in impurity

There are different algorithms for the selection of nodes. The one described by Yohannes and Hoddinott is the following (Yohannes & Hoddinott, 1999):

$$i(t) = \phi(p(1|t), p(2|t)) \tag{29}$$

Where $p(1|t)$ is the probability distribution of classification 1 at a node and $p(2|t)$ is the probability distribution of classification 2 at a given node. Where $\phi(\dots)$ denotes a measure of heterogeneity.

Step 3: Select the best split

After having evaluated the splits the best performing split is chosen as a node.

Step 4: Repeat step 1-3

Repeat the first three steps until for all of the variables at the root node. Then the algorithm ranks the best splits on each variable according to the reduction of impurity.

Step 5: Select the best performing variable

Select the best performing variable and split of that variable.

Step 6: Select node classes

After the split has happened the node classes are calculated. With regression these classes are just the values. These classes are the outcome of the tree if a datapoint followed it all the way through until that node. These classes are selected by minimizing an error value.

Step 7: Repeat steps 1-6

Repeat steps 1 to 6 until a predefined parameter has been reached. It can be that a maximum number of nodes have been reached, or the error value does not significantly decrease after the last split. When the algorithm is finished the tree is ready.

Keep generating new trees

Repeating the steps will result in a lot of trees. The amount of trees is predetermined by the user at the start of the algorithm. Increasing the amount of trees does not always increase the performance of the model, since all of the trees will be averaged, an average of 100 trees can have only marginal differences as an average of a 1000 trees.

Combine the trees by averaging their results

Each of the individual trees are overfitted to their own respective data. This is something we like to avoid and will be automatically done by averaging the outcome of all the trees (Breiman, 2001). A combination of the trees will therefore always perform better than the individual trees themselves.

2.7 Methods for model evaluation

Mean absolute percentage error

Mean Absolute Percentage Error. The MAPE is an error value which calculates the error as a percentage of the actual output. This way an absolute error of 20 will lead to a higher MAPE when the actual output was 100 compared to an actual output of 800. A MAPE of 0.40 means that on average the absolute value of the error is 40% of the value which was supposed to be predicted.

$$MAPE = \frac{\sum(\frac{|Error|}{Actual\ output})}{n} \tag{30}$$

Mean absolute error

Mean Absolute Error. The MAE is a simple error value which displays the average absolute size of the errors. It is an utmost simple value, but can only be used when comparing models with similar outputs. It gives an easily interpretable value, though it can be misleading since in general low values will have lower absolute errors, while high values will have higher absolute errors.

$$MAE = \frac{\sum|Error|}{n} \tag{31}$$

Mean squared error

Mean Squared Error. The MSE is similar to the MAE, but instead of taking the absolute of the errors, it squares the errors. This will punish high mistakes severely more than small faults. The square of an error of 100 is 10.000, while an error of 20 is only 400 when squared. Using this value together with the MAE can show important performance differences between models which have a similar MAE.

$$MSE = \frac{\sum Error^2}{n} \tag{32}$$

R-squared

R-squared is also called explained variance. It shows the amount of variance the model explains compared to a model with just the mean output value. A perfect model has a r-squared value of 1. Negative values can be get when models have a higher sum of squared errors than a model with just the mean output as output value would have.

$$R^2 = 1 - \frac{\sum {Error_i}^2}{\sum(y_i - \bar{y})^2} \tag{33}$$

Kolmogorov-Smirnov test

The Kolmogorov-Smirnov test is a test that can compare two distributions to each other by comparing their cumulative step functions. The comparison is done by calculating the maximum distance between the two functions. The null hypothesis is that the vectors are from the same continuous distribution. This is rejected if the percentage of points that are bigger than the maximum deviation set by the confidence level is allowed (Frank & Massey, 1951). This maximum deviation is called the critical value.

3 Methodology

In this chapter the methods for the stages of data understanding, data preparation and model generation are explained. As well as the experimental setup of all of the different models. Furthermore the derivation of weighted Gustafson Kessel is shown as well as the method to determine the weights.

3.1 Experimental setup

For each of the models the data is parted into a trainingset and a testset. The training data is used to generate a model. Which is then evaluated by the test data. This is repeated in a structured order of 10 times 10-fold cross validation. The errors of the models then are averaged to give an accurate representation of the method.

Since we have isomers of 2MCA, we can make multiple models. One of the models is called the benchmark model. This model uses the same inputs as in the previous study done on this subject (Wilbik et al., 2016). The rest of the models have an additional input of one or a combination of multiple 2MCA isomers. The models tested in each variation are as follows:

Table 4: Table showing the different models generated by selecting inputs

Model name	**Model input**	**Model output**
Model 1	B12, CKD-EPI, 2MCA1, 2MCA2, 2MCAR	MMA
Model 2	B12, CKD-EPI, 2MCA1	MMA
Model 3	B12, CKD-EPI, 2MCA2	MMA
Model 4	B12, CKD-EPI, 2MCAR	MMA
Model 5	B12, CKD-EPI, 2MCA1, 2MCA2	MMA
Model 6	B12, CKD-EPI, 2MCA1, 2MCAR	MMA
Model 7	B12, CKD-EPI, 2MCA2, 2MCAR	MMA
Benchmark	B12, CKD-EPI	MMA

All of the models were generated using the same data partitions, so that they were trained and tested on the same exact data. The methods of generating the models will be explained individually for each of the artificial intelligence methods.

3.2 Data changes

The first part of any data analysis project is changing the data from its raw format into a usable format for analysis. In this case the data was delivered in 3 separate Excel documents. One for the year 2014, one for 2015 and 2016, and one for the first three months of 2017. Before the Excel files were read into Matlab some changes were made, since they were done more easily in Excel than in Matlab.

- The age of the patients had a datatype which was not “Number”, but “String”, which means that it cannot be read into Matlab at the same time as the other variables, since Matlab only accepts one datatype in a matrix. The type was changed in Excel using the “number format” panel.
- Some of the B12 measurements were shown as >1475.999, since the measurement apparatus has that limit. Matlab treated this as “NaN” because of the “larger than” sign. Therefore all of the measurements were changed to 1475.999 with the help of “find and replace” in the editing panel. This had no impact on the project since all of these measurements will be filtered out in a later stage.

- The last change consisted of changing the gender to a numerical value. Gender was described with the letter M for Male and F for Female. Similar to the first change, strings cannot be imported in the same matrix as numbers, so gender was changed to numbers. The F for Female was changed into the number 1 and the M for Male was changed into the number 2.

After these changes the data could be easily loaded into Matlab, since all of the data was of the same datatype. The next changes are therefore done in Matlab instead of Excell.

- The time needed to be changed into a different format in order to check the time between measurements. The time was in the format "dd-mm-yyyy hh:mm". In this format checking the time in days between measurements is difficult to implement. Therefore the dates were changed into the standard Matlab format which displays in a single number the amount of days passed since 00/01/0000.

Most of the measurements did not measure all of the needed values required for this research at the same time. Most often, one of the values would be measured first and additional measurements were done at a later time. Therefore if patients had multiple measurements and if those measurements were done within 14 days of each other, the missing values would be filled in by the other measurements. Sometimes this would create measurements which were exactly similar to each other. Any double measurement was removed. This adding together of measurements was done under the assumption that the body would not change substantially inbetween the 14 days of the measurements.

3.3 Creating new values

For analysis we need to create two values which are not directly in the data. One is the CKD-EPI and another is the ratio of 2MCA1/2MCA2 which is called 2MCAR in this report.

CKD-EPI is a formula to calculate the estimated glomerular filtration rate (eGFR), which is a number indicating how well a kidney is performing (Levey et al., 2009). Lower values mean worse kidney performance. The CKD-EPI is calculated differently for males and females. Also a different formula is used depending on their creatine levels.

Males with serum creatine <=79.6 µmol/L

$$CKD - EPI = 141 * \left(\frac{Creatine}{79.6}\right)^{-0.411} * 0.993^{Age} \quad (34)$$

Male with serum creatine >79.6 µmol/L

$$CKD - EPI = 141 * \left(\frac{Creatine}{79.6}\right)^{-1.209} * 0.993^{Age} \quad (35)$$

Female with serum creatine <=61.9 µmol/L

$$CKD - EPI = 141 * \left(\frac{Creatine}{61.9}\right)^{-0.329} * 0.993^{Age} \quad (36)$$

Female with serum creatine >61.9 µmol/L

$$CKD - EPI = 141 * \left(\frac{Creatine}{61.9}\right)^{-1.209} * 0.993^{Age} \quad (37)$$

3.4 Data cleaning

First off we remove all measurements for which vitamin B12 is larger than 900. This is to prevent taking into account people who have received B12 supplements. Their values would distort the view of normal people who have not received a treatment yet, for which this model is intended.

Any measurement for which there are no values of 2MCA1 or 2MCA2 are removed, since those values are necessary.

There are also measurements in the dataset which do not have any patient numbers. These measurements were done at the Catharina Hospital in Eindhoven as a request by other hospitals or other third parties without the person being a patient of the Catharina Hospital. To prevent any unknown overrepresentation of certain patient groups those measurements are removed.

3.4.1 Removing wrong values

Wrong values can be caused by a mistake when inputting data into the database, or by mistake when doing the measurements. An easy example would be having a negative age or a negative amount of MMA. More difficult to spot are values which are positive, but are extremely high compared to the rest of the dataset. For age there are clear boundaries, but for other values the boundaries must be set by comparing values with the rest of the dataset. These outliers can be spotted by using histograms or other plots. Removing these outliers is vital since they will have a disproportionate effect on model generation. Furthermore they are biological extremities and not representative for the overall population.

Age

There are no negative values and the oldest patient is 102 years old. This does not seem unreasonable and therefore no changes were made.

Gender

No other genders than “Male” or “Female” were found.

Date/Time of measurement

All of the dates fit in the years the datasets take place in. Therefore no faults are assumed.

2MCA1

The histogram of 2MCA1 shows an extreme outlier of a measurement which is about 1200 times higher than the average value. The histogram keeps looking skewed, until we remove all of the 2MCA1 values higher than 40.000 nmol/L.

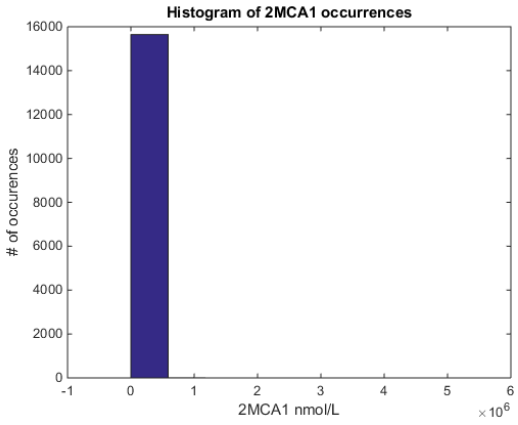

Figure 9: Histogram of 2MCA1 occurrences

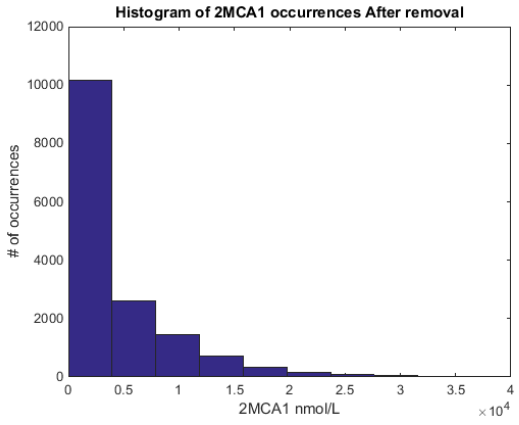

Figure 10: Histogram of 2MCA1 occurrences after removal

2MCA2

For 2MCA2 the maximum values aren't as extreme as with 2MCA1, but we can see the same skewedness in the histogram, so in this case the boundary is set at 40.000 nmol/L, which is identical to 2MCA1.

Creatine, MMA and CKD-EPI

All of these values do not have any outliers or negative values. The datasets look normally distributed.

Ratio 2MCA1/2MCA2

The ratio had a couple of extremely high values, having a ratio higher than 100. These ratios are extreme outliers compared to the rest of the data and made the histogram look skewed. These extreme outliers have a disproportionate effect on the model generation and their error rates. Therefore they are removed.

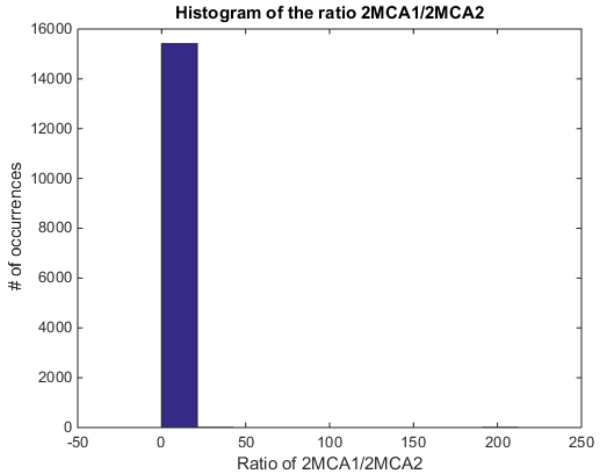

Figure 11: Histogram of 2MCAR before removal

Selecting the necessary data

Only data which had all the values of 2MCA1, 2MCA2, 2MCAR, MMA, B12 and CKD-EPI were retained, which resulted in a dataset with 7280 measurements.

3.5 Getting to know the data

An important part to data modelling is getting a feel for the data itself. It allows for selecting the right modelling techniques. We get to know the data by looking at graphs, like histograms and scatterplots.

3.5.1 Histograms

In the previous section we used histograms to determine outliers of the data. We can also look at histograms in order to see what distribution the data has. There are different distributions, but the most important to see is whether or not the data is skewed towards low or high values. In our case we can see that 2MCA1, 2MCA2, 2MCAR and MMA are clearly skewed towards lower values.

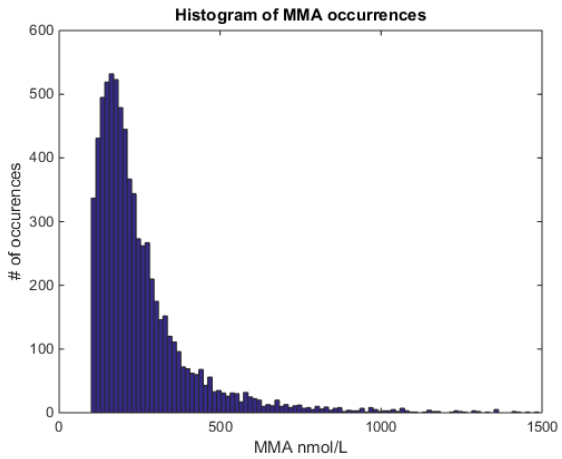

Figure 12: Histogram of MMA occurrences

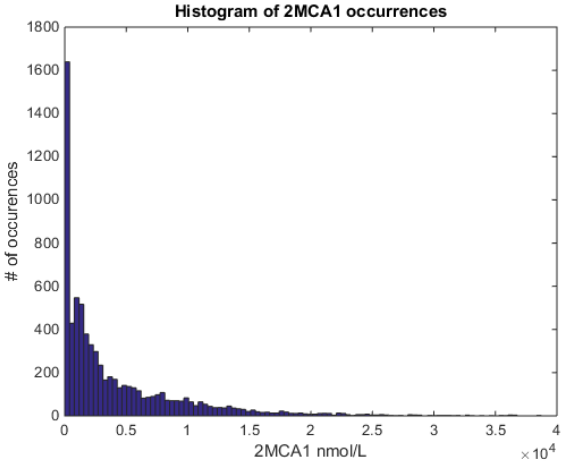

Figure 13: Histogram of 2MCA1 occurrences

The values being skewed is important, especially because MMA is the output value of the model. This leads to the conclusion that during modelling we need to search for methods that will be able to produce more accurate models with a skewed output distribution. The skewedness of 2MCA1 and 2MCA2 is in their nature, since low values are their standard, and high values are an exception.

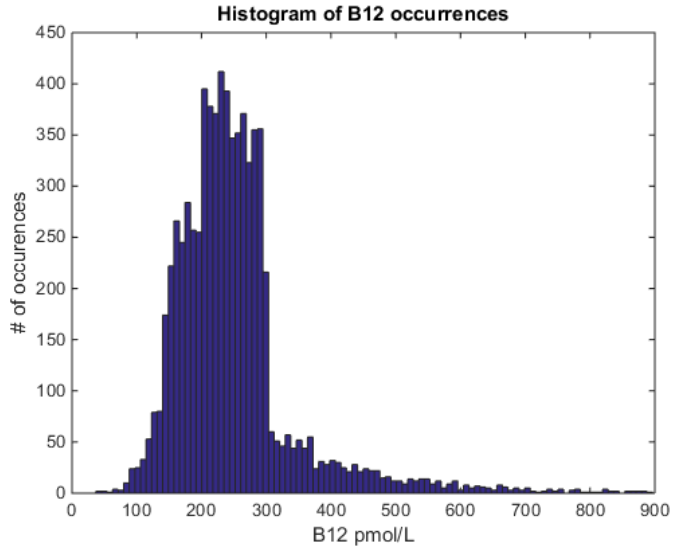

Figure 14: Histogram of B12 occurrences

The histogram of B12 shows an enormous amount of values below 300, compared to values above 300. This is most likely because additional tests are done when patients reach below that threshold. These patients are suspected to have a vitamin B12 deficiency. Since the goal of the model is to help the diagnosis of vitamin B12 deficiency, a larger suspected group to train the data on is not necessarily bad.

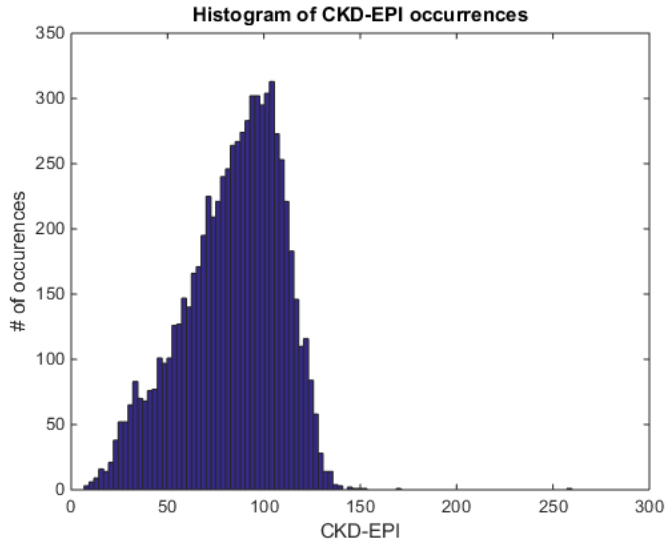

Figure 15: Histogram of CKD-EPI occurrences

The histogram of CKD-EPI does not show any irregularities, that we have to account for.

3.5.2 Scatterplots

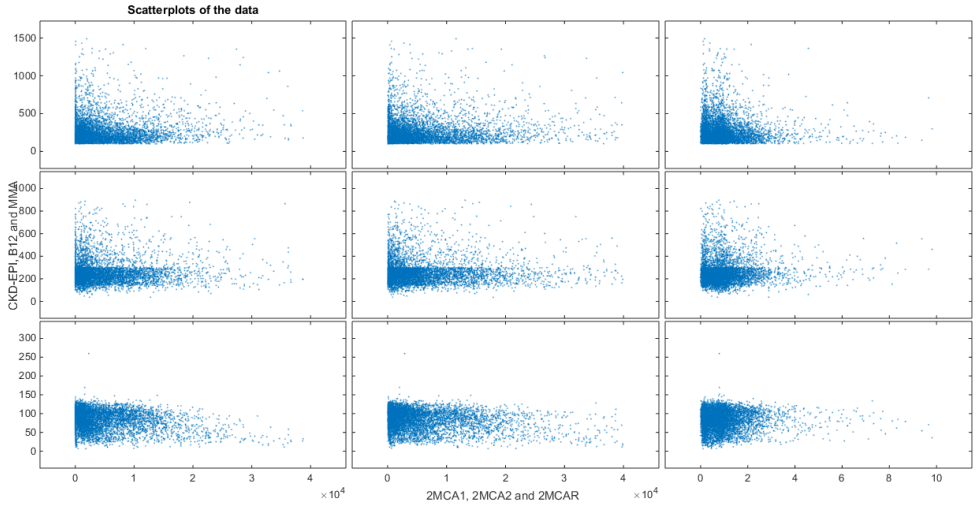

Figure 16: Scatterplots of the data

In order to get a better insight in the relationships between 2MCA1, 2MCA2, 2MCAR and the other values, their respective scatterplots were made. We can see a clear inverse relationship between all of the 2MCA values and B12 as well as MMA. This shows that there is a relationship as shown in research before. However this does not automatically mean that an artificial intelligence model will find a relationship, or that it is better than a model without 2MCA added. It can still be the case that the model with B12 and CKD-EPI added have the same information. Furthermore, more complicated relationships or relationships between more than two variables are not visible in scatterplots.

The histograms also show the visible bias in plasma vitamin B12 measurements. A lot of extra data points are visible below the 300 line.

3.6 Artificial intelligence methods

Three main modelling techniques were used:

- Fuzzy Inference Systems (FIS)
- Neural Networks
- Bootstrap aggregated decision trees

One of the important factors in fuzzy inference systems is the way that rules are generated or determined. The first method performed is by fuzzy c-means clustering, which was used in the previous study about this subject (Wilbik et al., 2016). The following variations in cluster selection were performed:

- Fuzzy c-means clustering
- Gustafson Kessel
- Subtractive clustering
- Maximum size partitioning with double gaussian membership functions
- Threshold based partitioning
- Equal distance partitioning with triangular membership functions
- Weighted Fuzzy c-means clustering
- Weighted Gustafson Kessel clustering
- Weighted equal distance partitioning
- Subtractive clustering with balanced training
- Subtractive clustering with balanced training and cluster selection
- Fuzzy c-means clustering with additional biomarkers

3.7 Fuzzy Inference systems

3.7.1 Fuzzy c-means clustering

3.7.1.1 Individual relationships between 2MCA and other data

The individual relationships between 2MCA1, 2MCA2, 2MCAR and CKD-EPI, B12, MMA are evaluated by performing fuzzy c-means clustering. These fuzzy c-means clusters were evaluated with the Pearson cluster correlation validity index and spearman cluster correlation validity index. The pairs were also made into Takagi-Sugeno 1st order fuzzy inference systems using the function genfis3 and the parameters were improved by the Matlab function anfis.

The models were evaluated with mean absolute percentage errors. All of this was done from 2 up to 10 clusters.

3.7.1.2 1st order Takagi-Sugeno

The benchmark and all 7 different variations were made into 1st order Takagi-Sugeno fuzzy inference systems using the Matlab function genfis3. The clusters were selected by fuzzy c-means clustering. The parameters were improved with anfis. This was done from 2 up to 10 clusters.

3.7.2 Gustafson Kessel clustering

The Gustafson Kessel clustering technique was used to make a fuzzy partition. This partition was turned into a 1st order Takagi-Sugeno fuzzy inference model with combined gaussian functions. The model consequent parameters were improved by global least squares.

10 times 10-fold cross validation was used. Sensitivity analysis was performed by changing the parameters individually compared to the standard. The initial values of ρ_i and M where chosen by Balasko, Abonyi and Feil (Balasko et al., 2005).

Table 5: Table showing the initial parameters for Gustafson Kessel

	Parameter	Initial value
ρ_i	Changes cluster volumes	[1,1,1,1,1,1,1,1,1,1]
C	Changes number of clusters	10
M	Changes the fuzziness	2

Table 6: Table showing the changed parameters of Gustafson Kessel

Analysis name	Changed parameter	Changed value
Variation 1	ρ_i	[1,1,1,1,1,0.5,0.5,0.5,0.5,0.5]
Variation 2	C	4
Variation 3	M	4

3.7.3 Subtractive clustering

Subtractive clustering was used to select cluster centres and make a fuzzy partition matrix. The fuzzy partition matrix was used to estimate combined gaussian membership functions. The consequent parameters were improved by global least squares.

Sensitivity analysis was conducted on the parameter R_a from 0.2 to 0.8

3.7.4 Partitioning

3.7.4.1 Maximum size partitions

In order to reduce the amount of partitions in the model a minimum amount of data points needed to transform a partition into a rule was set. This prevents the model from making a rule for only a couple of points, which might cause problems with the model overfitting to our current data. The minimum amount of data points that was accepted was 1% of the data, which when rounded was 73 points. The input vectors were partitioned into 3 parts.

The data was partitioned into three parts. The partitions then were filtered on a minimum amount of data. The cluster centres were calculated by doing subtractive clustering for all of the different partitions separately with $R_a = 0.2$. Afterwards the fuzzy membership matrix was determined with subtractive clustering with the same R_a.

The fuzzy membership matrix was used to fit combined Gaussian membership functions. A 1st order Takagi-Sugeno fuzzy inference system was made. The antecedent parameters were improved by global least squares minimization.

With and without MMA

Partitioning with and without MMA were tried in order to see the differences between the selected cluster values, and their respective performances.

3.7.4.2 Partitions based on human thresholds

These are the inputs and their low, normal and high values according to different sources. We only partition the inputs in this case and not MMA.

Table 7: Table showing the different partition of values based on human interpretation

Input	**Low**	**Normal**	**High**	**Source**
2MCA1	<21	21-97	>97	(Allen et al., 1993)
2MCA2	<37	37-136	>136	(Allen et al., 1993)
2MCAR	<0.43	0.43-0.95	>0.95	(Allen et al., 1993)
Plasma B12	<300	300-600	>600	(Arendt, Farkas, Pedersen, Nexo, & Sørensen, 2016)
CKD-EPI	<60	60-90	>90	(MedlinePlus, 2015)

The minimum amount of data points in a partition was selected to be 2,5% in order to avoid having a lot of rules, since those could not be automatically generated and transformed into a model in this case. 10 times 10-fold cross validation was used to evaluate the models.

The partitions which met the minimum of points had cluster centres selected by individual subtractive clustering. The degree of fulfilment was made with the function of subtractive clustering as well, by using $R_a = 0.2$.

$$\mu_{i=} e^{-\alpha\|y-y_i^*\|^2} \; where \; \alpha = \frac{4}{R_a^2} \tag{38}$$

3.7.4.3 Equal distance partitioning

The Matlab function genfis1 was used to create these equal distance partitions. The membership functions were triangular and can overlap as can be seen in figure X. The models made were 0 order Takagi-Sugeno models. This means that the output functions only have a constant value. The model consequent parameters were trained with global least squares optimization.

3 and 5 partitions were tried in order to check if the amount of partitions would matter.

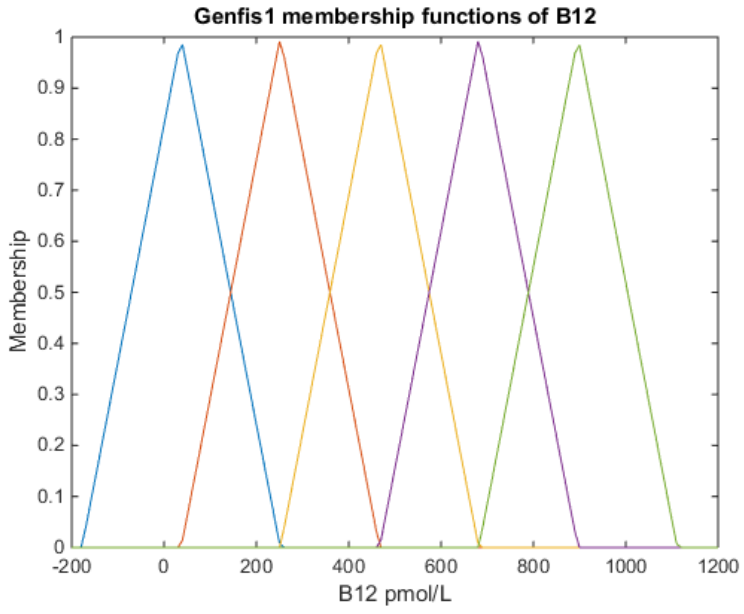

Figure 17: Visualization of the membership functions generated by genfis1

3.7.5 Adapting and balancing the data

3.7.5.1 Manner of generating weights

The weights were generated by normalizing the data between 0 and 1 in order to negate the difference between the different ranges of input. Then the sum of the Euclidean distances is calculated. This sum will be higher when the distance to the other points is bigger, therefore giving a higher weight to datapoints in a less dense area.

$$w_i = \sum_{j=1}^{N} \left\| x_i - x_j \right\|_2 \tag{39}$$

Where x_i is the input vector of datapoint I and *N* is the amount of datapoints.

The figures below show the weights generated by datapoints with both CKD-EPI and B12, which are then compared to the histograms. It is clear that this method of generating weights does give lower weights to values which are in high density areas.

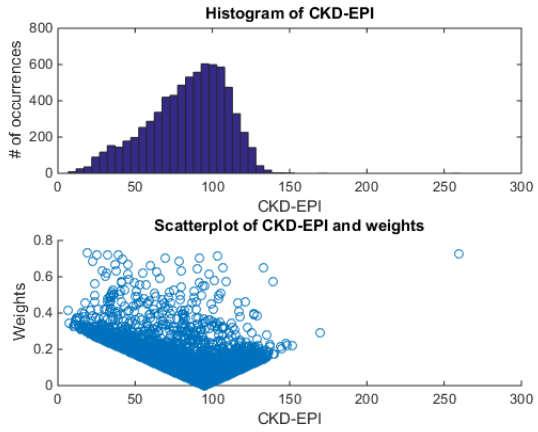

Figure 18: The histogram of CKD-EPI compared to the scatterplot of the weights

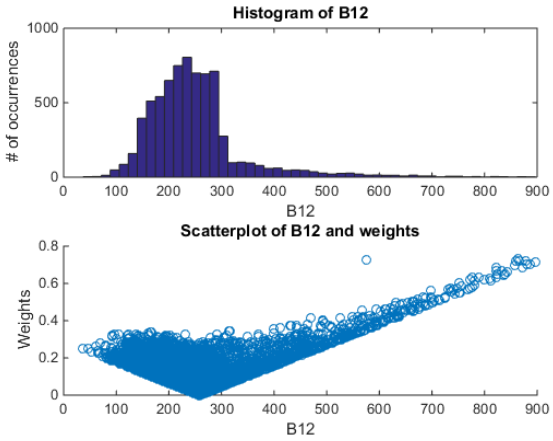

Figure 19: The histogram of B12 compared to the scatterplot of the weights

3.7.5.2 Fuzzy c-means clustering with weights

The clusters were selected by weighted fuzzy c-means clustering. Then the degree of fulfilment matrix was used to estimate combined gaussian membership functions. These functions then are used to generate a first order Takagi-Sugeno fuzzy inference system. The parameters were optimized with anfis.

3.7.5.3 Gustafson-Kessel clustering with weights

The Gustafson-Kessel clustering algorithm was adapted in order to facilitate the weights. The weighted Gustafson-Kessel generated a degree of fulfilment matrix. This matrix was used to fit combined gaussian membership functions and create a 1st order Takagi-Sugeno fuzzy inference system. The consequent parameters were improved with global least squares.

10 times 10-fold cross validation was used to evaluate. Sensitivity analysis was performed by changing the parameters individually compared to the standard. The initial values of ρ_i and M where chosen by Balasko, Abonyi and Feil (Balasko et al., 2005).

Table 8: Table showing the initial and changed values of the parameters of weighted Gustafson Kessel

	Parameter	Initial value	Changed value
ρ_i	Changes cluster volumes	[1,1,1,1,1,1,1,1,1,1]	[1,1,1,1,1,0.5,0.5,0.5,0.5,0.5]
C	Changes number of clusters	10	4
M	Changes the fuzziness	2	4

Derivation of optimal parameters for weighted Gustafson-Kessel

With the addition of weights to the Gustafson Kessel objective function, we will need to derive the necessary objective functions anew. Therefore we will follow the original GK report (Gustafson & Kessel, 1979).

Here w_{ij} is the membership of sample i of cluster j, x_i is a sample, M is a symmetric and positive-definite, α is a smoothing parameter and p_i is a volume constraint.

The distance matrix:

$$d_{ij}(\theta_j) = (x_i - v_j)^T M_j (x_i - v_j), 1 \leq j \leq k \tag{40}$$

The volume constraint of the determinant matrix M_j

$$|M_j| = \rho_j, \rho_j > 0 \tag{41}$$

$$new\ cost\ function: J(w, \theta, \lambda, \beta, c) = \sum_{i=1}^{N} \sum_{j=1}^{k} c_i w_{ij}^{\alpha} d_{ij}(\theta_j) + \sum_{i=1}^{N} \lambda_i \left(\sum_{j=1}^{k} w_{ij} - 1\right) + \sum_{j=1}^{k} \beta_j (|M_j| - \rho_j) \tag{42}$$

The new necessary conditions are now:

$$\left.\frac{\partial \bar{J}}{\partial v_j}\right|_{*} = -2 \sum_{i=1}^{N} c_i w_{ij}^{\alpha} M_j (x_i - v_j^*) = 0; j = 1,2 \ldots, k \tag{43}$$

And

$$\left.\frac{\partial \bar{J}}{\partial M_j}\right|_{*} = 0 = \sum_{i=1}^{N} c_i w_{ij}^{\alpha} (x_i - v_j)(x_i - v_j)^T + \beta_j |M_j^*| M_j^{*-1}; j = 1,2 \ldots, k \tag{44}$$

Rearranging the first condition (43) gives:

$$v_j^* = \frac{\sum_{i=1}^{N} c_i w_{ij}^{\alpha} x_i}{\sum_{i=1}^{N} c_i w_{ij}^{\alpha}} \tag{45}$$

Rearranging the second condition (44) gives:

$$M_j^{*-1} = \frac{1}{\beta_j |M_j^*|} \sum_{i=1}^{N} c_i w_{ij}^{\alpha} (x_i - v_j^*)(x_i - v_j^*)^T \tag{46}$$

Now we can define the fuzzy covariance matrix for Γ_j by

$$P_{fj} = \frac{\sum_{i=1}^{N} c_i w_{ij}^{\alpha} (x_i - m_{fj})(x_i - m_{fj})^T}{\sum_{i=1}^{N} c_i w_{ij}^{\alpha}}; \alpha > 1 \tag{47}$$

Now we can rewrite the determinant matrix of the optimum (46) with the help of the last function (47) and the constraints (41) to:

$$M_j^{*-1} = \left(\frac{1}{\rho_j |P_{fj}|}\right)^{1/n} P_{fj} \tag{48}$$

Where n is the feature space dimension

The function to calculate w_{ij}^* remains unaffected since the weights cancel out in the equation

$$\frac{d_{ij}}{d_{il}} = \frac{c_i d_{ij}}{c_i d_{il}}$$

$$w_{ij}^* = \frac{1}{\sum_{l=1}^{j} (d_{ij}/d_{il})^{1/(\alpha-1)}} \tag{49}$$

New algorithm:

Given the dataset Z, choose the standard parameters $K, \alpha, \varepsilon, p_i$ and the condition number threshold β. Initialize the partition matrix and compute the covariance matrix F_0 of the whole dataset.

Repeat for l = 1,2,…

Step 1: Compute the prototypes (means)

$$v_j^* = \frac{\sum_{i=1}^{N} c_i w_{ij}^{\alpha} x_i}{\sum_{i=1}^{N} c_i w_{ij}^{\alpha}} \tag{50}$$

Step 2: Compute the cluster covariance matrices

$$P_{fj} = \frac{\sum_{i=1}^{N} c_i w_{ij}^{\alpha} (x_i - m_{fj})\,(x_i - m_{fj})^T}{\sum_{i=1}^{N} c_i w_{ij}^{\alpha}} \; ; \; \alpha > 1 \tag{51}$$

Step 3: Reconstruct the distances

$$d_{ij}(\theta_j) = (x_i - v_j)^T M_j (x_i - v_j), 1 \le j \le k \tag{52}$$

Where

$$M_j^{*-1} = \left(\frac{1}{\rho_j |P_{fj}|}\right)^{1/n} P_{fj} \tag{53}$$

Step 4: Update the partition matrix

For $1 \le i \le N$, if $d_{ij} > 0 \; for \; 1 \le j \le K$,

$$w_{ij}^* = \frac{1}{\sum_{l=1}^{j} (d_{ij}/d_{il})^{1/(\alpha-1)}} \tag{54}$$

Otherwise

$$w_{ij}^* = 0 \; \boldsymbol{if} \; d_{ij} > 0 \; \boldsymbol{and} \; w_{ij}^* \in [0,1] \tag{55}$$

$$\boldsymbol{Else} \sum_{i=1}^{N} w_{ij} = 1$$

$$\textbf{Repeat until } \left\| W^l - W^{(l-1)} \right\| < \varepsilon \tag{56}$$

The algorithm used already resulted in a fuzzy membership matrix. This matrix then was used to estimate combined gaussian membership functions which were turned into a 1st order Takagi-Sugeno model. The model was trained and improved with global least squares minimization.

3.7.5.4 Balanced subtractive clustering

For training

The cluster selection was done with unbalanced data in the same way as during subtractive clustering. Then the fuzzy membership matrix was turned into a 1st order Takagi-Sugeno model with combined Gaussian membership functions. With $R_a = 0.2$. The difference was during the optimization of the model consequent parameters. The training was performed by a dataset which had gained an equal amount of points with MMA higher than 300, as points below 300. This was done by repeating the set fully, until it was no longer possible, then random selection without replacement was performed. Evaluation was performed with unbalanced data.

For training and cluster selection

The cluster selection was performed with data balanced in the same way as during training. The evaluation was still performed with unbalanced data.

3.7.5.5 Equal distance weighted training

The models were generated with genfis1, which made 3 equal distance partitions. The partitions were turned into 0-order Takagi-Sugeno models with triangular membership functions.

3.7.6 Adding biomarkers

Sometimes changes in models just do not suffice in generating better models. Adding more input variables can help the model find relationships which were not visible without those new inputs. In order to make the new input more likely to succeed in generating a better model we go to literature to find sensible options. It also needed to be in the data which was available.

3.7.6.1 Age

The method used was a standard fuzzy inference system of genfis3. The clusters were selected by fuzzy c-means clustering with 4 clusters selected. 10 times 10-fold partitioning was used for training and testing. Anfis was used to improve the model parameters.

3.7.6.2 Homocysteine

The same method was used when age was added. Only in this case, not all of the 7280 datapoints had Hcy. When removing the datapoints that did not have Hcy 3450 datapoints remained, which is about half. This makes us unable to correctly compare the results of the errors, though if the models do discover a pattern it should be visible in the scatterplot of the errors.

3.7.6.3 Folate

The same method was used as described in chapter 3.6.6.1, when age was added. This time 2308 data points had folate. So the error values cannot be compared to the other models. Though internally we can compare the different error values.

3.7.6.4 Haemoglobin

The method was the same as in section 3.6.6.1. As explained when age was added. This time 7099 data points had haemoglobin.

3.7.6.5 All 4 of the additional biomarkers

The same method as when adding age was used, which is described in section 3.6.6.1, though in this case only 970 data points were available.

3.8 Neural networks

3.8.1 Fitnet

A fitnet was created for which the parameters were improved by Levenberg-Marquardt backpropagation. The model was evaluated by 10 times 10-fold cross validation. 3 different models were tried to see if layer size or layer amount would change model performance.

- 3 Hidden layers with size 10-10-10
- 3 Hidden layers with size 30-30-15
- 4 Hidden layers with size 10-10-10-10

3.8.2 Cascadeforwardnet

A fitnet was created for which the parameters were improved by Levenberg-Marquardt backpropagation. The model was evaluated by 10 times 10-fold cross validation. 3 different models were tried to see if layer size or layer amount would change model performance.

- 3 Hidden layers with size 10-10-10
- 3 Hidden layers with size 30-30-15
- 4 Hidden layers with size 10-10-10-10

3.8.3 Balanced Fitnet and Cascadeforwardnet

Both fitnet as well as a cascadeforwardnet were created with balanced data. The models parameters were improved with Levenberg-Marquardt backpropagation. The dataset gained an equal amount of points with MMA higher than 300, as points below 300. This was done by copying the smallest set fully, until it was no longer possible, then random selection without replacement was performed. The error values were still calculated with unbalanced data. This was all done by 10 times 10-fold cross validation.

- 3 Hidden layers with size 10-10-10
- 3 Hidden layers with size 30-30-15
- 4 Hidden layers with size 10-10-10-10

3.9 Bootstrap aggregated decision trees

3.9.1 With normal data

Tree size was varied to see if performance changed. First a 100 trees were selected, afterwards 1000 were selected. The models were trained and tested by 10 times 10-fold cross validation.

3.9.2 With balanced data

The dataset to train the models gained an equal amount of points with MMA higher than 400, as points below 400. This was done by repeating the higher set fully, until it was no longer possible, then random selection without replacement was performed. The error values were still calculated with unbalanced data. This was all done by 10 times 10-fold cross validation.

4 Results

This chapter describes the results of all of the methods which were used to generate models. For some of the results the visual aids like scatterplots and histograms are in the appendix since they look too similar to previous results to warrant them being placed in this chapter.

4.1 Fuzzy inference systems

4.1.1 Individual relationships between 2MCA and the other data vectors and cluster selection

4.1.1.1 Mean absolute percentage error

The mean absolute percentage error showed in nearly all cases a preference for 2 clusters, with the exception for the measurement pair which had 2MCA2 as input and CKD-EPI as output. In that case 3 clusters performed the best. Though the most important observation of the MAPE values is that the performance of the fis-models does not change drastically when selecting different amount of clusters. There are big differences between the performances of the of the pairs. We can see that the data with the 2MCAR output is significantly lower than the other two outputs. While 2MCA2 is lower than 2MCA1, though both result in average errors of around 90%. There is not much difference between the inputs.

The better performance of 2MCAR could possibly also be prescribed to a different distribution of those values. It might be because the maximum value of 2MCAR is 10 and the maximum of 2MCA1 and 2MCA2 are 40.000, therefore making it less likely that a predicted value of 2MCAR would be high as a percentage of the actual value compared to 2MCA1 and 2MCA2.

When using the 2MCA values as input in the fis models we can see lower mean absolute percentage errors. There is not much difference between the selected amount of clusters. In only one case 5 clusters are selected, but the difference between 5 and 2 clusters was neglectable. Differences between 2MCA values are not seen, but there is a clear difference in predictability of B12 and CKD-EPI compared to MMA. Where CKD-EPI and B12 get really low errors of around 25%, MMA gets errors of around 40%. Compared to the situation which had 2MCA as output, even the 40% values of MMA are good.

This difference is contrary to what was expected, since the influence of B12 on CKD-EPI goes via MMA, the expectation would be that MMA is a better more direct prediction compared to B12, though this does not seem to be the case.

Table 9: Mean absolute percentage errors with 2MCA values as output with the amount of clusters in brackets

Lower is better		**Output**		
	MAPE	*2MCA1*	*2MCA2*	*2MCAR*
Input	*MMA*	(2) 0,92901	(2) 0,9014	(2) 0,6350
	B12	(2) 0,9319	(2) 0,9041	(2) 0,6322
	CKD-EPI	(2) 0,9207	(3) 0,8883	(2) 0,6354

Table 10: Mean absolute percentage errors with 2MCA values as input with the amount of clusters in brackets

Lower is better		**Input**		
	MAPE	*2MCA1*	*2MCA2*	*2MCAR*
Output	*MMA*	(2) 0,4173	(2) 0,4184	(2) 0,4220
	B12	(2) 0,2559	(2) 0,2562	(2) 0,2552
	CKD-EPI	(2) 0,2410	(2) 0,2441	(5) 0,2471

4.1.1.2 Pearson cluster correlation validity

The Pearson values selected nearly the same amount of clusters as the MAPE did. Though in this case the pair with 3 clusters is the 2MCAR and CKD-EPI pair. For the Pearson value there was a significant difference between clusters. 2 Clusters would start out with a high value, but it would go downhill from there usually ending with a value of 0.43xx at 10 clusters. The absolute difference between the Pearson values do not carry any significance, it only shows if some cluster is better or worse, but not how much better or how much worse. We can see that 2MCA2 performs the best, followed by 2MCA1 and then 2MCAR.

There is no noticeable difference between MMA, B12 and CKD-EPI values with the exception of being paired with 2MCAR. There we can see that 2MCAR can form better clusters with B12 and MMA than with CKD-EPI, though how much better is not visible with Pearson values.

Table 11: Pearson cluster correlation validity index with the amount of clusters in brackets

Higher is better

Pearson	*2MCA1*	*2MCA2*	*2MCAR*
MMA	(2) 0,77710	(2) 0,80452	(2) 0,75873
B12	(2) 0,77714	(2) 0,80452	(2) 0,76850
CKD-EPI	(2) 0,77713	(2) 0,80451	(3) 0,72487

4.1.1.3 Spearman cluster correlation validity

The Spearman values show a whole different story. Where the previous two evaluations generally selected 2 clusters, the Spearman values do not. The clusters range from 5 or 6 to 8 or 9. In most cases the performance of the amount of clusters would rise, then peak at the selected cluster and fall again looking like a mountain shaped graph. Differences between the worst values (typically 2 clusters) were generally 0.05. Besides that we can see that 2MCAR performs slightly worse than the other two measurements. While the pair of 2MCAR and B12 was the best 2MCAR pair with the Pearson values, in this case it is the worst 2MCAR pair. This affirms that the differences between values do not indicate any significant and noticeable performance improvement.

Table 12: Spearman cluster correlation validity with the amount of clusters in brackets

Higher is better

Spearman	*2MCA1*	*2MCA2*	*2MCAR*
MMA	(8) 0,90220	(9) 0,90199	(8) 0,87160
B12	(8) 0,89669	(6) 0,89533	(9) 0,84270
CKD-EPI	(5) 0,88967	(6) 0,89285	(9) 0,87921

4.1.1.4 Conclusion

No sensible differences could be made out between the performance of 2MCA1, 2MCA2 and 2MCAR. Where 2MCAR could best be predicted by the fis models, the same could not be seen in the other two evaluation criteria. The Pearson and Spearman value do show rankings, but those rankings were contradictory to each other, suggesting only minor differences between the performances of the clusters formed by the pairs.

From these results no clear indication can be gained that one of the 2MCA values would have a clearer relationship to any of the other variables.

4.1.2 Fuzzy c-means clustering

The better performing models such as the benchmark and model 4 performed best when using more than 2 clusters, while the worse performing models performed best when only using 2 clusters. This suggests that a more complex model will gain a better understanding of the relationships to MMA. The benchmark model (The model without any of the 2MCA indicators added to them) still performed better than all of the other models. Model 4 (The model with 2MCAR included) was a close second and has error values similar to the benchmark model.

Table 13: Error values for fuzzy c-means clustering

	Added input	MAPE	MAE	MSE	R-squared
Model 4	2MCAR	0,3667	95,34	21393	0,2177
Benchmark	NONE	0,3654	95,19	21371	0,2185

Model 1 which used all of the different 2MCA values performed the worst out of all the models. So in this case more information in the model, does not appear to result in a better performance.

The scatterplot of the errors from the benchmark model shows a model that has trouble differentiating between low and high MMA. Furthermore all of the outcomes are restricted between 100 and 500. Both the fact that the model cannot differentiate and the fact that the outcomes are clumped together are bad signs of a model.

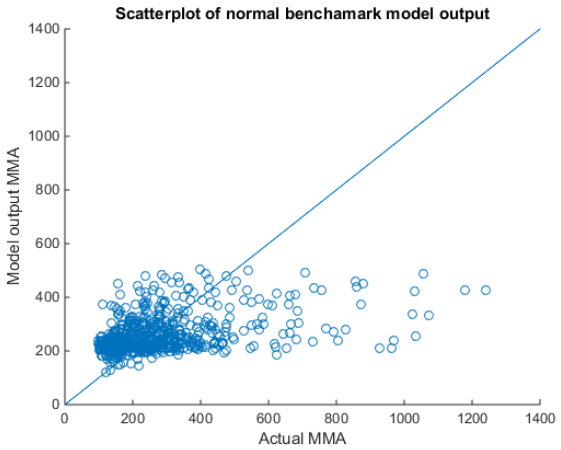

Figure 20: Scatterplot of the errors of the benchmark model, generated by fuzzy c-means clustering

This is also visible in the histogram of the errors. It shows a wide range of errors, with the most notable difference being between negative and positive errors. The positive errors go upto +300, while the negative errors go upto -800, showing how the model only has positive errors for low MMA values, while having negative errors for high MMA values.

The cluster centres in the benchmark model show that 3 of the clusters have low B12 values below 300 and only one has a value higher than 300. The CKD-EPI values are all close together.

Table 14: Cluster centres of the best performing benchmark model

Benchmark	**B12**	**CKD-EPI**
Maximum	896,70	259,73
Centre 1	234,28	90,36
Centre 2	299,13	82,19
Centre 3	165,43	81,63
Centre 4	558,99	70,23

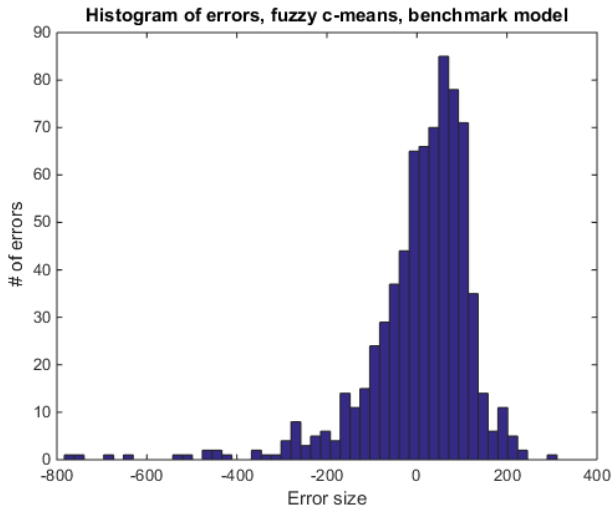

Figure 21: Histogram of the errors of the benchmark model, generated by fuzzy c-means clustering

4.1.3 Gustafson Kessel

Though the MAPE error value is worse compared to fuzzy c-means, we can see a positive difference in the scatterplot of the errors. Gustafson Kessel clustering resulted in more stretched out results. Instead of being limited to 500, the output now goes up to 600. The differences between models are smaller, but the benchmark model and the model with 2MCAR added stay the best.

Changes made to the model parameters did not affect the model performances and outcomes.

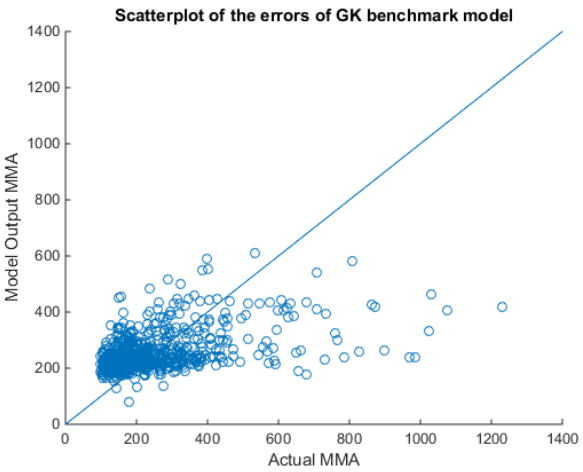

Figure 22: Scatterplot of the errors of the benchmark model, generated by Gustafson-Kessel clustering

Table 15: Error values for Gustafson Kessel clustering

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 4	2MCAR	0,3989	95,28	21519	0,2131
Benchmark	NONE	0,3972	94,92	21303	0,2210

4.1.4 Subtractive clustering

The best performing models with subtractive clustering were models with an Ra of 0.2.

Table 16: Error values for subtractive clustering

	Added input	Ra	MAPE	MAE	MSE	R-squared
Model 4	2MCAR	0,2	0,4019	95,80	21571	0.2112
Benchmark	NONE	0,2	0,4003	95,45	21624	0.2093

The model 4 with Ra of 0.2 is the best model when looking at the mean squared error and R-squared. We see a slight advantage for the benchmark model when looking at the MAPE and MAE. The differences are not big enough to say anything useful about them. The output did not change significantly.

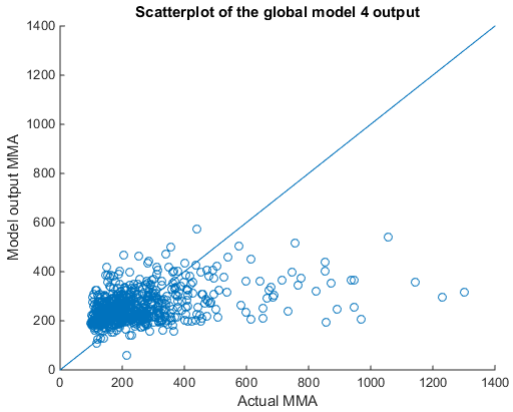

Figure 23: Scatterplot of the errors of model 4, generated by subtractive clustering

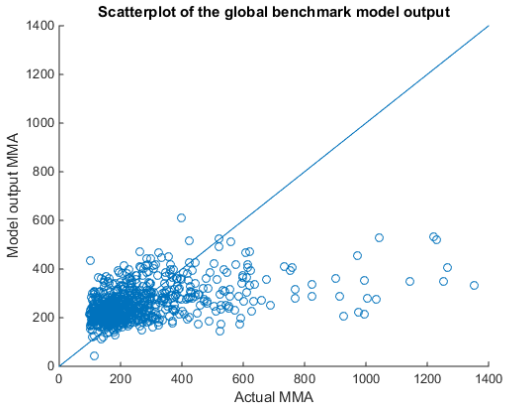

Figure 24: Scatterplot of the errors of the benchmark model, generated by subtractive clustering

As we can see in the scatterplots of the model output versus the actual MMA values, both the benchmark model and model 4 are not estimating high MMA correctly. Furthermore the model did stretch out compared to the models generated with fuzzy c-means clustering, but not compared to the Gustafson Kessel clustering. 600 still is the maximum output value of the models.

It is clear that due to the high density of Vitamin B12 measurements below 300, (as discussed in the section where we were getting to know the data), results in only cluster centres being chosen that have values of B12 below 300.

Table 18: Cluster centres of the benchmark model

Benchmark	**B12**	**CKD-EPI**
Maximum	896,70	259,73
Centre 1	238,35	95,52
Centre 2	277,82	70,06
Centre 3	179,64	112,27
Centre 4	189,21	51,89

Table 17: Cluster centres of model 4

Model 4	**2MCAR**	**B12**	**CKD-EPI**
Maximum	9,8124	896,70	259,73
Centre 1	0,6607	245,40	98,96
Centre 2	0,2878	224,00	77,26
Centre 3	1,0374	281,10	68,18
Centre 4	0,1724	182,90	115,24
Centre 5	0,9206	178,70	85,62

4.1.5 Partitioning

4.1.5.1 Maximum Size partition

Partitioning without MMA

The results show that the 4th model performed well again, with the benchmark as a close second. The difference between the two models is insignificant. We can see some improvement in the fact that more values get close to 600, instead of just a handful.

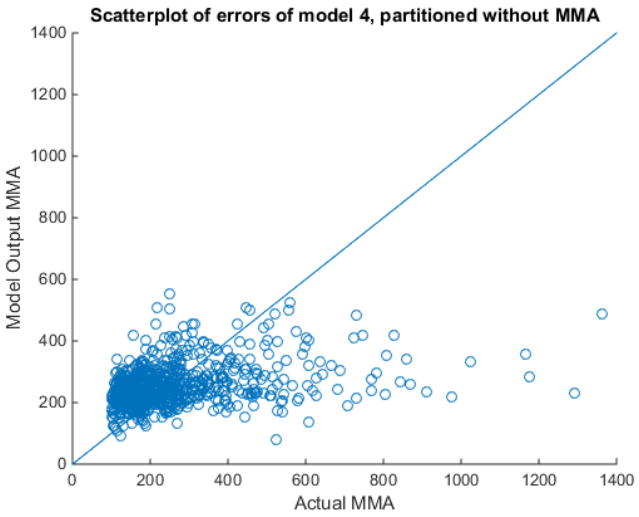

Figure 25: Scatterplot of the errors of model 4, generated by maximum size partitioning

Table 19: Error values for maximum size partitioning without MMA

	Added input	**MAPE**	**MAE**	**MSE**	**R-squared**
Model 4	2MCAR	0,4019	95,78	21577	0,2110
Benchmark	NONE	0,4044	96,36	22055	0,1935

Partitioning with MMA

Adding MMA resulted in 0-2 more clusters to fit the criteria of having at least 73 data points. Adding MMA to the cluster selection (not to the membership functions) showed a slight improvement in the model MAPE and MAE. No improvements were made in model MSE or R-squared values. A couple of points now are predicted to be higher than 600, whereas in the models without MMA during cluster selection this was not seen.

Table 20: Error values of maximum size partitioning with MMA

	Added input	MAPE	MAE	MSE	R-squared	Highest MMA of clusters
Model 4	2MCAR	0,4094	97,17	22775	0,1672	653
Benchmark	NONE	0,3988	95,23	21507	0,2136	301

The benchmark model is even more surprising since in this case it outperforms the other models, while having clusters with lower MMA values. It also has the biggest improvement in all of the error values, though the improvements are minimal.

The scatterplot shows some improvements over the previous methods, since now a handful of points are higher than 600. Though for a large part the scatterplot staid the same. The histogram of the errors show very little change as can be seen in the appendix.

Figure 26: Scatterplot of the errors of model 4, generated by maximum size partitioning

4.1.5.2 Partition based on human thresholds

The partitions were completely different compared to the previous partitions, but the results were worse. The errors were a bit worse, but mainly the scatterplot resided back to a maximum of about 500 like with the Fuzzy c-means clustering method. This way of partitioning did not stretch the data out anymore. Nor were any changes seen in the histogram of the errors.

Table 21: Error values of partitions based on the biomarkers

	Model 4	Benchmark
Added input	2MCAR	NONE
MAPE	0,4042	0,4030
MAE	96,27	96,13
MSE	21677	21667
R-squared	0,2074	0,2077

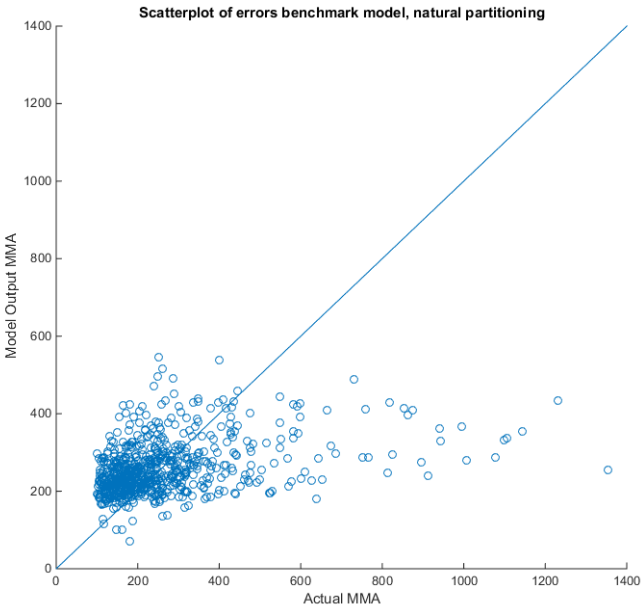

Figure 27: Scatterplot of the errors of the benchmark model, generated by partitioning based on the biomarkers

4.1.5.3 Equal distance partitioning

3 Partitions

The results did not improve, but worsened compared to the other ways of partitioning. Once again the benchmark model and the model with 2MCAR added were the best. Though the scatterplot of the errors showed no improvement and minimal differences.

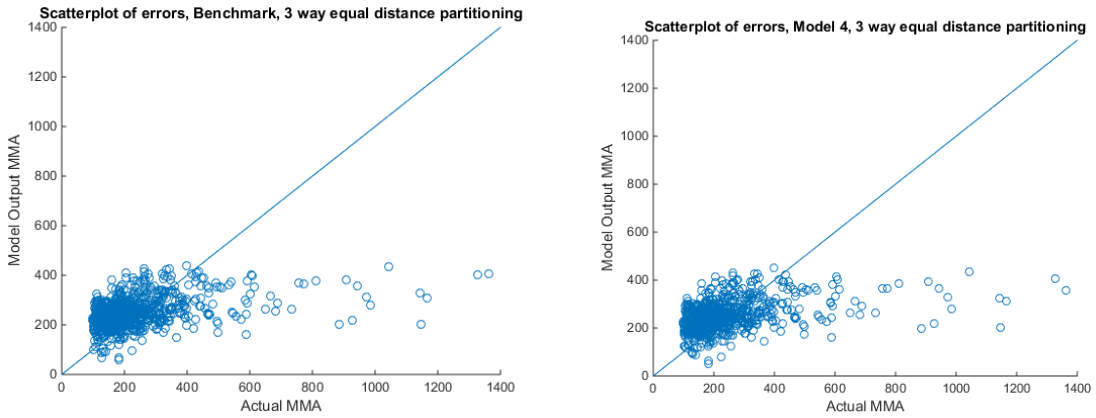

Figure 28: Scatterplot of the errors of the benchmark model versus model 4, generated by equal distance partitioning

Table 22: Error values of equal distance partitioning in 3 parts

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 4	2MCAR	0,4121	97,66	22165	0,1895
Benchmark	NONE	0,4112	97,49	22076	0,1927

5 Partitions

Five partitions did improve the error values significantly compared to 3 partitions. When compared to the other cluster selection techniques we do not see an improvement in error values, the scatterplot or the histogram of the errors.

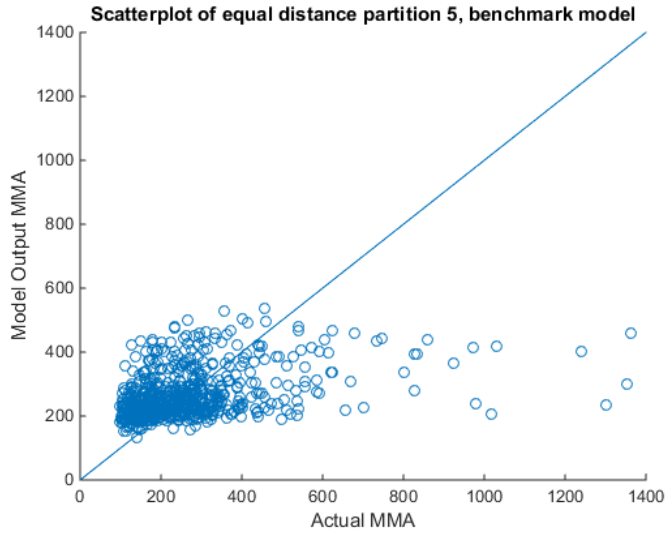

Table 23: Error values of equal distance partitioning in 5 parts

	Model 4	Benchmark
Added input	2MCAR	NONE
MAPE	0,3978	0,3971
MAE	95,20	95,00
MSE	21621	21405
R-squared	0,2094	0,2173

Figure 29: Scatterplot of the errors of the benchmark mode, generated by equal distance partitioning

4.1.6 Balancing and adapting the data

4.1.6.1 Fuzzy c-means weighted clustering

Compared to the membership functions of the original fuzzy c-means clustering technique without weights, the weighted fuzzy c-means shows a better distribution of membership functions for high vitamin B12.

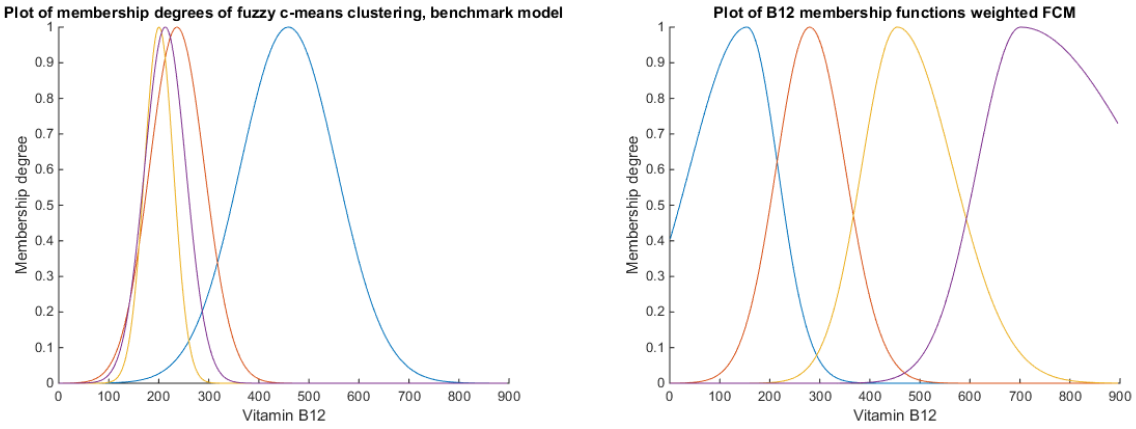

Figure 30: Membership degrees of fcm versus weighted fcm

The results of the clustering technique worsened compared to non-weighted fcm. They are now comparable to the other clustering techniques like Gustafson Kessel and subtractive clustering. Neither the scatterplot of the errors or the histogram of the errors show any improvement.

Table 24: Error values of weighted fcm

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 4	2MCAR	0,4049	96,48	21803	0,2028
Benchmark	NONE	0,4050	96,47	21799	0,2029

4.1.6.2 *Gustafson Kessel weighted clustering*

The weights added to Gustafson Kessel changed the clusters selected. A test with cluster selection of B12, CKD-EPI and MMA show that the weighted selection selected clusters with 40% higher MMA. Though to reiterate MMA was not used during the GK cluster selection, just the input values were used.

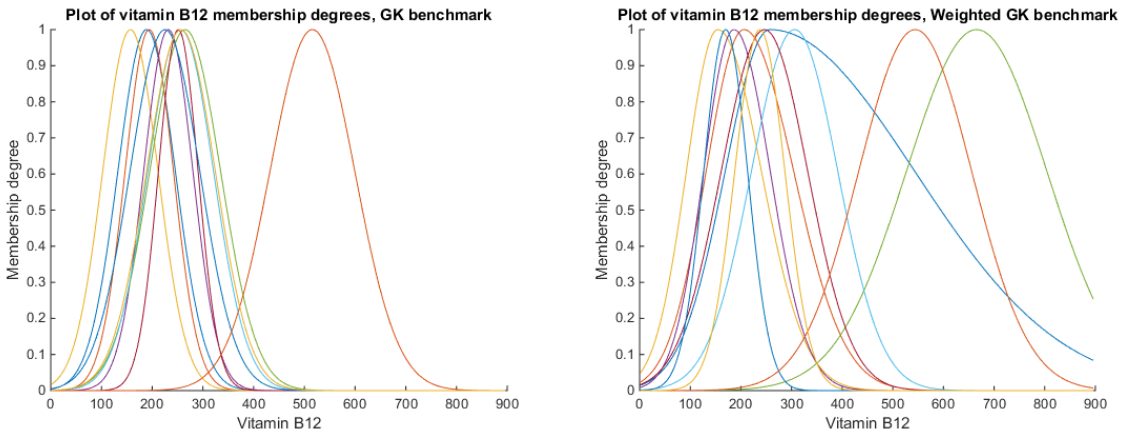

Figure 31: Membership functions of vitamin B12 from the benchmark model, non-weighted GK versus weighted GK

Besides selecting different clusters and clearly different membership functions, the results of the models do not change though. The errors stay the same which is confirmed by the Kolmogorov-Smirnov test which accepted the null hypothesis. Furthermore the scatterplot and histogram of the errors also look similar. No extra stretching occurred as a result of the clusters selected.

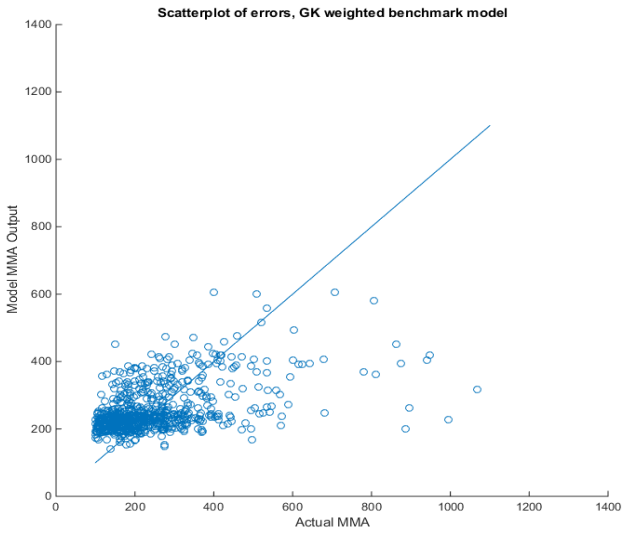

Figure 32: Scatterplot of the errors of the benchmark model, generated by weighted GK clustering

Table 25: Error values of weighted Gustafson Kessel

	Model 4	**Benchmark**
MAPE	0,3971	0,3958
MAE	94,97	94,73
MSE	21421	21250
R-squared	0,2167	0,2230

4.1.6.3 Balanced subtractive clustering for training

The scatterplot of model 4 shows that the dataset now does change its behaviour somewhat. It shows a greater propensity to high MMA values. The model does have a handful of values higher than 700. In general some stretching of the data is visible, but mainly for the low MMA values.

As usual, the benchmark model and the model with 2MCAR were close winners when compared to the rest of the models. The performance is worse compared to the original non-balanced methods, since the low MMA values are being estimated higher. This is visible in the histogram of errors showing less errors close to 0 and instead more positive errors.

Table 26: Error values of balanced subtractive clustering for training

Model name	**Added input**	**MAPE**	**MAE**	**MSE**	**R-squared**
Model 4	2MCAR	0,5919	121,63	25376	0,0721
Benchmark	NONE	0,5918	121,54	25321	0,0741

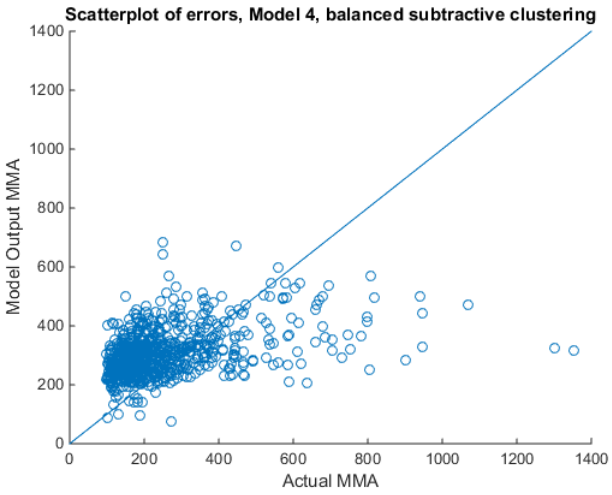

Figure 33: Scatterplot of the errors of model 4, generated by balanced subtractive clustering in training

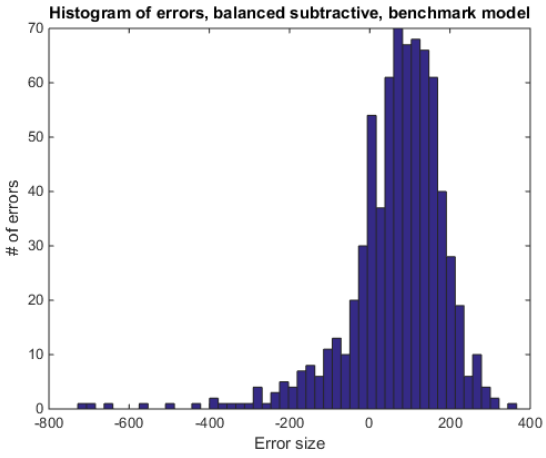

Figure 34: Histogram of errors, balanced training subtractive clustering, benchmark model

4.1.6.4 Balanced subtractive clustering for training and cluster selection

The results show comparable model to the model which was only balanced during training. One low MMA value would be predicted by the model to be 3500 and another was predicted to be -180. These mistakes cannot be seen since for comparison reasons the axes were restrained. The mistakes were big though, and we still do not see an improvement in the estimation of high MMA values.

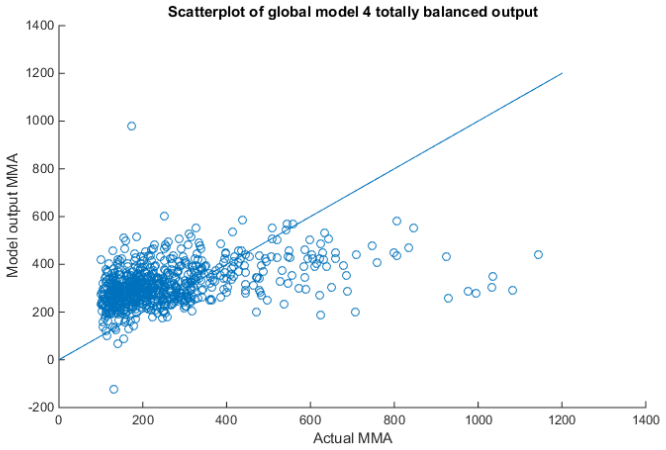

Figure 35: Scatterplot of the errors of model 4, generated by balanced subtractive clustering in training and cluster selection

Table 27: Error values of balanced subtractive clustering

	Model 4	Benchmark
MAPE	0,5942	0,5917
MAE	122,23	121,59
MSE	25560	25309
R-squared	0,0653	0,0745

4.1.6.5 0 order Takagi-Sugeno equal distance partition with weight adapted training

The errors of this method were higher compared to training without weights. This is expected since the model gets trained on weighted data and puts a different emphasis on the data points. The scatterplot of the errors still showed the same lack of any actual predictive power from the model. Furthermore, there was no stretching visible, the output is still clustered between 100 and 500.

The constants which are the consequents of the rules did change, but for the most part marginally. The biggest change between them was in the highest constant, which changed from 1430 to 1133.

There was a big difference between the MSE of model 4 and the benchmark model. Model 4 had a MSE and an R-squared value which are significantly lower. Though the benchmark model has lower MAPE and MSE values.

Table 28: Consequent parameters of non-weighted and weighted equal distance partitioning

		Consequent parameters	
B12	**CKD-EPI**	**Not Weighted**	**Weighted**
LOW	LOW	556,04	599,00
LOW	MEDIUM	243,85	220,55
LOW	HIGH	1430,11	1133,19
MEDIUM	LOW	303,91	341,55
MEDIUM	MEDIUM	31,50	40,95
MEDIUM	HIGH	0	0
HIGH	LOW	426,10	445,61
HIGH	MEDIUM	120,37	42,78
HIGH	HIGH	527,58	521,82

Table 29: Error values of equal distance partitioning with weight adapted training

	Model 4	Benchmark
MAPE	0,4546	0,4544
MAE	102,53	102,45
MSE	48487	85370
R-squared	-0.7730	-2.1217

4.1.7 Adding biomarkers

4.1.7.1 Age

The results vary a lot. All of the models still have the same previous inputs, but with Age added to them. We can see that some of the models are performing abysmal compared to their inputs without age. Models 3 (with 2MCA2), 4 (with 2MCAR) and 8 (the benchmark) are performing on an equal level as before. Though when we take a look at the scatterplot of the errors we can see the same mistakes as previously. Low MMA values are predicted too high and high MMA values are predicted too low.

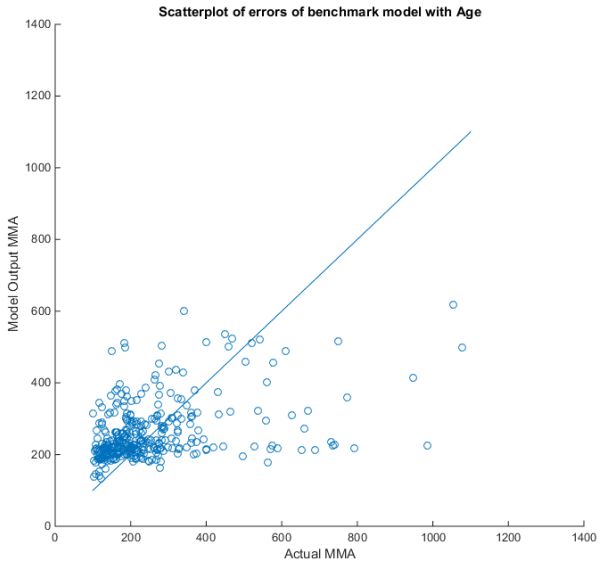

Figure 36: Scatterplot of errors of the benchmark model with age, generated by fuzzy c-means clustering

Table 30: Error values of fuzzy c-means clustering with age added

	MAPE	MAE	MSE	R-squared
Model 3	0,3801	96,68	21772	0,2039
Model 4	0,3822	95,80	21518	0,2132
Benchmark	0,3679	95,37	21379	0,2183

4.1.7.2 Homocysteine

The best performing models were models 4 and the benchmark model again. Though when we look at the scatterplot of the errors, no improvement was made in the predictive ability of the models. The best performing model still shows that it cannot determine the difference between low and high MMA values. Furthermore the models now predict negative MMA values. The benchmark model would even predict negative MMA values.

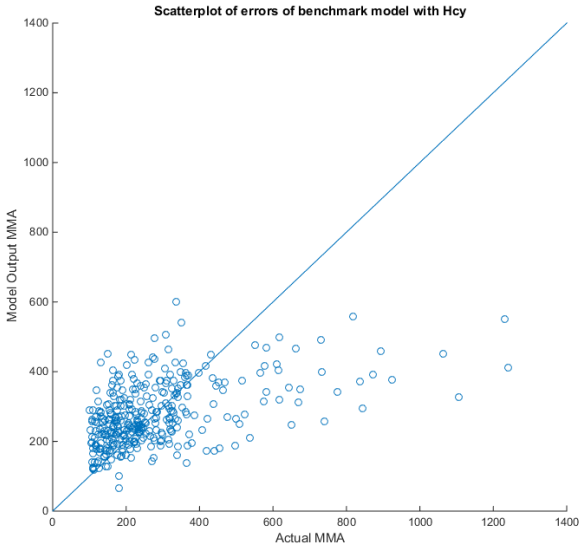

Figure 37: Scatterplot of errors of the benchmark model with Hcy, generated by fuzzy c-means clustering

Table 31: Error values of fuzzy c-means clustering with Homocysteine added

	Model 4	Benchmark
MAPE	0,3782	0,3765
MAE	111,36	110,85
MSE	28485	28245
R-squared	0,2161	0,2227

4.1.7.3 Folate

The results do not show any improvement in the predictive capabilities of the models. The only models that performed well were the model with 2MCAR and the benchmark model. Other models were performing abysmal. The scatterplot showed no improvements.

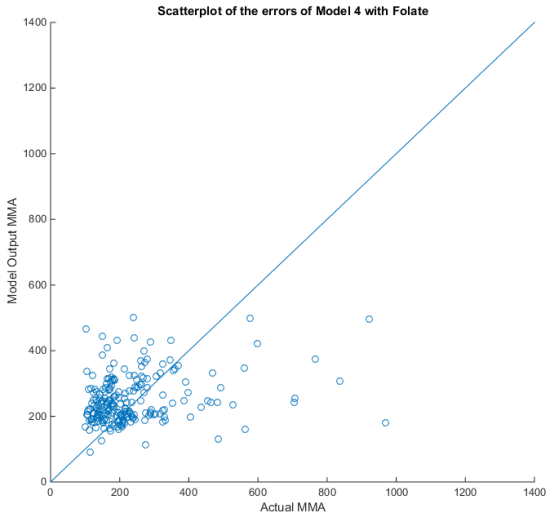

Figure 38: Scatterplot of errors of model 4 with folate, generated by fuzzy c-means clustering

Table 32: Error values of fuzzy c-means clustering with folate added

	Model 4	Benchmark
MAPE	0,3758	0,3759
MAE	98,15	98,3401
MSE	22831	22883
R-squared	0,1700	0,1681

4.1.7.4 Haemoglobin

No indication of any improvement could be seen in the scatterplots. Interestingly not only the benchmark model and the 2MCAR models with Hb added performed well, also model 3 which has 2MCA2 added performed reasonably well. Still, no improvement in the predictive capabilities of the models were visible. All the other models were performing much worse, with enormous errors.

Table 33: Error values of fuzzy c-means clustering with haemoglobin added

	MAPE	MAE	MSE	R-squared
Model 3	0,3877	95,95	21697	0,1540
Model 4	0,3641	94,41	21001	0,1811
Benchmark	0,3627	94,12	20914	0,1845

4.1.7.5 All 4 of the additional biomarkers

Though the dataset is significantly different compared to the models without the additional biomarkers, there is no improvement in the error values of the models. The models do not perform better, or close to better.

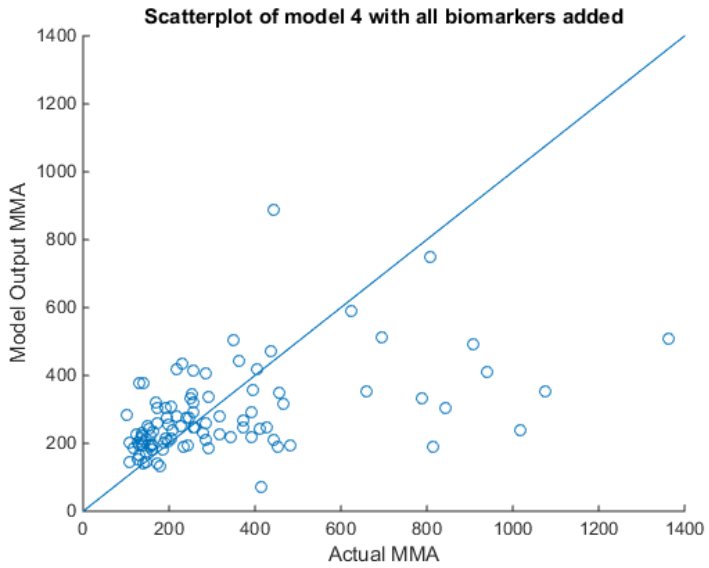

Figure 39: Scatterplot of errors of model 4 with all biomarkers, generated by fuzzy c-means clustering

Table 34: Error values of fuzzy c-means clustering with all 4 additional biomarkers added

	Model 4	Benchmark
MAPE	0,4441	0,4311
MAE	120,88	117,65
MSE	34136	32122
R-squared	0,1237	0,1754

4.2 Neural Networks

4.2.1 Fitnet

The models with 4 layers were a bit better compared to the models with 3 layers. Varying the layer sizes did not improve the models any further. We can see that the neural networks models suffer from the same traits as in the fuzzy inference models. The scatterplot shows high MMA values being underestimated, while low MMA values are overestimated. Furthermore the neural networks do not stretch the output over a larger area.

The methods did change which model was selected as best performing. Models 3,4,6 and 7 all have some error value the lowest in one of the variations. No information can be derived from this.

Table 35: Error values of the best performing fitnet neural networks

10.10.10	MAPE	MAE	MSE	R-squared
Model 3	0,3933	94,91	21723	0,2057
Model 6	0,3977	94,75	20995	0,2323
30.30.15	MAPE	MAE	MSE	R-squared
Model 3	0,4062	95,15	20715	0,2425
Model 7	0,4028	96,60	22234	0,1870
10.10.10.10	MAPE	MAE	MSE	R-squared
Model 3	0,3900	94,91	21835	0,2016
Model 4	0,4001	94,08	20645	0,2451

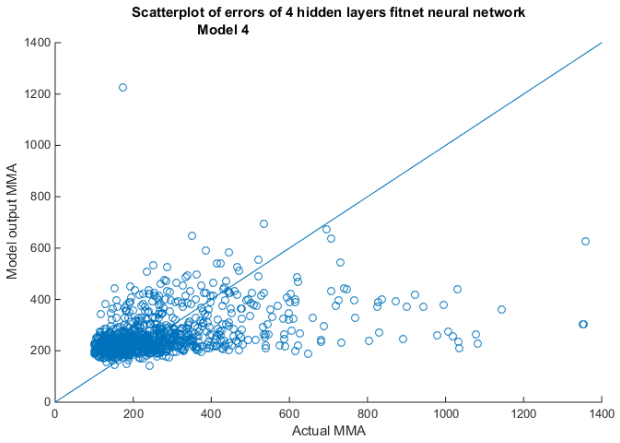

Figure 40: Scatterplot of the errors of model 4, generated by 4 layered fitnet

4.2.2 Cascadeforwardnet

The cascadeforward neural networks, like the fitnets have a wide variability in best performing models. The models 2,5,6 and all performed best at some point in time. The errors were similar to the fitnet models and neither the scatterplot of the errors or the histograms showed a noticeable difference.

4.2.3 Balancing

Balancing the neural networks during training did not lessen their variability. Now, in all 6 variants, models 1,2,5 and 7 were the best performing at some time. No model was consistently the best. Therefore no decisive conclusion can be drawn from the error values.

Looking at the scatterplots of the balanced data, we can see that balancing increased the output range only marginally. It did not improve the predictive value of the models.

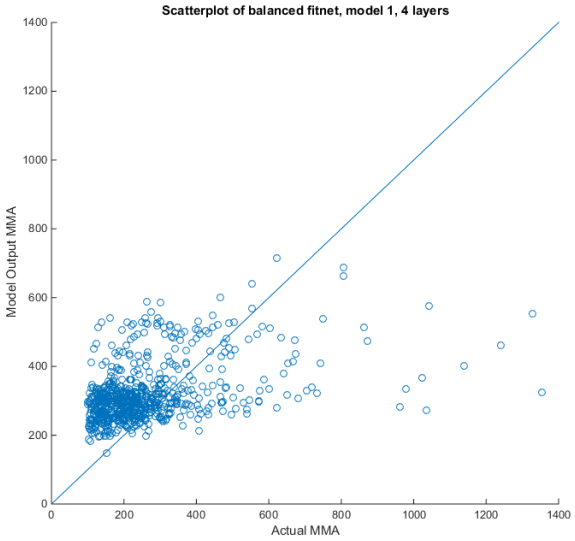

Figure 41: Scatterplot of errors of model 1, generated by balanced fitnet

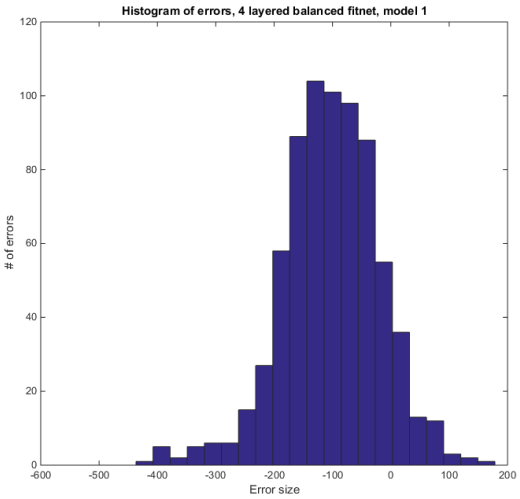

Figure 42: Histogram of errors, 4 layered fitnet, model 1

4.3 Bootstrap aggregated decision trees

4.3.1 Tree size

4.3.1.1 100 trees

The results were quite different compared to the neural networks and the fuzzy inference models. First off the best model was model 1, instead of model 4 or the benchmark model. Model 1 is the model which has all of the different 2MCA isomers added. Though all of the errors were similar. When looking at the scatterplot of the errors we can see some improvements in high MMA values. The general scatterplot still has the same mistakes, that it estimates everything to be in-between about 100 and 600.

When comparing the error values to for example the standard fuzzy inference models we can see that the MAE, MSE and R-squared are similar, but the MAPE values are way higher when doing tree bagging.

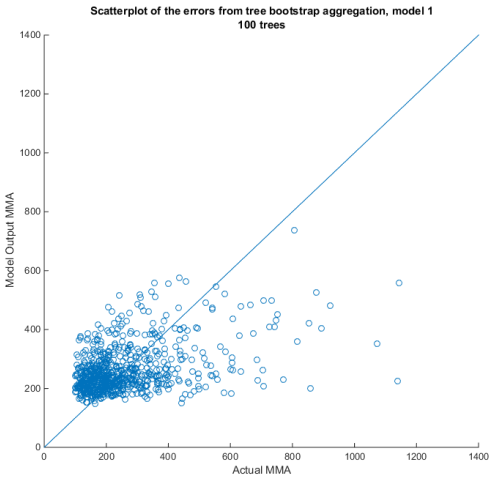

Figure 43: Scatterplot of errors of model 1, generated by bootstrap aggregated decision trees with 100 trees

Table 36: Error values of 100 bagged decision trees

	MAPE	MAE	MSE	R-squared
Model 1	0,4988	95,18	21078	0,2292
Model 4	0,5026	96,04	21684	0,2071
Benchmark	0,5105	98,46	22607	0,1733

4.3.1.2 1000 trees

More trees did not result in different outcomes. The performances staid the same and the scatterplot of the errors does not look any different. No changes were seen.

Table 37: Error values of 1000 bagged decision trees

	MAPE	MAE	MSE	R-squared
Model 1	0,4981	94,95	21023	0,2313
Benchmark	0,5107	98,42	22562	0,1750

4.3.2 Balancing

The error values were worse, as can be expected when training the models on a different set. Though the best performing model was still the model with all of the information in it. The scatterplot does show a big difference. We see that the low MMA values vary more, but the majority of the high MMA values can still not be predicted accurately by the models. The range of the output from the model has increase from 100-600, to about 100-1000. However, the values above 600 are a minority and mainly seem to come from low MMA values, instead of high MMA values.

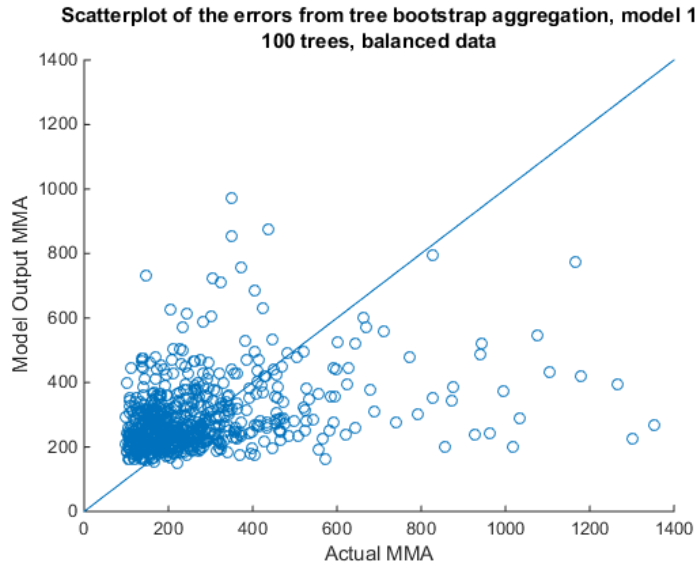

Figure 44: Scatterplot of errors of model 1, generated by balanced bootstrap aggregated decision trees with 100 trees

Table 38: Error values of 100 balanced bagged decision trees

	MAPE	MAE	MSE	R-squared
Model 1	0,6016	108,97	23717	0,1328
Benchmark	0,6375	118,21	28766	-0,0519

5 Conclusion

Can 2MCA be used to improve the diagnostic capabilities of MMA in detecting B12 deficiency?

a. Which relationship does 2MCA have to renal failure?
b. Which relationship does 2MCA have to MMA?
c. Which relationship does 2MCA have to vitamin B12 plasma?
d. Can 2MCA determine the elevation of MMA values due to renal failure?

Hypothesis a,b and c

The scatterplots showed the relationships as described in literature, but the individual fuzzy clustering methods did not reveal any important information regarding the individual differences between 2MCA1, 2MCA2 and the ratio of 2MCA1 divided by 2MCA2 (2MCAR). All of the isomers had similar error values. When generating fuzzy inference models we could see a clear difference between 2MCAR and the individual isomers of 2MCA. Models generated with 2MCAR would always be of similar error values compared to the benchmark model which just had B12 and CKD-EPI. For neural networks there was no discernible better model, while for bootstrap aggregated decision trees models with more data performed better.

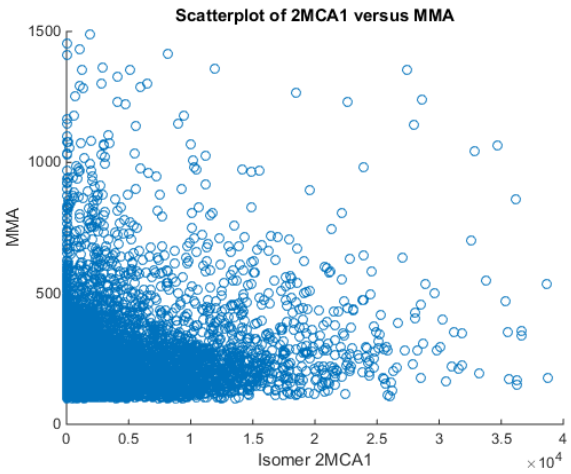

Figure 45: Scatterplot of 2MCA1 versus MMA in which the relationship between them is visible

The difference between bagged decision tree models and fuzzy inference models was mainly in the mean absolute percentage errors, which was higher for the bagged decision trees.

Therefore it can be concluded that individually there is not much difference between the relationships of 2MCA1, 2MCA2 and 2MCAR, but that when incorporated in models 2MCAR performed best.

Hypothesis d

The scatterplot of the errors of the 1st order Takagi-Sugeno fis model generated with fuzzy c-means of the benchmark model shows two problems which occurred in all of the models.

- Overestimating low MMA values and underestimating high MMA values.
- Having an output between 100 and 500.

These problems continued to occur across different artificial intelligence techniques. Variations in cluster selection increased the output range from 500 to 600.

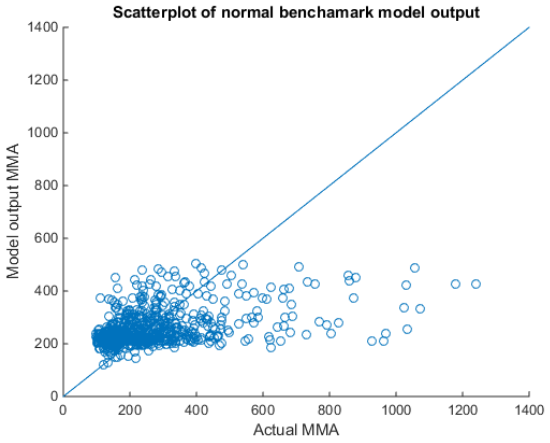

Figure 46: Scatterplot of errors, FCM first order Takagi-Sugeno model with 4 clusters, benchmark model

Balancing the data or adding weights to it did not manage to create model that stopped overestimating low MMA and underestimate high MMA. Though it did manage to get the output increased slightly to 800, but only for a handful of points. In most cases

balancing and adding weights would impact the low MMA values significantly more than high MMA values. Causing high error rates, without fixing any of the aforementioned problems.

It was suspected that the model performance was the result from bias in the data. Patients with a B12 value lower than 300, would get additional tests, including MMA values, which resulted in a selection bias in the data. After adding weights based on density to the Gustafson-Kessel function we could see that the bias disappeared in the membership functions. But a Kolmogorov-Smirnov test showed that the model itself did not change its performance.

Adding other biomarkers to the FCM generated models did not increase performance either.

The benchmark model, which is a model with only vitamin B12 and CKD-EPI as inputs nearly always outperformed the other models. In most cases the model with 2MCAR added would perform similar. One case with weights added to training of an equal distance partition 0-order Takagi-Sugeno model showed the model with 2MCAR outperforming the benchmark model in mean squared errors and in R-squared value. This model of 2MCAR did not perform any better than the original model without weights though, nor were any of the problems fixed.

Since none of the models with a variation of 2MCA isomers managed to improve compared to the benchmark models, it is clear that 2MCA does not add additional information to determine the elevation of MMA due to renal failure.

Limitations and further research

The data gathered for this project came from the Catharina Hospital in Eindhoven. A large part of the measurements came from patients which underwent bariatric surgery. Though it is not clear how large that part is, the effects of a vitamin B12 deficiency might still be in their early stages. This might have had an effect on the dataset.

Furthermore the overrepresentation of patients with vitamin B12 lower than 300 also had an effect on the dataset. Though in this report efforts were undertaken to reduce those effects, it might be better to exclude it altogether in further research and take a sample from the normal population.

Since vitamin B12 deficiency is a disease that goes through different stages of severity, additional research could focus on generating models which involve time series. This could show the development of an individual from non-deficient to deficient, showing relationships between B12, MMA and renal failure which might not be visible in one measurement.

During the final stage of the project, it came to light that during the data collection of the 2MCA isomers their scale was changed. The data thus included 2MCA measurements with two different scales. Due to time limitations only the best performing model (fuzzy c-means) was redone with the data of the biggest set of the two.

The models which had 2MCA1 or 2MCA2 in their input showed an improved performance. The models with just 2MCAR added and the benchmark were unaffected. Though the performance of those other models improved, they still were not as good as the benchmark model, or the 2MCAR model and therefore insignificant to the final conclusions. The creation and results of those models are shown in Appendix 7.4.

Recommendations for the Catharina Hospital

The models with isomers added did not provide additional information for determining vitamin B12 deficiency. This leads to the recommendation that the Catharina Hospital does not change its current

procedures and models regarding to vitamin B12 deficiency. Therefore it is justified to reconsider the usefulness of performing the tests measuring 2MCA isomers.

In accordance with the recommendations for further research the Catharina Hospital should consider to start a research project in which patients are deliberately followed for a longer period of time, in order to see if information regarding the development of B12 deficiency could help in detecting deficient patients.

6 Bibliography

Agrawal, V. (2014). What is a fuzzy set and what are its applications. Retrieved from https://www.quora.com/What-is-a-fuzzy-set-and-what-are-its-applications

Allen, R. H., Stabler, S. P., Savage, D. G., & Lindenbaum, J. (1993). Elevation of 2-methylcitric acid I and II levels in serum, urine, and cerebrospinal fluid of patients with cobalamin deficiency. *Metabolism*, *42*(8), 978–88. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/8345822

Andrès, E., Loukili, N. H., Noel, E., Kaltenbach, G., Ben Abdelgheni, M., Perrin, A. E., … Blicklé, J. F. (2004). Vitamin B12 (cobalamin) deficiency in elderly patients. *Canadian Medical Association Journal*, *171*(3), 251–259. https://doi.org/10.1503/cmaj.1031155

Arendt, J. F. H., Farkas, D. K., Pedersen, L., Nexo, E., & Sørensen, H. T. (2016). Elevated plasma vitamin B12 levels and cancer prognosis: A population-based cohort study. *Cancer Epidemiology*, *40*, 158–165. https://doi.org/10.1016/j.canep.2015.12.007

Babuška, R., Veen, P. J. Van Der, & Kaymak, U. (2002). Improved covariance estimation for Gustafson-Kessel clustering. In *2002 IEEE World Congress on Computational Intelligence. 2002 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE'02. Proceedings (Cat. No.02CH37291)* (Vol. 2, pp. 8–12). https://doi.org/10.1109/FUZZ.2002.1006654

Badde, D. S., Gupta, A., & Patki, V. K. (2009). Cascade and Feed Forward Back propagation Artificial Neural Network Models for Prediction of Compressive Strength of Ready Mix Concrete. *IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE)*, (2278–1684), 1–6.

Balasko, B., Abonyi, J., & Feil, B. (2005). Fuzzy Clustering and Data Analysis Toolbox. Retrieved from https://nl.mathworks.com/matlabcentral/fileexchange/7486-clustering-toolbox?focused=5062647&tab=function

Bezdek, J. C., & Hathaway, R. J. (2016). VAT: a tool for visual assessment of (cluster) tendency. In *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290)* (pp. 2225–2230). https://doi.org/10.1109/IJCNN.2002.1007487

Breiman, L. (2001). *Random Forests*. https://doi.org/10.1017/CBO9781107415324.004

Busch, M., Franke, S., Müller, A., Wolf, M., Gerth, J., Ott, U., … Stein, G. (2004). Potential cardiovascular risk factors in chronic kidney disease: AGEs, total homocysteine and metabolites, and the C-reactive protein. *Kidney International*, *66*(1), 338–347. https://doi.org/10.1111/j.1523-1755.2004.00736.x

Byoung-Tak, Z. (2001). *Artificial Neural Networks - Supplement to 2001 Bioinformatics lecture on neural nets*. Retrieved from https://bi.snu.ac.kr/Courses/g-ai01/Chapter5-NN.pdf

Chiu, S. L. (1994). Fuzzy model identification based on cluster estimation. *Journal of Interlligent and Fuzzy Systems*, *2*, 267–278.

Fedosov, S. N. (2010). Metabolic signs of vitamin B12 deficiency in humans: Computational model and its implications for diagnostics. *Metabolism: Clinical and Experimental*, *59*(8), 1124–1138. https://doi.org/10.1016/j.metabol.2009.09.036

Fedosov, S. N. (2013). Biochemical markers of vitamin B12 deficiency combined in one diagnostic parameter: The age-dependence and association with cognitive function and blood hemoglobin. *Clinica Chimica Acta*, *422*, 47–53. https://doi.org/10.1016/j.cca.2013.04.002

Frank, J., & Massey, J. (1951). The Kolmogorov-Smirnov Test for Goodness of Fit. *Journal of the American Statistical Association*, *46*(253), 68–78.

Gustafson, D. E., & Kessel, W. C. (1979). Fuzzy clustering with a fuzzy covariance matrix. In *Proc. of IEEE CDC*.

Hannibal, L., Lysne, V., Bjørke-Monsen, A.-L., Behringer, S., Grünert, S. C., Spiekerkoetter, U., … Blom, H. J. (2016). Biomarkers and Algorithms for the Diagnosis of Vitamin B12 Deficiency. *Frontiers in Molecular Biosciences*, *3*(June). https://doi.org/10.3389/fmolb.2016.00027

Havens, T. C., & Bezdek, J. C. (2012). An efficient formulation of the improved visual assessment of cluster tendency (iVAT) algorithm. *IEEE Transactions on Knowledge and Data Engineering*, *24*(5), 813–822. https://doi.org/10.1109/TKDE.2011.33

Henning, B. F., Riezler, R., Tepel, M., Langer, K., Raidt, H., Graefe, U., & Zidek, W. (1999). Evidence of altered homocysteine metabolism in chronic renal failure. *Nephron*, *83*(4), 314–322. https://doi.org/45423

Henning, B. F., Tepel, M., Graefe, U., & Zidek, W. (2000). [Homocysteine and its metabolites in chronic renal insufficiency and the effect of a vitamin replacement]. *Medizinische Klinik (Munich, Germany : 1983)*, *95*(9), 477–481.

International Birch University. (2012). Fuzzy inference systems.

Jain, A. K., Mao, J., & Mohiuddin, K. M. (1996). Artificial neural networks: A tutorial. *Computer*. https://doi.org/10.1109/2.485891

Jang, J.-S. R., Sun, C.-T., & Mizutani, E. (1997). *Neuro-Fuzzy And Soft Computing Jang: a computational approach to learning and machine intelligence*.

Kaymak, U. (2003). Data and Cluster Weighting in Target Selection. In *10th International Fuzzy Systems Association World Congress* (pp. 568–575).

Kaymak, U., & Van Berg, J. Den. (2004). On constructing probabilistic fuzzy classifiers from weighted fuzzy clustering. In *IEEE International Conference on Fuzzy Systems* (Vol. 1, pp. 395–400). https://doi.org/10.1109/FUZZY.2004.1375757

Levey, A. S., Stevens, L. A., Schmid, C. H., Zhang, Y. L., Castro, A. F., Feldman, H. I., … CKD-EPI (Chronic Kidney Disease Epidemiology Collaboration). (2009). A new equation to estimate glomerular filtration rate. *Annals of Internal Medicine*, *150*(9), 604–12. https://doi.org/10.7326/0003-4819-150-9-200905050-00006

Mathworks. (2017). Multilayer Neural Network Architecture. Retrieved from http://nl.mathworks.com/help/nnet/ug/multilayer-neural-network-architecture.html

MedlinePlus. (2015). Glomerular filtration rate. Retrieved from https://medlineplus.gov/ency/article/007305.htm

Naurath, H. J., Joosten, E., Riezler, R., Stabler, S. P., Allen, R. H., & Lindenbaum, J. (1995). Effects of vitamin B12, folate, and vitamin B6 supplements in elderly people with normal serum vitamin concentrations. *Lancet (London, England)*, *346*(8967), 85–9. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/7603218

Oliveira, J. V. De, & Pedrycz, W. (2007). *Advances in Fuzzy Clustering and its Applications*. *Wiley*.

Ortigueira, M. D., & Tribolet, J. M. (1984). Global versus local minimization in least-squares AR spectral estimation. *Signal Processing*, *7*(3), 267–281. https://doi.org/10.1016/0165-1684(84)90004-5

Palacios, G., Sola, R., Barrios, L., Pietrzik, K., Castillo, M. J., & Gonzalez-Gross, M. (2013). Algorithm for the early diagnosis of vitamin B12 deficiency in elderly people. *Nutr Hosp*, *28*(5), 1447–1452.

https://doi.org/10.3305/nh.2013.28.5.6821

Popescu, M., Keller, J. M., Bezdek, J. C., & Havens, T. (2011). Correlation cluster validity. In *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics* (pp. 2531–2536). https://doi.org/10.1109/ICSMC.2011.6084057

Popescu, M., Member, S., Bezdek, J. C., Fellow, L., Havens, T. C., & Keller, J. M. (2013). A Cluster Validity Framework Based on Induced Partition Dissimilarity, *43*(1), 308–320.

Princeton. (2007). Fuzzy inference systems (Mamdani). Retrieved from http://www.cs.princeton.edu/courses/archive/fall07/cos436/HIDDEN/Knapp/fuzzy004.htm

Risch, M., Meier, D. W., Sakem, B., Medina Escobar, P., Risch, C., Nydegger, U., & Risch, L. (2015). Vitamin B12 and folate levels in healthy Swiss senior citizens: a prospective study evaluating reference intervals and decision limits. *BMC Geriatrics*, *15*(1), 82. https://doi.org/10.1186/s12877-015-0060-x

Schwarz, J., Morstadt, E., Dura, A., Wintgens, K. F., Hartmann, K., Armbruster, F. P., & Dschietzig, T. (2015). Biochemical identification of vitamin B12 deficiency in a medical office. *Clinical Laboratory*, *61*(7), 687–692. https://doi.org/10.7754/Clin.Lab.2014.141219

Solomon, L. R. (2005). Cobalamin-responsive disorders in the ambulatory care setting: Unreliability of cobalamin, methylmalonic acid, and homocysteine testing. *The American Society of Hematology*, *105*(3), 978–985. https://doi.org/10.1182/blood-2004-04-1641

Stein, G., Muller, A., Busch, M., Fleck, C., & Sperschneider, H. (2001). Homocysteine, its metabolites, and B-group vitamins in renal transplant patients. *Kidney Int Suppl*, *78*, S262-5. https://doi.org/10.1046/j.1523-1755.2001.59780262.x

University of Nevada Reno. (2003). *Mathematical Methods for Computer Vision - Neural Networks*. Retrieved from https://www.cse.unr.edu/~bebis/MathMethods/NNs/lecture.pdf

Van Der Put, N. M. J., Van Straaten, H. W. M., Trijbels, F. J. M., & Blom, H. J. (2001). Folate, Homocysteine and Neural Tube Defects: An Overview. *Experimental Biology and Medicine*, *226*(4), 243–270. https://doi.org/10.1177/153537020122600402

Wilbik, A., Loon, S. Van, Boer, A., & Kaymak, U. (2016). Fuzzy modeling for Vitamin B12 Deficiency, *610*, 462–471. https://doi.org/10.1007/978-3-319-40596-4

Yohannes, Y., & Hoddinott, J. (1999). *Classification and regression trees: an introduction*. International Food Policy Research Institute.

7 Appendix

7.1 Error values of models

Error values of fuzzy c-means

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	0,3870	97,63	21938	0,1978
Model 2	2MCA1	0,3797	97,17	21817	0,2022
Model 3	2MCA2	0,3812	96,87	21750	0,2047
Model 4	2MCAR	0,3667	95,34	21393	0,2177
Model 5	2MCA1, 2MCA2	0,3870	97,52	21878	0,2000
Model 6	2MCA1, 2MCAR	0,3803	97,21	21853	0,2009
Model 7	2MCA2, 2MCAR	0,3820	97,07	21873	0,2002
Benchmark	NONE	0,3654	95,19	21371	0,2185

Error values of Gustafson Kessel standard

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	0,4052	96,49	21808	0,2025
Model 2	2MCA1	0,3988	95,40	21524	0,2129
Model 3	2MCA2	0,3997	95,61	21560	0,2116
Model 4	2MCAR	0,3989	95,28	21519	0,2131
Model 5	2MCA1, 2MCA2	0,4002	95,78	21623	0,2093
Model 6	2MCA1, 2MCAR	0,4009	95,86	21654	0,2082
Model 7	2MCA2, 2MCAR	0,4016	95,90	21502	0,2137
Benchmark	NONE	0,3972	94,92	21303	0,2210

Error values of Gustafson Kessel changed clusters to 4

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	0,408522	96,92753	21818	0,202195
Model 2	2MCA1	0,40267	96,05904	21676,64	0,207364
Model 3	2MCA2	0,398905	95,42403	21494,74	0,214015
Model 4	2MCAR	0,397088	94,92579	21300,96	0,221101
Model 5	2MCA1, 2MCA2	0,396096	94,86299	21365,17	0,218753
Model 6	2MCA1, 2MCAR	0,399776	95,46877	21424,84	0,216571
Model 7	2MCA2, 2MCAR	0,402542	96,0332	21708	0,206217
Benchmark	NONE	0,396528	94,83043	21338,53	0,219727

Error values of Gustafson Kessel changed rho

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	0,405092	96,48114	21801,07	0,202814
Model 2	2MCA1	0,398709	95,38201	21516,37	0,213224
Model 3	2MCA2	0,399704	95,60335	21555,68	0,211787
Model 4	2MCAR	0,398856	95,27667	21516,07	0,213235
Model 5	2MCA1, 2MCA2	0,400151	95,76669	21619,52	0,209453
Model 6	2MCA1, 2MCAR	0,400852	95,84071	21645,81	0,208491
Model 7	2MCA2, 2MCAR	0,401584	95,8923	21497,81	0,213903
Benchmark	NONE	0,397289	94,9346	21309,53	0,220788

Error values of Gustafson Kessel changed alpha to 4

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	0,40515	96,49267	21808,4	0,202546
Model 2	2MCA1	0,398754	95,39503	21524,1	0,212942
Model 3	2MCA2	0,399705	95,60749	21559,8	0,211636
Model 4	2MCAR	0,398872	95,28076	21519,33	0,213116
Model 5	2MCA1, 2MCA2	0,400175	95,77529	21622,9	0,209329
Model 6	2MCA1, 2MCAR	0,400927	95,85583	21653,82	0,208198
Model 7	2MCA2, 2MCAR	0,401625	95,90221	21502,03	0,213749
Benchmark	NONE	0,397232	94,92181	21302,79	0,221034

Error values of subtractive clustering

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	0,4183	98,60	22384	0,1815
Model 2	2MCA1	0,4164	98,36	22278	0,1854
Model 3	2MCA2	0,4178	98,64	22495	0,1774
Model 4	2MCAR	0,4026	95,84	21580	0,2109
Model 5	2MCA1, 2MCA2	0,4150	98,12	22151	0,1900
Model 6	2MCA1, 2MCAR	0,4178	98,69	22531	0,1761
Model 7	2MCA2, 2MCAR	0,4260	100,05	23329	0,1469
Benchmark	NONE	0,3981	95,02	21314	0,2206

Error values of partitioning without MMA

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	0,4138	97,95	22006	0,1953
Model 2	2MCA1	0,4098	97,19	21883	0,1998
Model 3	2MCA2	0,4158	98,25	22233	0,1870
Model 4	2MCAR	0,4019	95,78	21577	0,2110
Model 5	2MCA1, 2MCA2	0,4156	98,29	22163	0,1896
Model 6	2MCA1, 2MCAR	0,4149	98,24	22224	0,1874
Model 7	2MCA2, 2MCAR	0,4161	98,32	22194	0,1885
Benchmark	NONE	0,4044	96,36	22055	0,1935

Error values of partitioning with MMA

	ADDED INPUT	MAPE	MAE	MSE	R-squared	Highest MMA of clusters
Model 1	ALL	0,4154	98,20	22032	0,1944	349
Model 2	2MCA1	0,4155	98,32	22266	0,1858	664
Model 3	2MCA2	0,4169	98,30	22162	0,1896	681
Model 4	2MCAR	0,4094	97,17	22775	0,1672	653
Model 5	2MCA1, 2MCA2	0,4146	98,12	22188	0,1887	598
Model 6	2MCA1, 2MCAR	0,4151	98,30	22270	0,1857	707
Model 7	2MCA2, 2MCAR	0,4157	98,00	22127	0,1909	707
Benchmark	NONE	0,3988	95,23	21507	0,2136	301

Error values of biomarker partition

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	0,4147	98,16	22207	0,1880
Model 2	2MCA1	0,4161	98,35	22178	0,1890
Model 3	2MCA2	0,4166	98,22	22132	0,1907
Model 4	2MCAR	0,4042	96,27	21677	0,2074
Model 5	2MCA1, 2MCA2	0,4180	98,60	22232	0,1871
Model 6	2MCA1, 2MCAR	0,4196	99,68	29302	-0,0714
Model 7	2MCA2, 2MCAR	0,4278	100,39	23436	0,1430
Benchmark	NONE	0,4030	96,13	21667	0,2077

Error values of equal distance partition 3

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	1,0701	184,59	5,36E+08	-19597,6
Model 2	2MCA1	0,4197	98,66	40208	-0,4703
Model 3	2MCA2	0,4199	98,56	40503	-0,4810
Model 4	2MCAR	0,4121	97,66	22165	0,1895
Model 5	2MCA1, 2MCA2	0,5312	113,51	16513618	-602,84
Model 6	2MCA1, 2MCAR	0,4210	98,83	43425	-0,5879
Model 7	2MCA2, 2MCAR	0,4197	98,60	57860	-1,1157
Benchmark	NONE	0,4112	97,49	22076	0,1927

Error values of equal distance partition 5

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	0,4198	101,24	118086	-3,3180
Model 2	2MCA1	0,4000	95,96	27877	-0,0194
Model 3	2MCA2	0,4019	96,39	27232	0,0042
Model 4	2MCAR	0,3978	95,20	21621	0,2094
Model 5	2MCA1, 2MCA2	0,4160	100,05	92907	-2,3973
Model 6	2MCA1, 2MCAR	0,4047	97,23	47305	-0,7298
Model 7	2MCA2, 2MCAR	0,4035	96,84	28095	-0,0273
Benchmark	NONE	0,3971	95,00	21405	0,2173

Error values of weighted fuzzy c-means

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	0,4133	97,85	22172	0,1892
Model 2	2MCA1	0,4160	98,35	22225	0,1873
Model 3	2MCA2	0,4160	98,32	22224	0,1874
Model 4	2MCAR	0,4049	96,48	21803	0,2028
Model 5	2MCA1, 2MCA2	0,4142	98,06	22169	0,1894
Model 6	2MCA1, 2MCAR	0,4154	98,30	22236	0,1869
Model 7	2MCA2, 2MCAR	0,4165	98,44	22272	0,1856
Benchmark	NONE	0,4050	96,47	21799	0,2029

Error values of weighted Gustafson-Kessel standard

Weighted	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	0,403782	96,31505	21766,53	0,204077
Model 2	2MCA1	0,397277	95,17513	21465,72	0,215076
Model 3	2MCA2	0,401132	95,7225	21544,77	0,212186
Model 4	2MCAR	0,397066	94,96759	21420,65	0,216724
Model 5	2MCA1, 2MCA2	0,399466	95,63227	21497,22	0,213925
Model 6	2MCA1, 2MCAR	0,401133	95,8482	21637,95	0,208778
Model 7	2MCA2, 2MCAR	0,401613	95,89385	21504,88	0,213645
Benchmark	NONE	0,395762	94,73196	21250,2	0,222957

Error values of weighted Gustafson Kessel 4 clusters

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	0,413725	98,02308	22175,22	0,189132
Model 2	2MCA1	0,395993	94,85803	21291,05	0,221463
Model 3	2MCA2	0,396791	94,94575	21374,56	0,21841
Model 4	2MCAR	0,397476	94,98605	21363,76	0,218805
Model 5	2MCA1, 2MCA2	0,397109	95,09532	21491,18	0,214145
Model 6	2MCA1, 2MCAR	0,400964	95,63841	21422,39	0,216661
Model 7	2MCA2, 2MCAR	0,400862	95,60204	21499,41	0,213844
Benchmark	NONE	0,396191	94,80039	21325,84	0,220191

Error values of weighted Gustafson Kessel changed rho

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	0,403745	96,3008	21682,2	0,207161
Model 2	2MCA1	0,398033	95,19699	21495,04	0,214004
Model 3	2MCA2	0,396931	95,02637	21351,27	0,219261
Model 4	2MCAR	0,397428	95,06012	21397,03	0,217588
Model 5	2MCA1, 2MCA2	0,400286	95,74617	21574,51	0,211098
Model 6	2MCA1, 2MCAR	0,399307	95,57357	21584,62	0,210729
Model 7	2MCA2, 2MCAR	0,402062	95,90839	21560,54	0,211609
Benchmark	NONE	0,395534	94,70431	21231,96	0,223624

Error values of weighted Gustafson Kessel changed alpha = 4

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	0,403786	96,30451	21683,23	0,207123
Model 2	2MCA1	0,398023	95,19916	21498,55	0,213876
Model 3	2MCA2	0,396949	95,03251	21354,34	0,219149
Model 4	2MCAR	0,397442	95,06249	21400,32	0,217468
Model 5	2MCA1, 2MCA2	0,400343	95,76185	21582,76	0,210797
Model 6	2MCA1, 2MCAR	0,399382	95,59163	21596,65	0,210289
Model 7	2MCA2, 2MCAR	0,402142	95,92769	21568,14	0,211331
Benchmark	NONE	0,395468	94,68708	21225,12	0,223874

Error values of balanced subtractive clustering for training

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	0,604619	123,6497	25799,83	0,056594
Model 2	2MCA1	0,607807	124,1284	25941,27	0,051422
Model 3	2MCA2	0,608448	124,1733	25865,23	0,054202
Model 4	2MCAR	0,591934	121,6341	25375,75	0,072101
Model 5	2MCA1, 2MCA2	0,60854	124,3836	26067,71	0,046798
Model 6	2MCA1, 2MCAR	0,605303	123,8104	25854,91	0,05458
Model 7	2MCA2, 2MCAR	0,611	124,7559	26401,95	0,034576
Benchmark	NONE	0,591845	121,5381	25320,69	0,074114

Error values of totally balanced subtractive clustering for training and cluster selection

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	0,602345	123,477	25847,06	0,054867
Model 2	2MCA1	0,605525	123,7208	25802,55	0,056494
Model 3	2MCA2	0,607805	124,1141	25851,53	0,054703
Model 4	2MCAR	0,59419	122,234	25560,4	0,065349
Model 5	2MCA1, 2MCA2	0,606255	123,9924	26231,27	0,040818
Model 6	2MCA1, 2MCAR	0,606507	124,1703	25976,43	0,050136
Model 7	2MCA2, 2MCAR	0,607755	124,3208	26207,64	0,041682
Benchmark	NONE	0,591675	121,5867	25309,05	0,07454

Error values of 0-order Takagi-Sugeno equal distance partitioning with weight adapted training

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	1,227616	204,7149	7,06E+08	-25819
Model 2	2MCA1	0,471678	104,7569	109186,3	-2,99255
Model 3	2MCA2	0,47028	104,4431	108840,6	-2,9799
Model 4	2MCAR	0,454597	102,5283	48486,67	-0,77298
Model 5	2MCA1, 2MCA2	0,584581	119,7722	17363694	-633,927
Model 6	2MCA1, 2MCAR	0,484616	106,5252	266882,3	-8,75892
Model 7	2MCA2, 2MCAR	0,475999	105,3308	122278,3	-3,47127
Benchmark	NONE	0,454361	102,4497	85370,46	-2,12169

Error values of FCM with age added

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	1,200551	941,168	20620458	-753,015434
Model 2	2MCA1	0,812606	508,5861	16487216	-601,877763
Model 3	2MCA2	0,380114	96,67864	21772,44	0,2038606
Model 4	2MCAR	0,382249	95,80167	21517,68	0,213176437
Model 5	2MCA1, 2MCA2	1,253226	685,1238	11019977	-401,960619
Model 6	2MCA1, 2MCAR	0,568418	129,0112	194784	-6,12254619
Model 7	2MCA2, 2MCAR	0,490789	175,4735	3285147	-119,125914
Benchmark	NONE	0,367918	95,37465	21378,53	0,218264694

Error values of FCM with homocysteine added

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	3,383272	16809,85	2,28E+10	-829163,6463
Model 2	2MCA1	0,541879	340,4019	12218716	-443,531998
Model 3	2MCA2	0,784258	1113,16	4,3E+08	-15631,44931
Model 4	2MCAR	0,375802	98,1537	22830,98	0,170012592
Model 5	2MCA1, 2MCA2	1,97495	62618,23	2,56E+11	-9298964,035
Model 6	2MCA1, 2MCAR	0,586283	1502,235	2,5E+09	-91125,78384
Model 7	2MCA2, 2MCAR	0,541354	110,2496	67823,85	-1,466551822
Benchmark	NONE	0,375893	98,34017	22882,73	0,16813196

Error values of FCM with haemoglobin added

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	1,497956	970,2364	20225932,12	-787,75678
Model 2	2MCA1	1,033986	1448,382	210832733,6	-8220,8225
Model 3	2MCA2	0,387699	95,94749	21697,03319	0,1539594
Model 4	2MCAR	0,364095	94,41289	21001,2485	0,1810932
Model 5	2MCA1, 2MCA2	0,999672	1071,241	28251795,77	-1100,5756
Model 6	2MCA1, 2MCAR	0,400997	100,7039	31513,18203	-0,2288404
Model 7	2MCA2, 2MCAR	1,074007	427,4913	11024684,38	-428,8528
Benchmark	NONE	0,3627	94,12413	20914,07272	0,1844925

Error values of FCM with all four biomarkers added

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	2,97	350018,80	219273557621588,00	-5628798283
Model 2	2MCA1	0,87	1597,08	622299485,29	-15973,55851
Model 3	2MCA2	1,42	18281,16	114867462494,67	-2948670,891
Model 4	2MCAR	0,44	120,88	34135,67	0,123730061
Model 5	2MCA1, 2MCA2	1,26	424075,67	375216979459710,00	-9631898679
Model 6	2MCA1, 2MCAR	1,64	1766,72	5292454107,24	-135857,4087
Model 7	2MCA2, 2MCAR	1,11	10413,05	25235664388,04	-647803,8058
Benchmark	NONE	0,43	117,65	32122,12	0,17541845

Error values of Neural networks fitnet 10.10.10

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	0,400793	96,38825	21683,17	0,207125
Model 2	2MCA1	0,404232	95,92329	21861,9	0,20059
Model 3	2MCA2	0,393307	94,90602	21722,53	0,205686
Model 4	2MCAR	0,394854	95,34265	22279,39	0,185323
Model 5	2MCA1, 2MCA2	0,399499	95,31646	21410,85	0,217083
Model 6	2MCA1, 2MCAR	0,397707	94,74534	20995,27	0,232279
Model 7	2MCA2, 2MCAR	0,393401	95,1642	22242,34	0,186678
Benchmark	NONE	0,40163	96,79567	22847,83	0,164538

Error values of fitnet 30.30.15

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	0,426333	99,40302	22163	0,18958
Model 2	2MCA1	0,404443	96,27261	21920,68	0,19844
Model 3	2MCA2	0,406238	95,14658	20714,82	0,242534
Model 4	2MCAR	0,413531	99,63862	23799,81	0,129727
Model 5	2MCA1, 2MCA2	0,41348	100,2851	23938,6	0,124652
Model 6	2MCA1, 2MCAR	0,410205	97,37839	22483,28	0,177868
Model 7	2MCA2, 2MCAR	0,402845	96,60337	22233,55	0,187
Benchmark	NONE	0,40598	97,1386	22399,27	0,18094

Error values of fitnet 10.10.10.10

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	0,396947	96,22204	22312,37	0,184117
Model 2	2MCA1	0,394987	94,93228	21265,15	0,222411
Model 3	2MCA2	0,389998	94,90874	21835,44	0,201557
Model 4	2MCAR	0,400145	94,08151	20645,33	0,245075
Model 5	2MCA1, 2MCA2	0,412023	97,9067	22283,89	0,185159
Model 6	2MCA1, 2MCAR	0,410544	95,70778	21066,4	0,229678
Model 7	2MCA2, 2MCAR	0,410248	96,22284	21205,2	0,224603
Benchmark	NONE	0,40617	95,64808	21078,15	0,229249

Error values of cascadeforwardnet 10.10.10

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	0,402694	95,37744	21567,42	0,211358
Model 2	2MCA1	0,408269	95,48211	21246,38	0,223097
Model 3	2MCA2	0,401534	95,6791	21588,81	0,210575
Model 4	2MCAR	0,406368	97,62637	22837,23	0,164925
Model 5	2MCA1, 2MCA2	0,402136	95,33331	21709,72	0,206154
Model 6	2MCA1, 2MCAR	0,399493	96,26071	22472,39	0,178266
Model 7	2MCA2, 2MCAR	0,405569	96,4325	21856,14	0,2008
Benchmark	NONE	0,39197	95,38446	21991,65	0,195845

Error values of cascadeforwardnet 30.30.15

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	0,407232	96,58483	21810,54	0,202468
Model 2	2MCA1	0,403539	95,78964	21132,12	0,227275
Model 3	2MCA2	0,403012	98,12062	23501,29	0,140643
Model 4	2MCAR	0,403894	96,75315	22073,33	0,192858
Model 5	2MCA1, 2MCA2	0,396017	97,58456	22872,07	0,163651
Model 6	2MCA1, 2MCAR	0,414344	96,81623	21401,14	0,217438
Model 7	2MCA2, 2MCAR	0,408285	97,12264	22043,15	0,193962
Benchmark	NONE	0,401782	97,00776	22673,89	0,170898

Error values of cascadeforwardnet 10.10.10.10

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	0,406321	96,88223	22019,71	0,194819
Model 2	2MCA1	0,396372	96,56192	22282,19	0,185221
Model 3	2MCA2	0,399471	96,3116	21868,01	0,200366
Model 4	2MCAR	0,40329	96,03297	21421,67	0,216687
Model 5	2MCA1, 2MCA2	0,401887	96,45792	22048,73	0,193758
Model 6	2MCA1, 2MCAR	0,408511	95,82043	20853,01	0,237481
Model 7	2MCA2, 2MCAR	0,403226	96,49789	21871,1	0,200253
Benchmark	NONE	0,39451	95,5267	21812,69	0,202389

Error values of balanced fitnet 10.10.10

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	0,563314	115,1532	23057,62	0,196213
Model 2	2MCA1	0,576697	117,9486	23977,86	0,164133
Model 3	2MCA2	0,577516	117,6721	23553,64	0,178921
Model 4	2MCAR	0,577956	118,4153	24153,78	0,158001
Model 5	2MCA1, 2MCA2	0,569428	116,2075	23377,92	0,185047
Model 6	2MCA1, 2MCAR	0,588127	119,5631	24122,26	0,159099
Model 7	2MCA2, 2MCAR	0,564424	115,4609	23720,63	0,1731
Benchmark	NONE	0,579808	118,8968	24350,35	0,151148

Error values of balanced fitnet 30.30.15

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	0,556641	112,781	22244,56	0,224556
Model 2	2MCA1	0,563382	114,6781	23015,94	0,197666
Model 3	2MCA2	0,557756	113,7703	22626,75	0,211233
Model 4	2MCAR	0,572608	117,3412	23970,8	0,164379
Model 5	2MCA1, 2MCA2	0,552739	112,1228	22355,99	0,220671
Model 6	2MCA1, 2MCAR	0,558909	114,4547	23443,8	0,18275
Model 7	2MCA2, 2MCAR	0,556051	113,4847	22851,42	0,203401
Benchmark	NONE	0,579323	118,5916	24224,98	0,155518

Error values of balanced fitnet 10.10.10.10

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	0,593861	120,1539	24660,78	0,140327

Model 2	2MCA1	0,567518	115,6267	23083,74	0,195302
Model 3	2MCA2	0,576419	117,6038	23734,25	0,172625
Model 4	2MCAR	0,572016	117,3314	23936,9	0,165561
Model 5	2MCA1, 2MCA2	0,575198	116,6529	23642,41	0,175827
Model 6	2MCA1, 2MCAR	0,573717	117,5712	23770,76	0,171353
Model 7	2MCA2, 2MCAR	0,5777	117,4716	23605,57	0,177111
Benchmark	NONE	0,573739	117,9637	23944,17	0,165308

Error values of balanced cascadeforwardnet 10.10.10.

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	0,570009	116,344	23596,67	0,177421
Model 2	2MCA1	0,5775	118,937	24336,93	0,151616
Model 3	2MCA2	0,570523	116,8611	23531,75	0,179684
Model 4	2MCAR	0,581001	118,7304	24148,86	0,158172
Model 5	2MCA1, 2MCA2	0,57165	117,1764	23511,3	0,180397
Model 6	2MCA1, 2MCAR	0,571524	117,5064	23885,83	0,167341
Model 7	2MCA2, 2MCAR	0,576834	117,3958	23591,86	0,177589
Benchmark	NONE	0,578451	118,7847	24751,31	0,137171

Error values of balanced cascadeforwardnet 30.30.15

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	0,5603	113,9886	22547,67	0,213989
Model 2	2MCA1	0,570089	116,2932	23236,07	0,189992
Model 3	2MCA2	0,57269	117,1936	23778,98	0,171066
Model 4	2MCAR	0,566697	115,6214	23368,95	0,18536
Model 5	2MCA1, 2MCA2	0,557119	113,3067	22330,66	0,221555
Model 6	2MCA1, 2MCAR	0,566142	115,6278	23398,83	0,184318
Model 7	2MCA2, 2MCAR	0,546988	112,0847	22285,62	0,223125
Benchmark	NONE	0,563742	115,8844	23411,15	0,183888

Error values of balanced cascadeforwardnet 10.10.10.10

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	0,565487	116,3681	23531	0,179711
Model 2	2MCA1	0,575922	117,5697	23701,65	0,173762
Model 3	2MCA2	0,578107	118,1703	24393,32	0,14965
Model 4	2MCAR	0,579301	118,9325	24344,37	0,151356
Model 5	2MCA1, 2MCA2	0,560682	114,4758	22730,43	0,207618
Model 6	2MCA1, 2MCAR	0,569765	116,8242	24236,76	0,155108
Model 7	2MCA2, 2MCAR	0,563163	115,6861	23220,89	0,190521
Benchmark	NONE	0,583234	119,4545	24308,72	0,152599

Error values of bootstrap aggregated decision trees 100 trees

	ADDED INPUT	MAPE	MAE	MSE	R-squared
Model 1	ALL	0,4988	95,18	21078	0,2292

Model 2	2MCA1	0,4991	95,60	21411	0,2171
Model 3	2MCA2	0,5013	96,13	21546	0,2121
Model 4	2MCAR	0,5026	96,04	21684	0,2071
Model 5	2MCA1, 2MCA2	0,5002	95,69	21289	0,2215
Model 6	2MCA1, 2MCAR	0,4996	95,42	21277	0,2220
Model 7	2MCA2, 2MCAR	0,5009	95,74	21371	0,2185
Benchmark	NONE	0,5105	98,46	22607	0,1733

Error values of bootstrap aggregated decision trees 1000 trees

	ADDED INPUT	**MAPE**	**MAE**	**MSE**	**R-squared**
Model 1	ALL	0,498052	94,95396	21022,57	0,231285
Model 2	2MCA1	0,498568	95,4209	21356,72	0,219066
Model 3	2MCA2	0,500918	95,98996	21479,38	0,21458
Model 4	2MCAR	0,501951	95,88719	21645,53	0,208505
Model 5	2MCA1, 2MCA2	0,499908	95,5759	21252,82	0,222865
Model 6	2MCA1, 2MCAR	0,498842	95,21737	21226,13	0,223841
Model 7	2MCA2, 2MCAR	0,500173	95,57917	21321,2	0,220365
Benchmark	NONE	0,510715	98,42121	22562,1	0,174989

Error values of bootstrap aggregated decision trees, 100 balanced trees

	ADDED INPUT	**MAPE**	**MAE**	**MSE**	**R-squared**
Model 1	ALL	0,6016	108,97	23717	0,1328
Model 2	2MCA1	0,6157	111,83	24867	0,0907
Model 3	2MCA2	0,6165	112,27	25002	0,0858
Model 4	2MCAR	0,6209	112,56	25139	0,0808
Model 5	2MCA1, 2MCA2	0,6034	109,98	24406	0,1076
Model 6	2MCA1, 2MCAR	0,6028	109,67	24259	0,1129
Model 7	2MCA2, 2MCAR	0,6037	109,88	24237	0,1137
Benchmark	NONE	0,6375	118,21	28766	-0,0519

7.2 Histograms of errors

Histogram of errors, Gustafson-Kessel

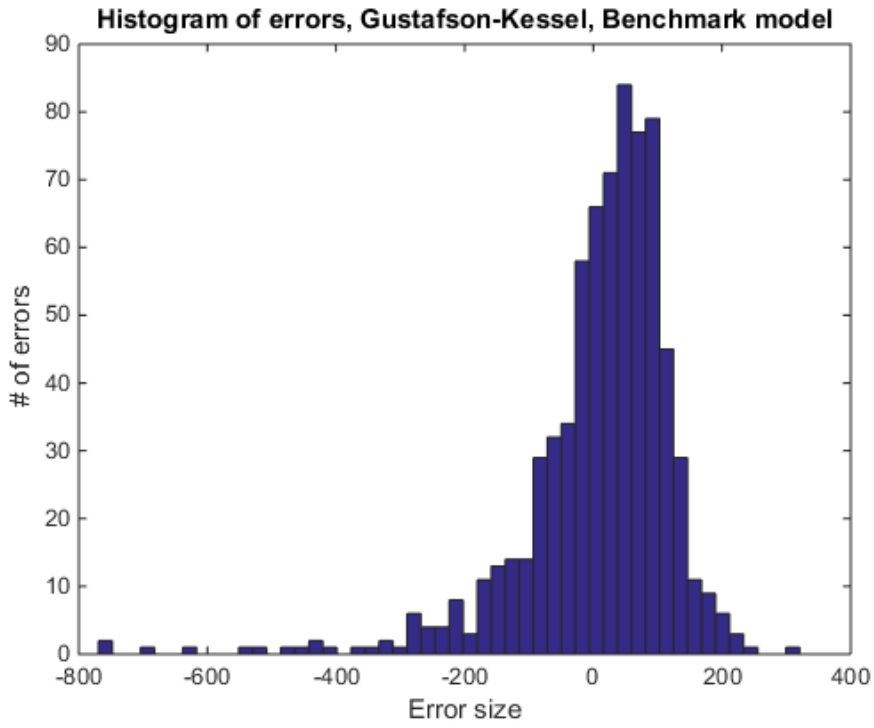

Figure 47: Histogram of errors, GK, Benchmark model

Histogram of errors, Subtractive clustering

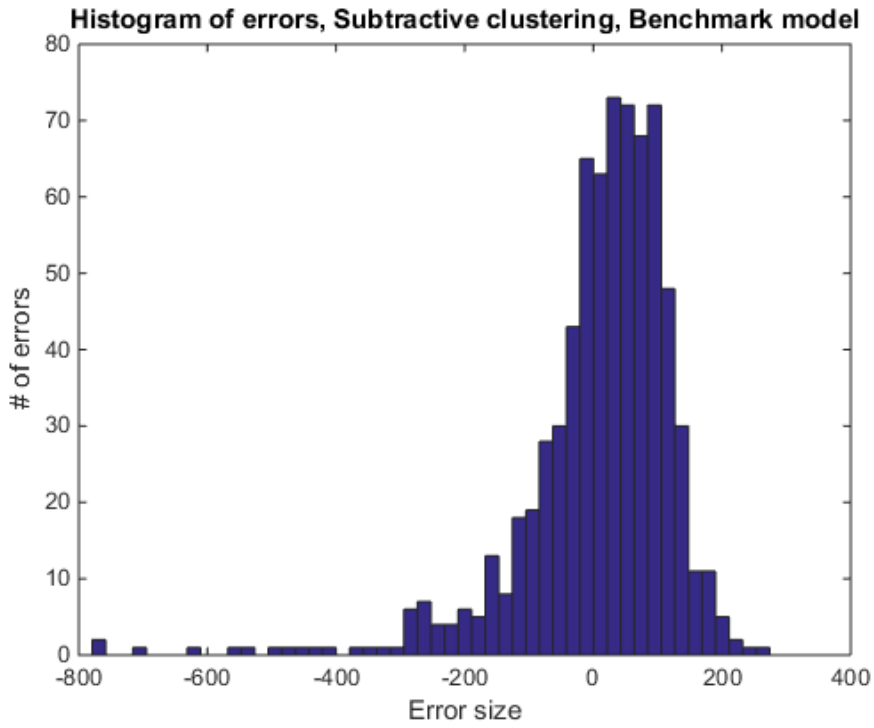

Figure 48: Histogram of errors, Subtractive clustering, Benchmark model

Histogram of errors, Maximum size partition with MMA

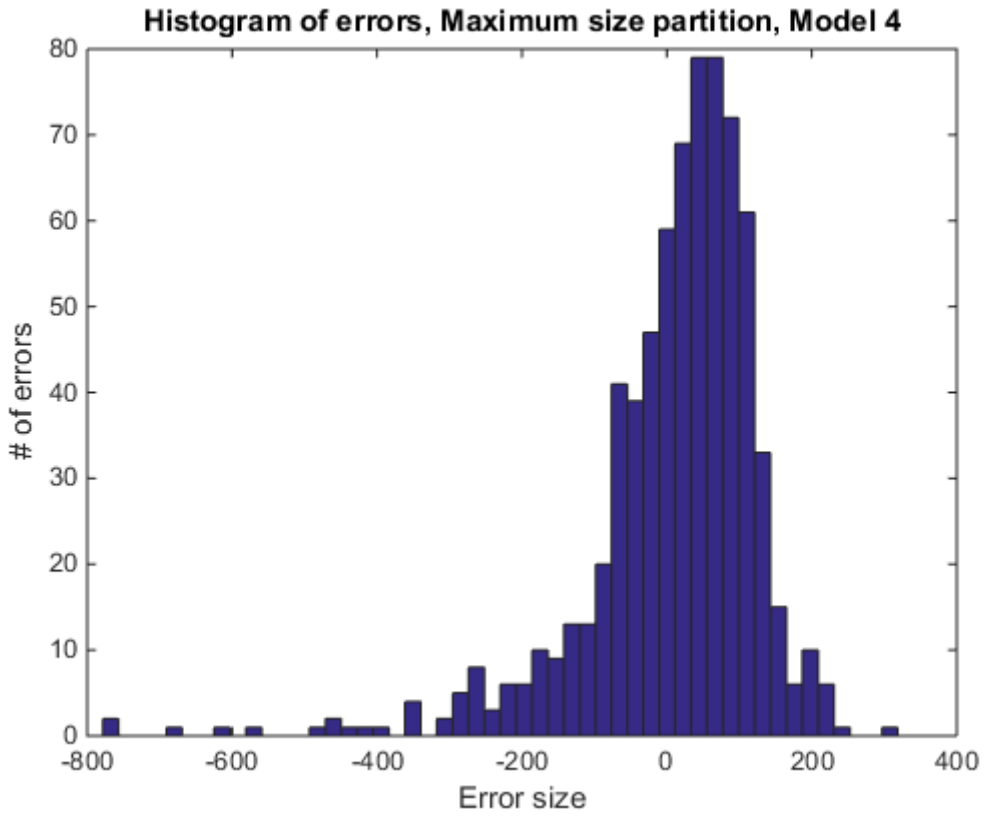

Figure 49: Histogram of errors, Maximum size partition without MMA, Model 4

Histogram of errors, Maximum size partition without MMA

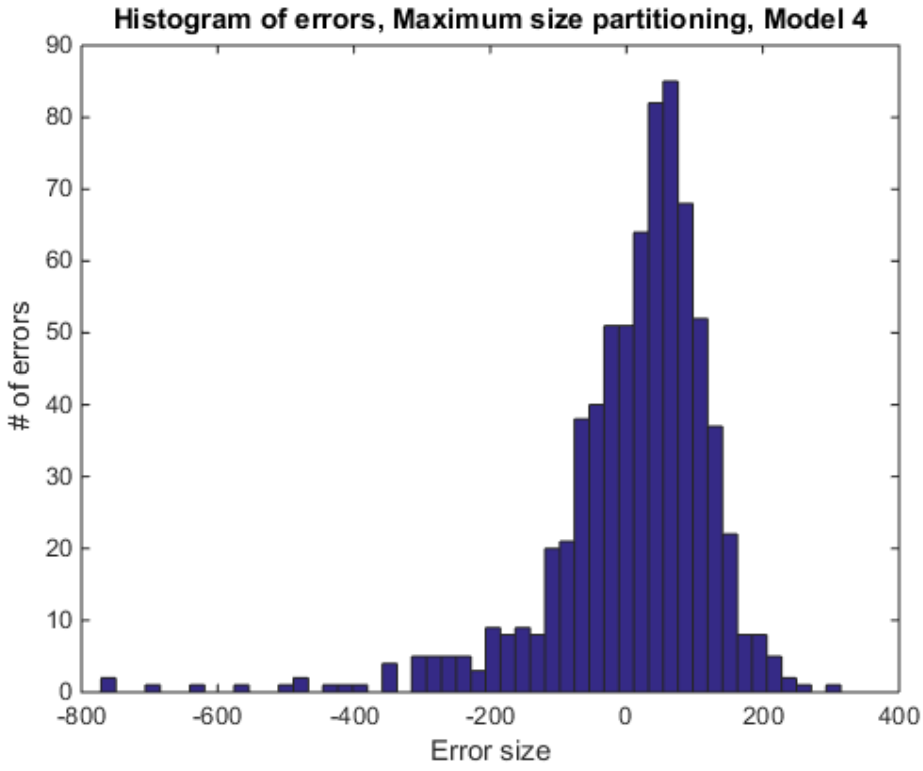

Figure 50: Histogram of errors, Maximum size partitioning with MMA, Model 4

Histogram of errors, Biomarker partition, benchmark model

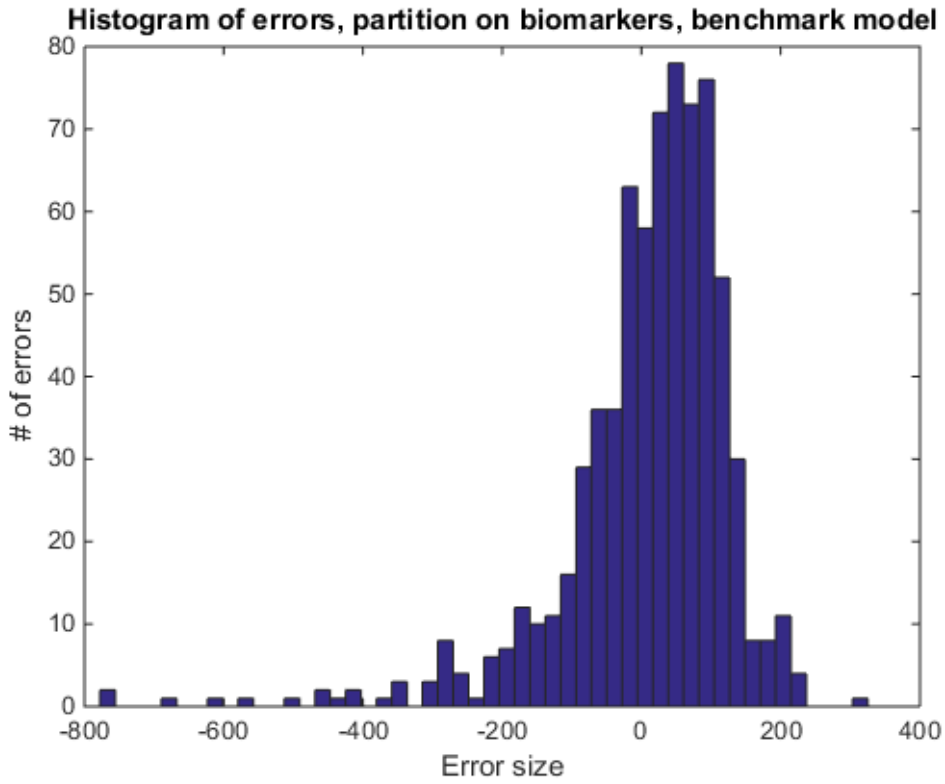

Figure 51: Histogram of errors, Biomarker partition, benchmark model

Histogram of errors, Equal distance partition 3, benchmark model

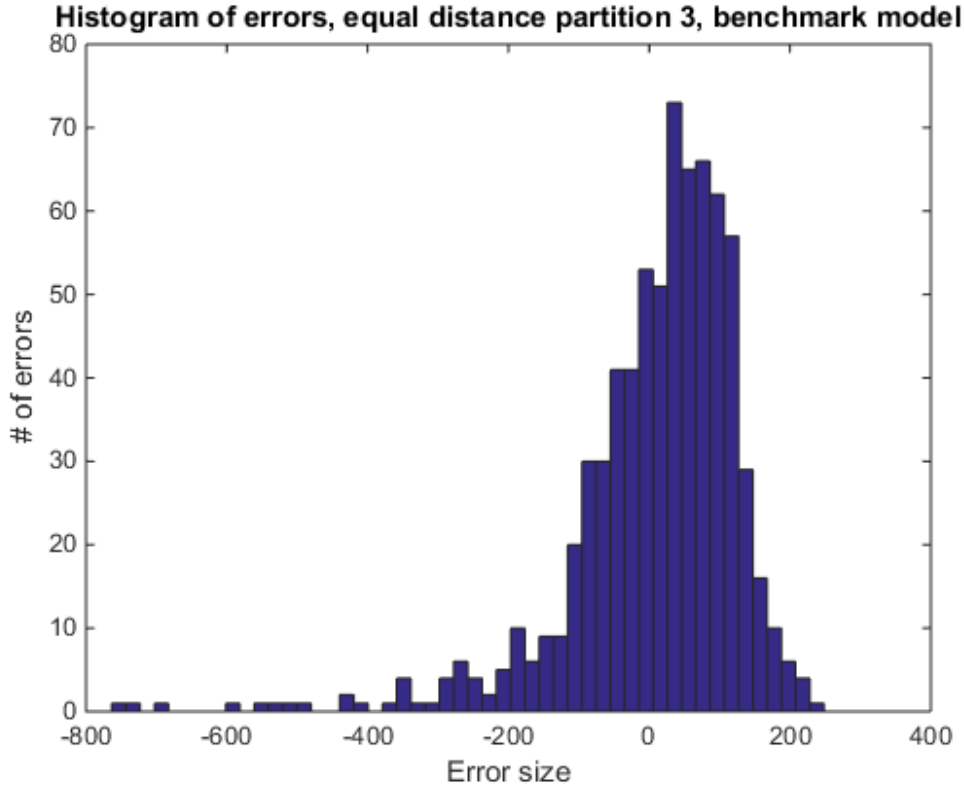

Figure 52: Histogram of errors, equal distance partition 3, Benchmark model

Histogram of errors, Equal distance partition 5, Benchmark model

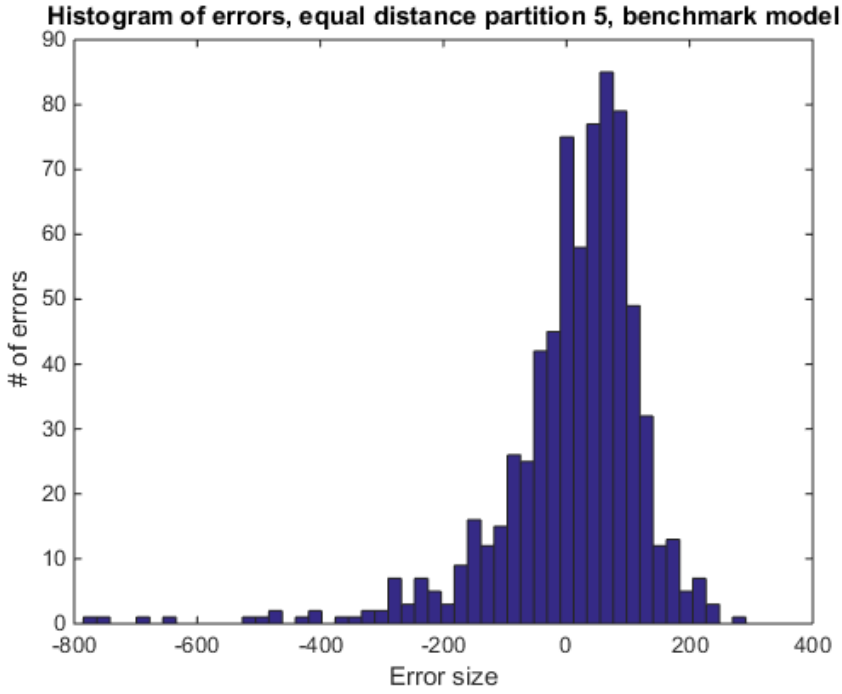

Figure 53: Histogram of errors, equal distance partition 5, Benchmark model

Histogram of errors, Weighted fuzzy c-means, benchmark model

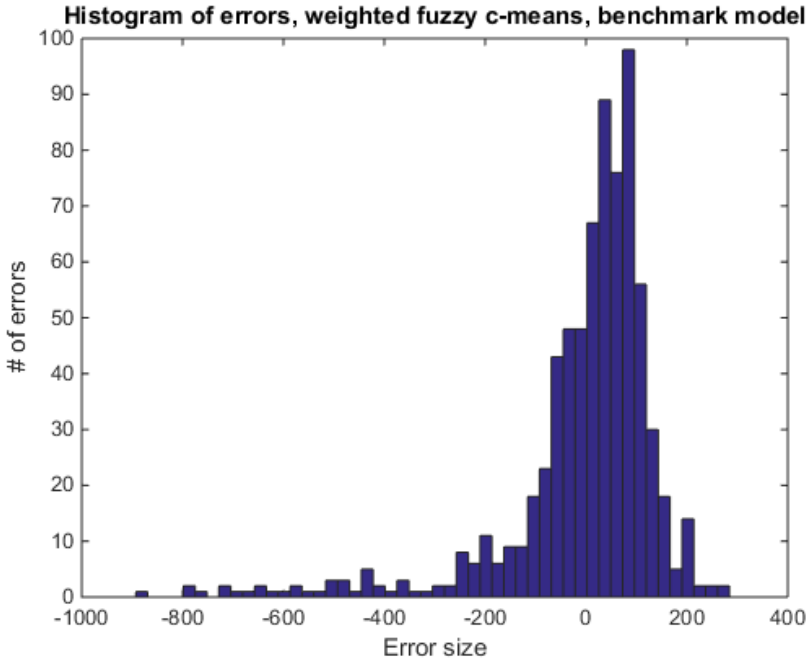

Figure 54: Histogram of erros, weighted fuzzy c-means, benchmark model

Histogram of errors, weighted GK, benchmark model

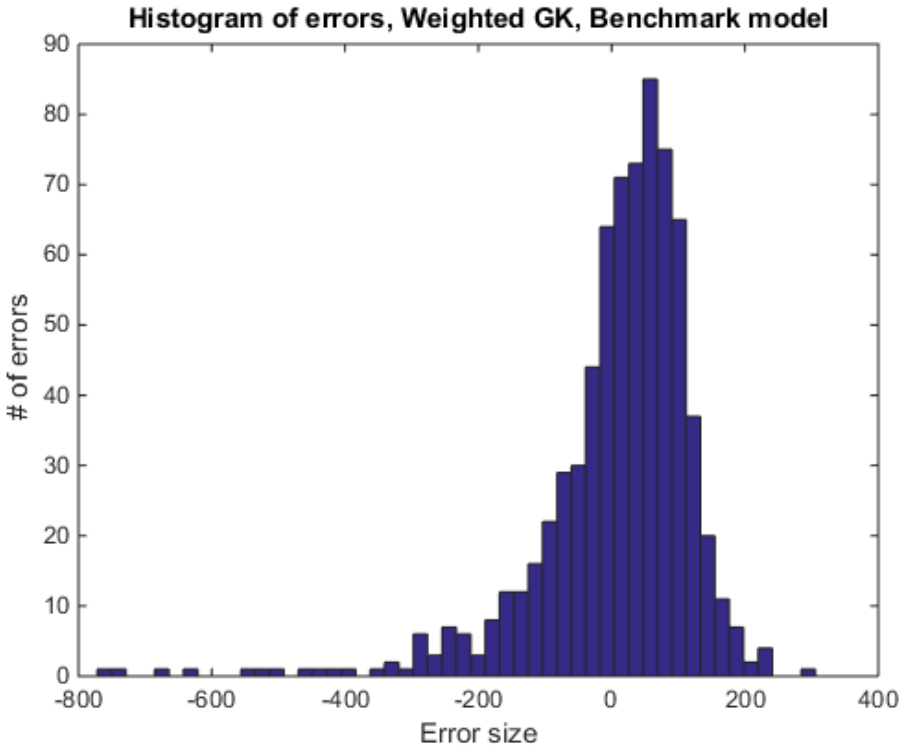

Figure 55: Histogram of errors, weighted GK, Benchmark model

Histogram of errors, totally balanced subtractive, benchmark model

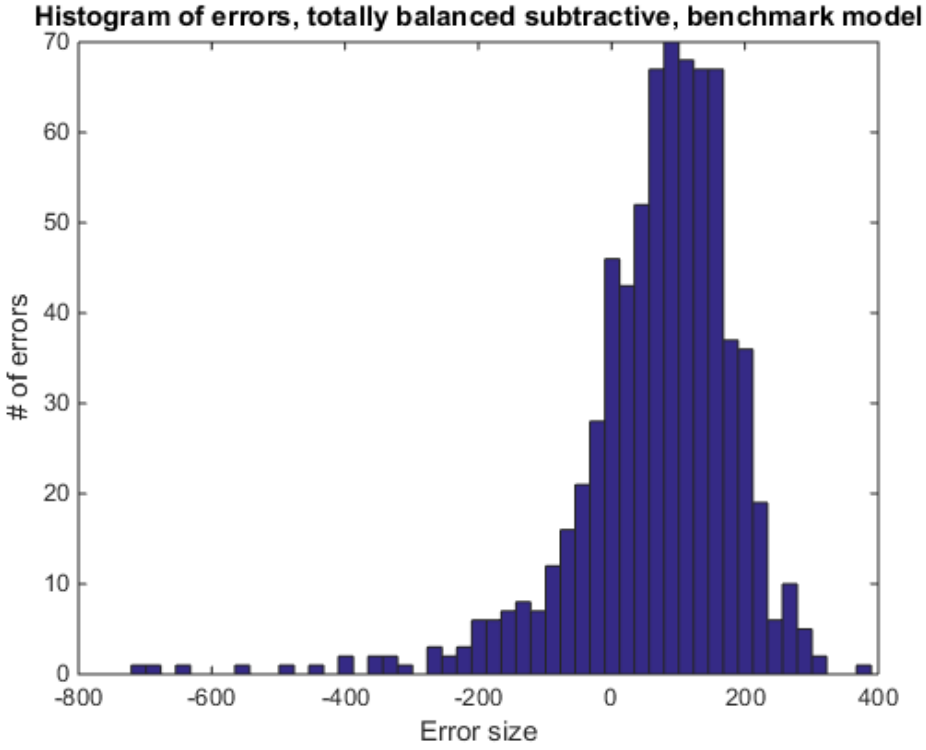

Figure 56: Histogram of errors, totally balanced subtractive clustering, benchmark model

Histogram of errors, 0-order Takagi-Sugeno with weighted training, Benchmark model

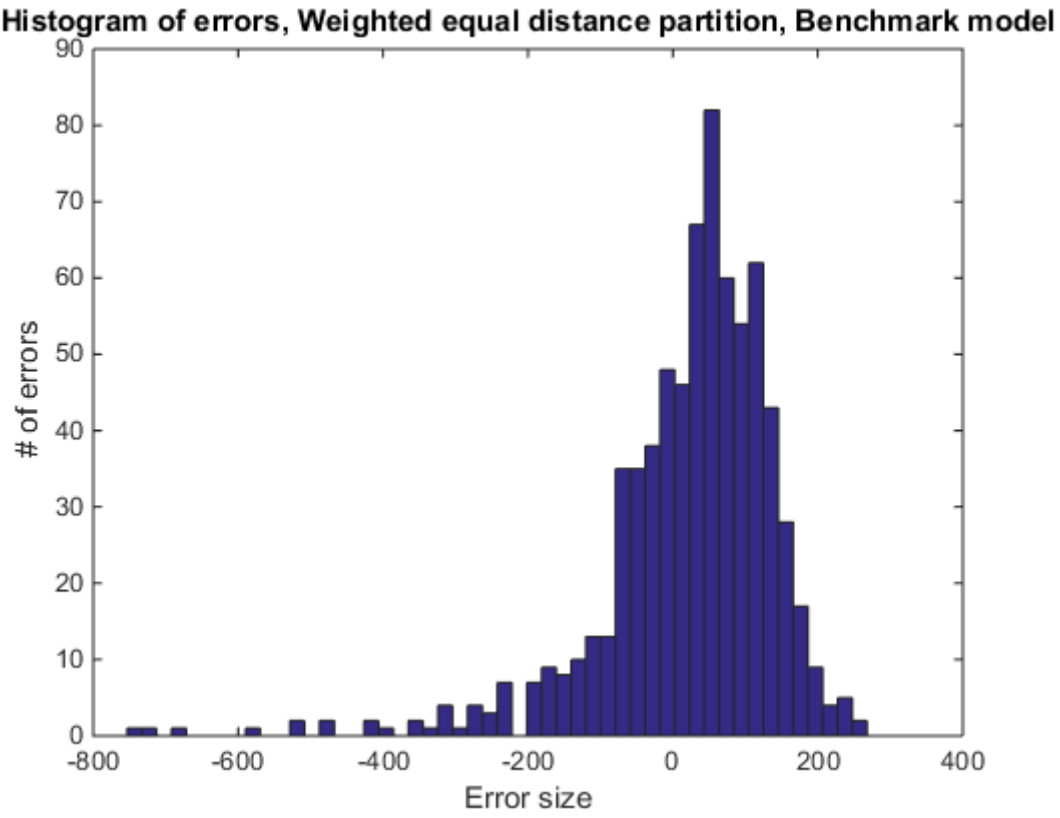

Figure 57: Histogram of errors, 0-Order Takagi-Sugeno weighted training, Benchmark model

Histogram of errors, FCM with age added, benchmark model

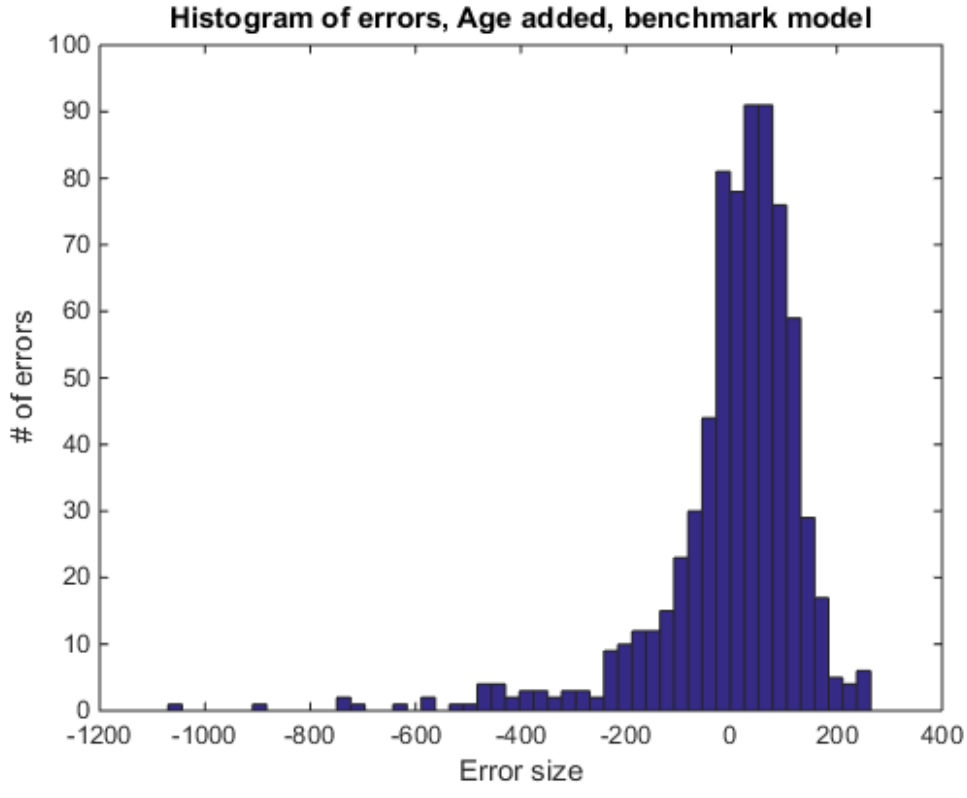

Figure 58: Histogram of errors, FCM with age added, benchmark model

Histogram of errors, FCM with Hcy added, benchmark model

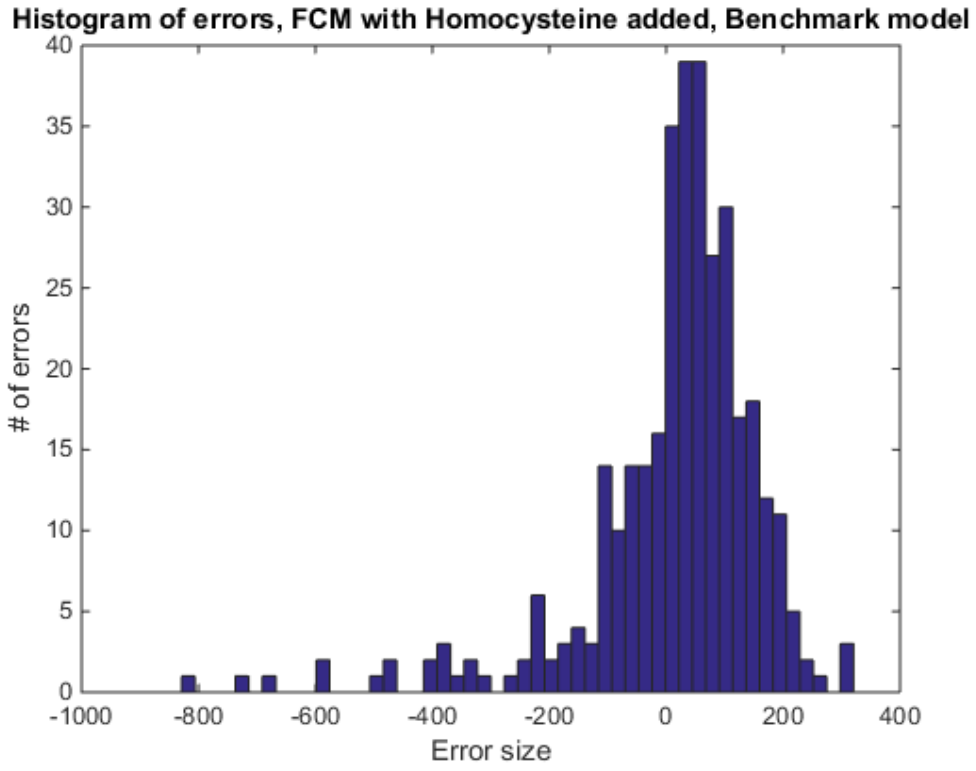

Figure 59: Histogram of errors, FCM with Hcy added, benchmark model

Histogram of errors, FCM with folate added, model 4

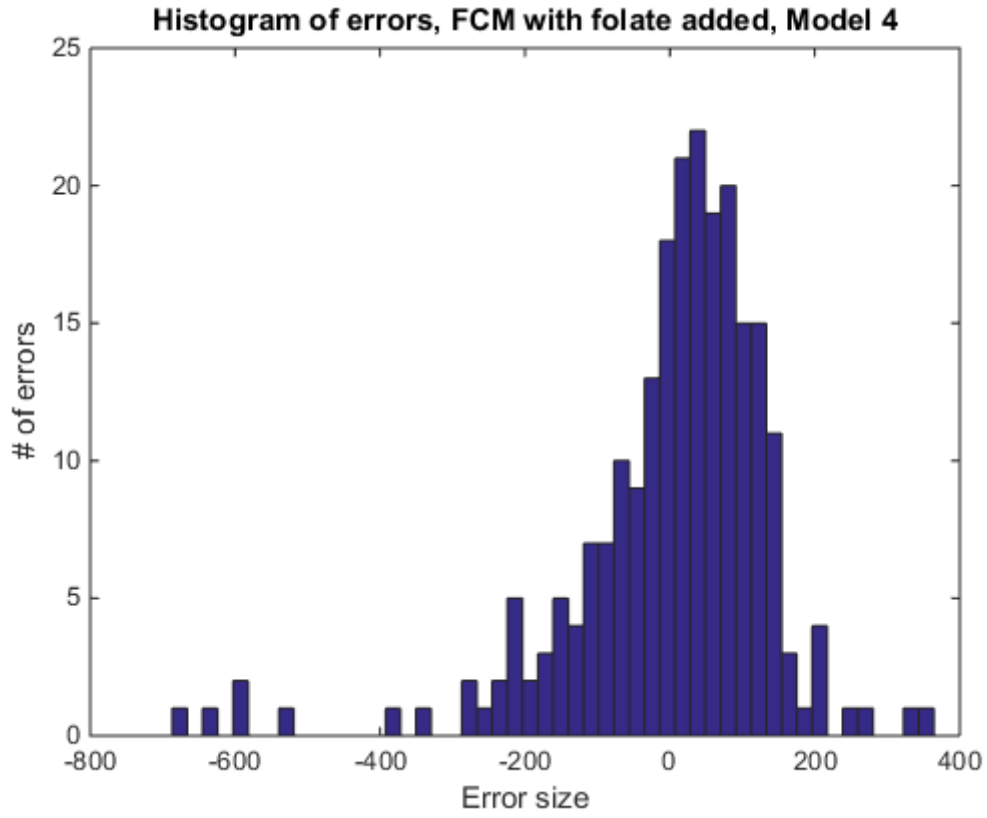

Figure 60: Histogram of errors, FCM with folate added, model 4

Histogram of errors, FCM with haemoglobin added, benchmark model

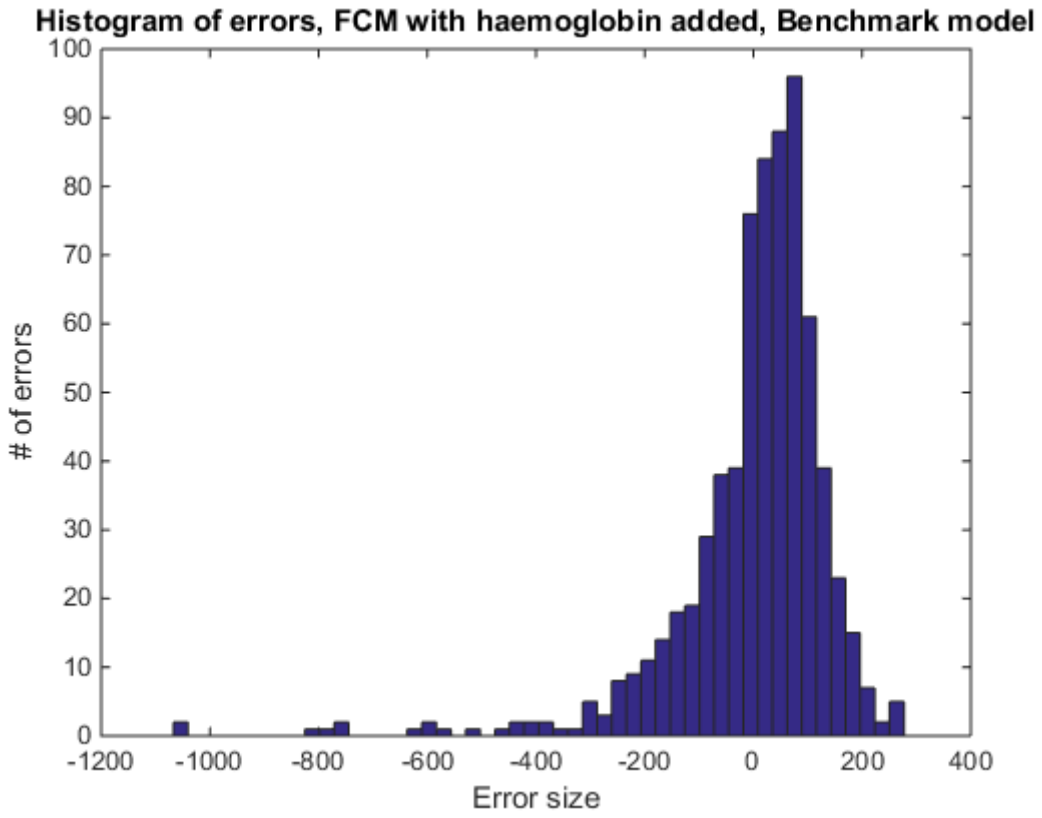

Figure 61: Histogram of errors, FCM with haemoglobin added, benchmark model

Histogram of errors, FCM all biomarkers added, benchmark model

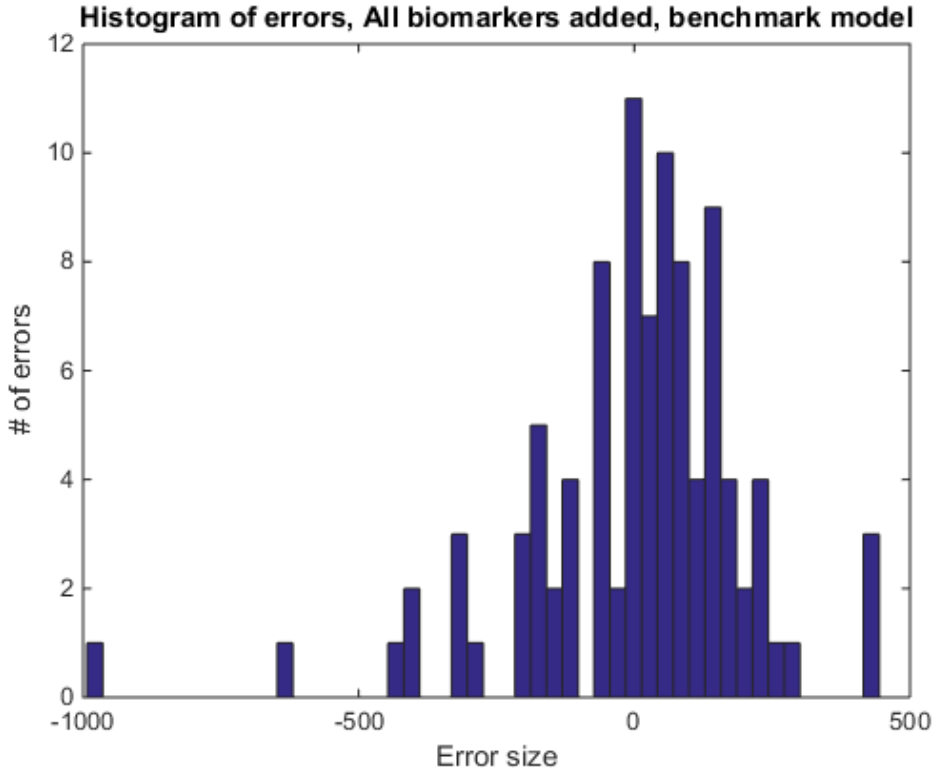

Figure 62: Histogram of errors, 0-order Takagi-Sugeno, Benchmark model

Histogram of errors, 4 layered fitnet, model 4

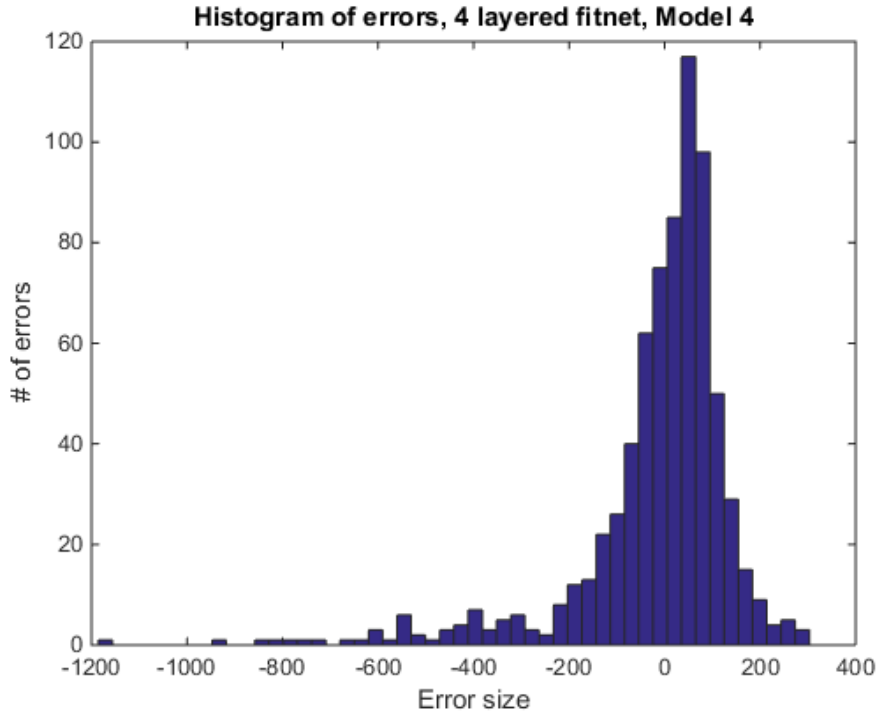

Figure 63: Histogram of errors, 4 layered fitnet, model 4

Histogram of errors, 4 layered cascadeforwardnet, model 6

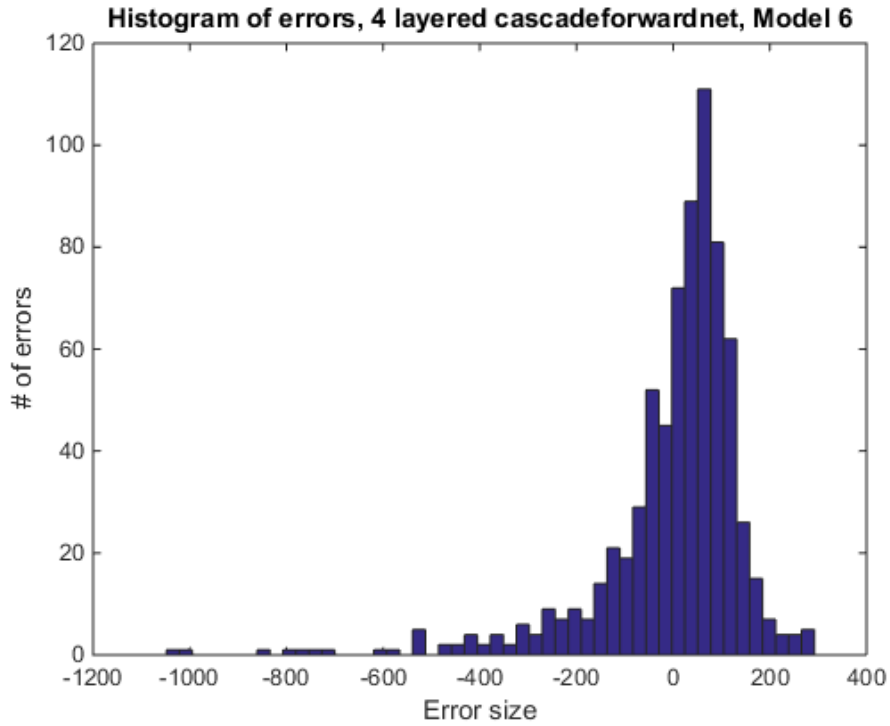

Figure 64: Histogram of errors, 4 layered cascadeforwardnet, model 6

Histogram of errors, bootstrap aggregated decision trees, 100 trees

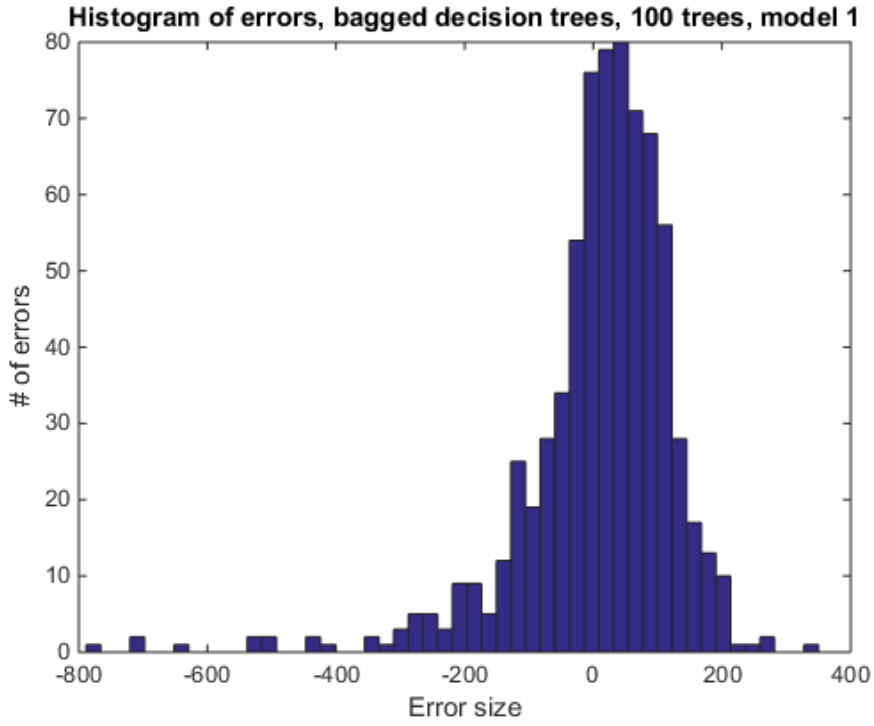

Figure 65: Histogram of errors, bootstrap aggregated decision trees, 100 trees

Histogram of errors, bootstrap aggregated decision trees, 1000 trees

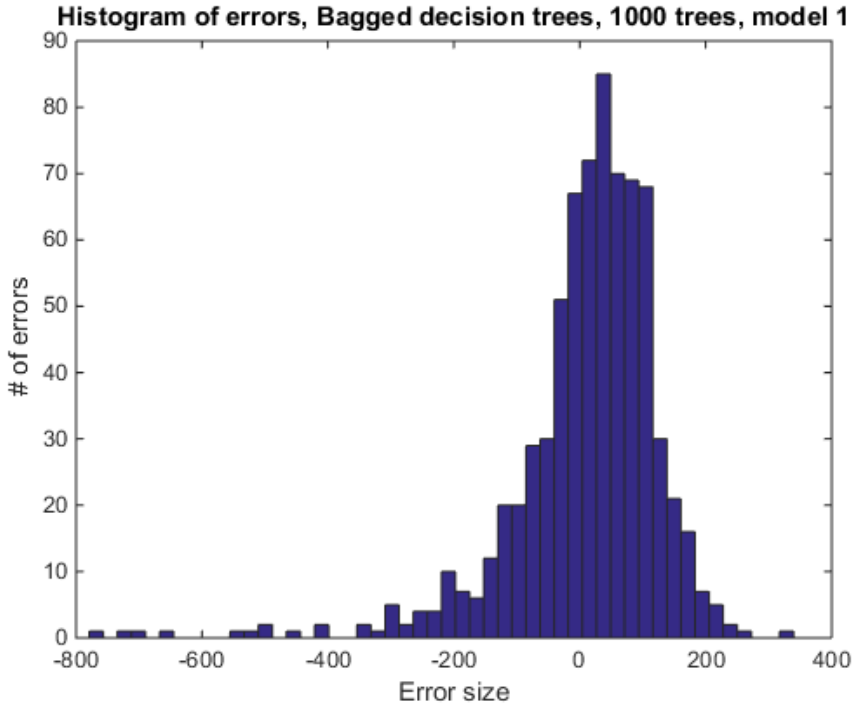

Figure 66: Histogram of errors, bootrstrap aggregated decision trees, 1000 trees

Histogram of errors, bootstrapped aggregated decision trees, 100 balanced trees

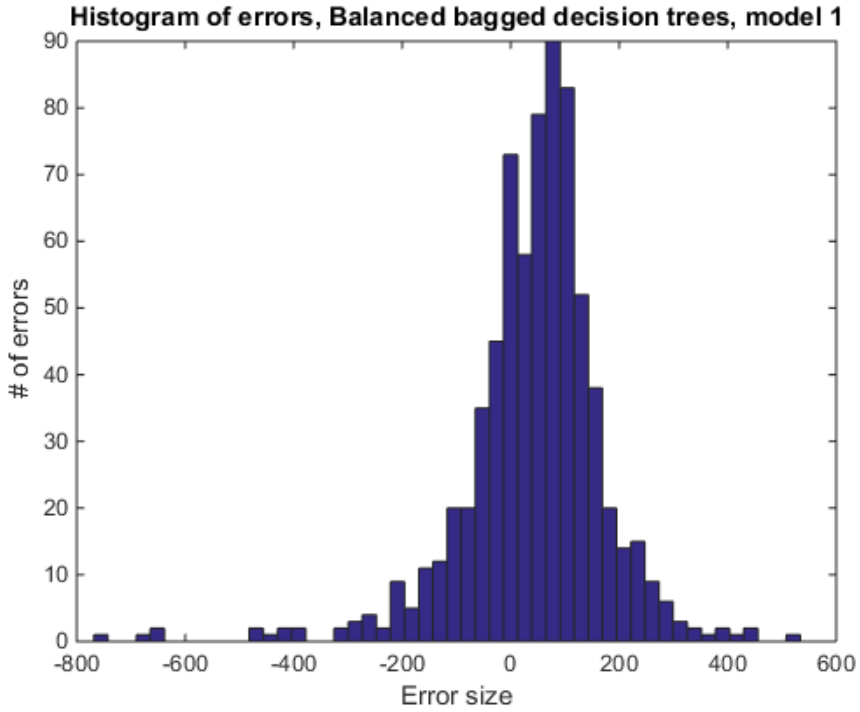

Figure 67: Histogram of errors, bootstrap aggregated decision trees, 100 balanced trees

7.3 Scatterplot of errors

Scatterplot of errors, weighted fcm, benchmark model

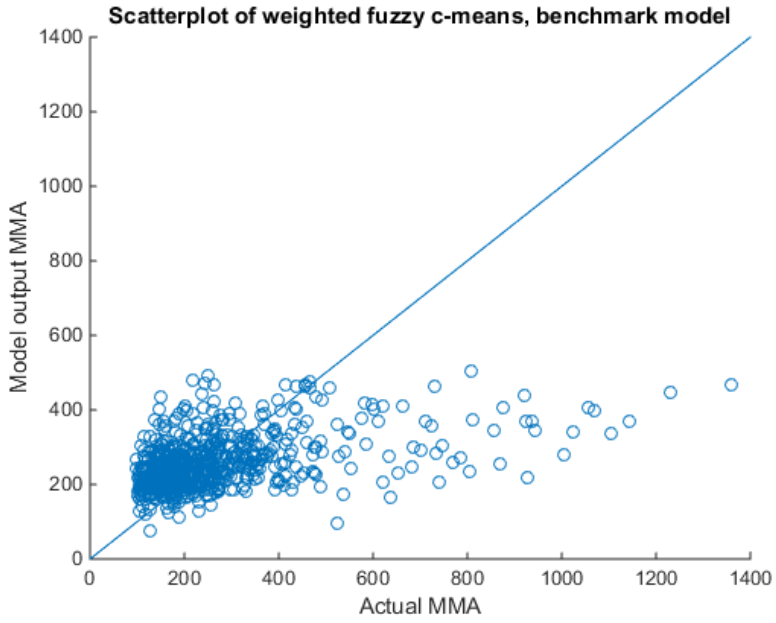

Figure 68: Scatterplot of errors, weighted fuzzy c-means, benchmark model

Scatterplot of errors, Fuzzy c-means with Haemoglobin, benchmark model

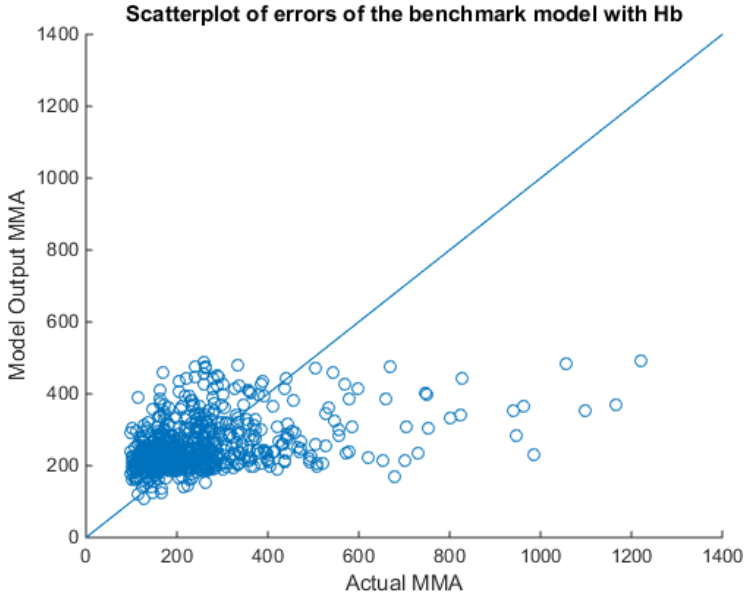

Figure 69: Scatterplot of errors of benchmark model, fuzzy c-means with haemoglobin

Scatterplot of errors, 4 layered cascadeforwardnet model 4

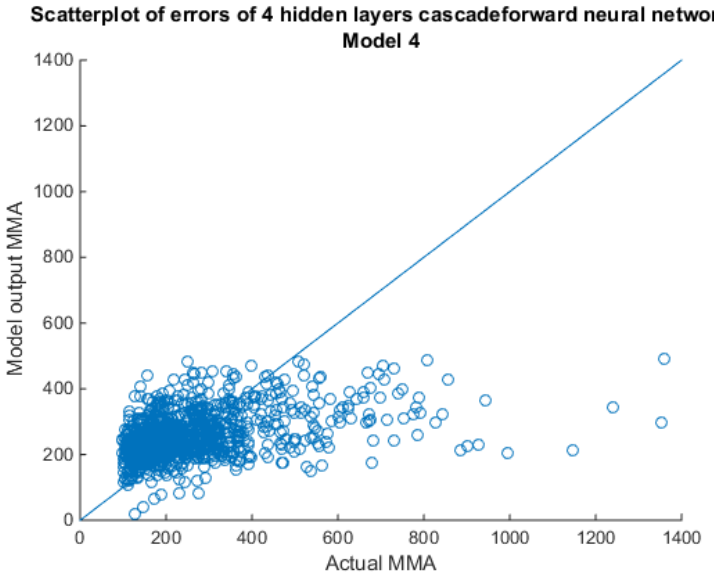

Figure 70: Scatterplot of errors of model 4, generated by 4 layered cascadeforwardnet

Scatterplot of errors, bootstrap aggregated decision trees, 1000 trees

Figure 71: Scatterplot of errors, bootstrap aggregated decision trees, 1000 trees, model 1

7.4 Fuzzy c-means model with correct data

The difference in scale can be seen in the figure which shows the 2MCA1 measurements per date. Measurements before 14 November 2014 have a completely different scale compared to measurements afterwards. The measurements done before were removed after which 5870 points remained. Then fuzzy c-means clustering was done since that was the best performing method of the previously made models.

This included a 10 times 10-fold cross validation for training and testing. Furthermore 2 up to 10 clusters were tried to get the best results.

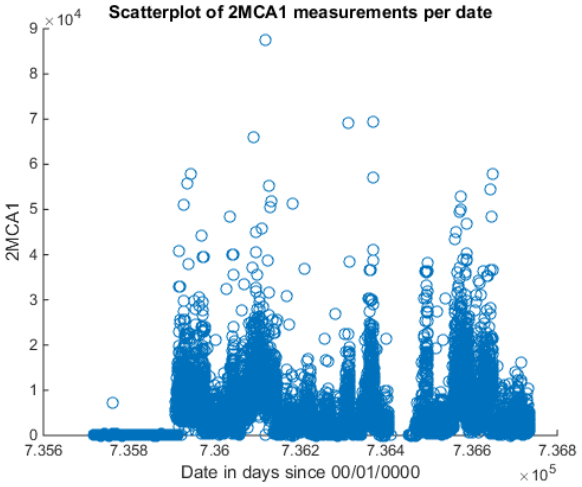

Figure 72: Scatterplot of 2MCA1 measurements per date

Results

The models with 2MCA1 and 2MCA2 now perform better compared to the models with the two different scales mixed together. These removals had no effect on the performance of the benchmark model and the model with 2MCAR added since their data was unaffected. These unaffected models also still perform the best, with the benchmark being a bit better than model 4.

Table 39: Best error values of the models with corrected data

	MAPE	MAE	MSE	R-squared
Model 1	0,3775	95,63	21468	0,3670
Model 2	0,3679	94,68	21189	0,3753
Model 3	0,3711	95,21	21290	0,3723
Model 4	0,3648	94,55	21159	0,3761
Model 5	0,3777	95,27	21363	0,3701
Model 6	0,3699	95,01	21266	0,3730
Model 7	0,3711	95,28	21391	0,3693
Benchmark	0,3634	94,24	21094	0,3781