Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Biotechnology and Food Science

Amalie Mello

A comparison of machine learning algorithms to investigate methylation profiles and predict type 1 diabetes

Master's thesis in Industrial chemistry and biotechnology
Supervisor: Professor Eivind Almaas
Co-supervisor: Postdoc. André Voigt
July 2021

NTNU
Norwegian University of
Science and Technology

Amalie Mello

A comparison of machine learning algorithms to investigate methylation profiles and predict type 1 diabetes

Master's thesis in Industrial chemistry and biotechnology
Supervisor: Professor Eivind Almaas
Co-supervisor: Postdoc. André Voigt
July 2021

Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Biotechnology and Food Science

Summary

The incidence and prevalence of type 1 diabetes in the world has increased the last four decades [1; 2]. Rakyan et al. had a hypothesis that some of the non-genetic factors were due to epigenetic variation [3]. The aim of this Master's thesis was to train models to predict from methylation profiles whether a person had developed type 1 diabetes, and identify possible type 1 diabetes associated genes. This was executed by training models by various machine learning algorithms.

A dataset composed from methylation profiles generated by Rakyan et al. and Bell et al. was split and used as training and testing data. After pre-processing, the dataset consisted of 27,006 lines of CpG sites, 226 columns of individuals diagnosed with type 1 diabetes and 68 columns of individuals without the diagnosis. Methylation levels for all CpG sites for all individuals were given as a value between zero and one.

The best K-value for the K-nearest neighbours classifier was identified by training models with different K-values. A K-value of 15 gave the highest Matthews correlation coefficient. The machine learning classifier algorithms of logistic regression, decision tree, K-nearest neighbours, random forest and multilayer perceptron were compared with the same comma-separated values file as training data.

Matthews correlation coefficient was considered a proper performance measure, because it may be used to evaluate binary classification predictions on

imbalanced datasets [4]. The machine learning algorithms performed evenly high with an average Matthews correlation coefficient of around 0.65. Training data may therefore be more important than the model.

The genetic algorithm Sklearn-genetic and the feature selector from Scikit learn were used to find feature selections that alone trained the most suitable models [5; 6]. Models trained with a limited feature selection tended to score higher. Nine CpG sites were found in more than one feature selection. All nine CpG sites were considered candidates for T1D relevance. Among these nine sites, two were type 1 diabetes associated. Based on literature search [7] and the results, DNA methylation of the LY86 gene appears to be associated with insulin deficiency. The approach was suitable for type 1 diabetes prediction and to identify possible type 1 diabetes associated genes. Some possible adjustments to the approach were suggested in order to reach its full potential.

Sammendrag

Forekomsten og utbredelsen av type 1 diabetes i verden har økt de siste fire tiårene [1; 2]. Rakyan et al. hadde en hypotese om at noen av de ikke-genetiske faktorene skyldtes epigenetisk variasjon [3]. Målet med denne masteroppgaven var å trene modeller for å forutsi fra metyleringsprofiler om en person hadde utviklet type 1 diabetes, og identifisere mulige type 1 diabetes assosierte gener. Dette ble utført ved å trene modeller med forskjellige maskinlæringsalgoritmer.

Et datasett sammensatt av metyleringsprofiler generert av Rakyan et al. og Bell et al. ble splittet og brukt som trenings- og testdata. Etter prosessering bestod datasettet av 27,006 linjer med CpG-dinukleotider, 226 kolonner av individer med type 1 diabetes diagnoser og 68 kolonner av individer uten type 1 diabetes. Metyleringsnivåer for alle CpG-dinukleotider for alle individer ble gitt som en verdi mellom null og én.

Den beste K-verdien for K-nærmeste naboer klassifikator ble identifisert ved å trene modeller med forskjellige K-verdier. En K-verdi på 15 gav den høyeste Matthews korrelasjonskoeffisienten. Klassifikator-maskinlæringsalgoritmene for logistisk regresjon, beslutningstre, K-nærmeste naboer, tilfeldig skog og flerlags perceptron ble sammenlignet med samme kommaseparerte fil som treningsdata.

Matthews korrelasjonskoeffisient ble ansett som et passende ytelsesmål, da det kan brukes til å evaluere binære klassifiserings prediksjoner på ubalanserte datasett [4]. Maskinlæringsalgoritmene presterte jevnt med en gjennomsnittlig Matthews

korrelasjonskoeffisient på rundt $0,65$. Treningsdata kan derfor være viktigere enn valg av modell.

Den genetiske algoritmen Sklearn-genetic og algoritmen for utvalg av parametere fra Scikit learn ble brukt til å finne parametere som alene trente de mest egnede modellene [5; 6]. Modeller trent med et begrenset utvalg parametere tenderte til å prestere høyere. Ni CpG-dinukleotider ble funnet i mer enn ett utvalg av parametere. Alle ni ble ansett som kandidater for type 1 diabetes relevans. Blant disse var to type 1 diabetes assosierte. Basert på litteratursøk [7] og resultatene, ser det ut til at DNA metylering av LY86 genet er assosiert med insulinmangel. Tilnærmingen var egnet for type 1 diabetes prediksjon og for å identifisere mulige type 1 diabetes assosierte gener. Noen mulige justeringer av tilnærmingen ble foreslått for å nå dens fulle potensiale.

Preface

This master's thesis was carried out in the spring of 2021, as the final part of a Master's degree at the Department of Biotechnology and Food Science, Norwegian University of Science and Technology (NTNU).

The author would like to express her gratitude to the supervisors Professor Eivind Almaas and Postdoc. André Voigt, for being helpful and supportive throughout the process. She wold also like to thank her boyfriend Kristoffer Klungseth. It has been nice to spend the graduate year with him during the pandemic. Her time with fellow students Helge Bergo, Kristin Bentzen, Idun Burgos, Chrisine Sjevelås, Bettina Grorud and Synne Standal Solheim were also highly appreciated.

Contents

Summary i

Sammendrag i

Preface iii

Table of Contents vi

List of Tables viii

List of Figures x

Abbreviations xi

1 Introduction 1

2 Theory 3

2.1 Type 1 diabetes [8] 3

2.2 Gene-regulation [8] 4

2.3 Machine learning 5

2.3.1 Independent and dependent variables 6
2.3.2 Parametric and non-parametric methods 7
2.3.3 Regression and classification problems 7
2.3.4 Supervised and unsupervised learning 8
2.3.5 Machine learning algorithms 8
2.3.6 Comparison of algorithms 14
2.4 Genetic algorithms 16

3 Method **17**
3.1 Data 17
3.2 Methods 19

4 Results and analysis **23**
4.1 Results from Machine Learning 23
4.2 CpG sites that exist in several selections 29
4.3 Comparison with CSD results 30
4.4 Biological function 31

5 Discussion **35**

6 Conclusion **41**

Bibliography **43**

Appendix A **53**

List of Tables

3.1 GeneticSelectionCV was used with the parameters listed. 21

4.1 The corresponding CpG sites to the indices in the Decision tree in Figure 4.3 26

4.2 The MCC of the various algorithms without selection. 27

4.3 The results of the genetic selection. CpG sites which exist in several selections are marked in **bold**. CpG sites which both exists in several selections and have T1D related functions are marked in in **blue**. 28

4.4 The results of the select from model features selection. CpG sites which exist in several selections are marked in **bold**. CpG sites which both exists in several selections and have T1D related functions are marked in in **blue**. 29

4.5 The table lists the CpG sites found in more than one selection. It was not included if it was only repeated in the same ML algorithm, same type of selection and same number of maximum features, but with different population sizes. 30

4.6 The table lists the nodes (CpG sites) with the highest degrees k_i, which means the highest number of neighbours. The node with the highest degree is found in the biggest component of 212 nodes. The CpG sites cg10031456 and cg09088193 are neighbours, and is found in the second biggest component of 45 nodes [8]. 31

4.7 Biological functions and processes of the nine CpG sites repeated in several selections listed in Table 4.5. They were accessed with the NCBI gene tool and literature search [9; 10; 6; 11; 12; 7; 13; 14]. T1D related functions are marked in in blue. 32

List of Figures

2.1 An example of a decision tree created by Raschka and Mirjalili from the book *Python Machine Learning* [15]. The example is using the Iris dataset, and different features are used to decide class [16]. The decision criterion is entropy, and the maximum depth is 4 [15]. 11

2.2 An example of a neural network. Neuron a_k in Layer 2 is the result of three inputs including neuron a_j. The figure is based on a figure from the book *Machine Learning Meets Quantum Physics* [17]. . . 13

3.1 A flow chart that shows input and output from various files. 19

4.1 Different K-values in K-nearest neighbours affected the performance score. (a) The different performance scores were plotted against K-values between 1 and 50. (b) The same points were fitted to a polynomial curve. 24

4.2 Different K-values in K-nearest neighbours affected the performance score. In separate windows, the different performance scores were plotted as box plots against K-values between 1 and 30. 25

4.3 Decision tree classifier created with 'machine_learning.py' and visualised using Visual Studio Code [18]. 26
4.4 The box plots created as a result of 30 of each model type, with different kind of performance measures. 27
4.5 Wang et al. created the disease-associated lncRNA-mRNA-pathway network with a weighed gene coexpression network approach [12]. The red dots were lncRNAs, the blue dots were mRNAs, the orange square were the disease pathway, and the bigger nodes were the disease genes [12]. 33

Abbreviations

Symbol		**definition**
AI	=	Artificial intelligence
CpG	=	Cytosine and guanine separated by a phosphate group
CSD	=	Systematic framework that takes conserved, specific and differentiated co-expression into account
CSV	=	Comma-separated values
DNA	=	Deoxyribonucleic acid
GA	=	Genetic algorithm
ML	=	Machine learning
MLP	=	Multilayer perceptron
MSE	=	Mean squared error
MZ	=	Monozygotic
RSS	=	Residual sum of squares
SL	=	Statistical learning
T1D	=	Type 1 diabetes
T1D-MVP	=	Type 1 diabetes methylation variable position
TF	=	Transcription factor

Chapter 1

Introduction

Between 2001 and 2009 a study showed a 30% increase in prevalence of type 1 diabetes (T1D) among children and adolescents aged 0 – 19 years in the USA [2]. A meta-analysis on databases from January 1980 to September 2019 also concluded that the incidence and prevalence of T1D were increasing in the world [1]. Rakyan et al. had a hypothesis that some of the non-genetic factors were due to epigenetic variation [3]. In order to understand complex biological systems, like the epigenetic variation of T1D, experimental and computational research are important contributors. This is the field of systems biology, where computational biology, pragmatic modelling and theoretical exploration are used [19].

In an earlier specialisation project, the author executed a differential co-expression network analysis using software programs developed by Voigt et al. [20; 8]. Differential co-expression network analysis is an important tool for investigation of differentiation and dysfunctional gene-regulation in diseases. The DNA methylation profiles generated by Rakyan et al. were used to create a network that was analysed. The project aimed to identify possible T1D associated network

patterns. In order to achieve results with statistical power, further systematic analyses were proposed [8]. Approaches using machine learning (ML), were suggested to obtain this [21].

The aim of this Master's thesis was to create models to predict from methylation profiles whether a person had T1D and to identify possible T1D associated genes. This Master's thesis executed training of models by ML algorithms to predict whether a person had developed T1D or not. DNA methylation profiles generated by Rakyan et al. and Bell et al. were used as training data [3; 22].

Exploring this thesis was a next step after the earlier specialisation project. Therefore, the theory about T1D and gene-regulation in Sections 2.1 and 2.2 is a revised edition of the same subsections in the earlier project [8]. Some of the input data used in the current study were also used in the other project, and some of the descriptions of the data in Section 3.1 are based on earlier work [8]. Some of the code used in this thesis is based on code from the previous project [8].

Chapter 2

Theory

Part of the aim in this thesis was to create models to predict from methylation profiles whether a person had T1D or not. Theory of T1D, gene-regulation and ML will be presented in this chapter. The aim was also to identify possible genes associated with T1D. Genetic algorithms (GA) generate new sample points that are optimal values of a function, and will also be reviewed [23; 5].

2.1 Type 1 diabetes [8]

T1D is a chronic disease [24]. The symptoms of untreated T1D are impaired general condition, polyuria, thirst and loss in weight [25]. The blood glucose levels can become so high that the patient becomes dizzy or falls into a coma [25]. Insulin is secreted by β-cells and is necessary for the transport of glucose from the blood to the cells [26]. People with T1D have absolute insulin deficiency due to β-cell destruction [27]. The β-cell destruction is normally caused by the immunity mechanisms. Patients therefore need a life long treatment with insulin [24]. $5-10\%$ of all diabetes cases are type 1 diabetes [28]. The prevalence of

T1D among children and adolescents has increased since the beginning of the millennium [2]. Dabelea et al. reported that in 2009, 6666 out of 3,4 million US youth were diagnosed with T1D for a prevalence of 1,93 per 1000. This was an increase of 30% since 2001. Further, a meta-analysis on databases from the year 1980 until 2019, concluded that the T1D prevalence in the world is increasing [1]. Inheritance is a large part of the cause of why a person develops T1D [28]. Nevertheless, the triggering factors of unset of the clinical disease is not fully understood. The monozygotic (MZ) twin pair discordance for the complex autoimmune disease childhood-onset T1D is around 50% [3]. Rakyan et al. had a hypothesis that some of the non-genetic factors were due to epigenetic variation. From purified immune effector CD14+ monocytes, they generated genome-wide DNA methylation profiles. After array processing, identification of T1D methylation variable positions (T1D-MVPs), pyrosequencing validation and analysis, it was suggested that very early in the etiological process to the onset of T1D, T1D-MVPs arise [3].

2.2 Gene-regulation [8]

Genetic variation and diversity lead to a range of human phenotypes and regulate gene expression in cell differentiation over time [29]. However, in some cases it has been linked to disease [29]. Transcription factors (TFs) are proteins that regulate gene expression. At the promoter of a gene, there are interactions of TFs. The sum of these interactions determines whether the gene is activated, repressed, or not regulated [30].

DNA methylation in cytosine and guanine separated by a phosphate group (CpG) is one of the ways that a TF can regulate the gene expression [31]. It is the biological process whereby a methyl group is covalently added to a cytosine, and gives

5-methylcytosine. The biological process is an important epigenetic mark in eukaryotes [31; 32]. The enzymes that carry out the biological process is called DNA methyltransferases. DNA methylation is affecting transcriptional activity, and this may be associated with diseases [33]. Recent genomic technological advances have made it possible to run large scale studies of human disease associated or tissue-specific epigenetic variation, such as comparing DNA methylation profiles [3]. DNA methylation as gene regulator may be a more complex process than repression of gene expression [31]. Causality and the physiological explanation of DNA methylation level variance are not fully studied. However, network theory and ML can be used to study differences between conditions and predict condition.

2.3 Machine learning

ML is a sub-area of artificial intelligence (AI) [34]. AI is simulation of human intelligence [35]. ML is not necessary a simulation of human intelligence, but a tool commonly used in that context, and is suggested as a solution to the information overload challenge in the 21st century [36]. A large amount of academic research publications could for example be systematically mapped with computer assistance that aim to catalogue broad evidence bases [37]. There is often improvement in scaling up training data sets in current ML systems [31].

Statistical learning (SL) is an area in statistics that blends with the parallel development of ML, and is referred to as a branch of ML [38; 36]. Rather than making a distinction between SL and ML, the term ML will mainly be used in this thesis. ML is a field where systems use datasets, consisting of samples, to learn [39]. ML is an essential tool for extracting regularities in a dataset and for making inferences [17]. The samples are of different features, and can be categorical, ordinal,

or numerical [39]. Therefore, different kinds of data collected from patients can be used.

2.3.1 Independent and dependent variables

When a set of independent variables (X) are known, a dependent variable (Y) may be predicted, as shown in Equation 2.1 [38].

$$Y = f(X) + \epsilon, \tag{2.1}$$

where ϵ is the random error term and does not depend on X. With a sufficient number of independent variables, ϵ will be approximately zero. f is estimated to predict something, to look at the inference between Y and X, or a combination of the two [38]. p independent variables X are denoted X_1, X_2 ... X_p, and n observations are denoted x_1, x_2 ... x_n, together forming an $n \cdot p$ matrix.

When a set of input variables X is available, and one wants to predict Y, Equation 2.2 can be used, as ϵ averages to zero [38].

$$\hat{Y} = \hat{f}(X) + \epsilon, \tag{2.2}$$

where $\hat{Y}$ is a prediction of Y, and $\hat{f}$ is an estimate of f. If the goal is only to predict, and not to understand the inference, $\hat{f}$ can be looked at as a black box. To investigate how Y is affected by X, an estimate $\hat{f}$ is calculated to reach an understanding [38]. For example, it can be looked at which predictors that are associated with the response, then $\hat{f}$ is no longer considered a black box.

2.3.2 Parametric and non-parametric methods

Various ML methods can be used to obtain $\hat{f}$. The methods can be divided into parametric and non-parametric [38]. Parametric methods start with assuming a shape or form of f. Then a procedure is followed to train the model using training data, that defines a set of parameters [38]. In general, it requires a large number of parameters to make a more flexible model, but this can also lead to overfitting [38]. Non-parametric methods do not assume a shape of f, but seek to find a function close to the training data [38]. As non-parametric methods are not trying to fit any shape, overfitting will not be an issue. The disadvantage with these types of methods is that they require a large number of observations (n) [38].

2.3.3 Regression and classification problems

Variables can be quantitative or qualitative [38]. Quantitative variables have a numeric value, while qualitative variables are categorical [38]. An example of a quantitative variable thus can be the degree of methylation of a CpG site. Whether a person has a disease or not can be a qualitative variable. When the response variable (Y) is quantitative it is a regression problem, and when it is qualitative, it is a classification problem [38].

Linear regression models model a straight line, and are often used when the response variables are quantitative [40]. If the response variable is qualitative with two categories, a cut off of $Y = 0.5$ could be used, but with three categories or more it seldom will make any sense to range the variables in some order. Therefore, logistic regression is more suitable when the response value is qualitative [38]. Logistic regression and other methods will be covered later in the section. Some methods can be used both with quantitative and qualitative response [38].

2.3.4 Supervised and unsupervised learning

In supervised learning, statistical models are trained using training data (x_1, y_1), ..., (x_n, y_n) in order to predict or estimate output [38]. When a dataset includes dependent variables $y_1, ..., y_n$, it is called labelled data [41]. From unsupervised learning, relationships and structure can be learned from input data $x_1, ..., x_n$ [38]. Thus, unsupervised learning takes place without supervising output. Clustering is in that case useful. Based on unlabelled input data $x_1, ..., x_n$, a cluster analysis will reveal whether the observations fall into relatively distinct groups [38].

2.3.5 Machine learning algorithms

Logistic regression

As seen in Section 2.3.3, the linear regression model causes problems when used to predict binary response. The linear regression model fits Equation 2.3, but when the response is binary, the logistic function in Equation 2.4 is more appropriate to use [38]:

$$p(X) = \beta_0 + \beta_1 X \tag{2.3}$$

$$p(X) = \frac{e^{\beta_0 - \beta_1 X}}{1 + e^{\beta_0 + \beta_1 X}} \tag{2.4}$$

The logistic regression model forms an S-shape between zero and one [38]. The parameters β_0 and β_1 are estimated based on training data, enabling the predicted probability $\hat{p}(x_i)$ for each individual to correspond as closely as possible to the values of the observed individual [38]. This is done by using the likelihood

function of Equation 2.5 [38]. When multiple predictors are used, multiple logistic regression is performed following Equation 2.6 [38].

$$l(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y'_i=0} (1 - p(x_{i'})) \tag{2.5}$$

$$p(X) = \frac{e^{\beta_0+\beta_1 x_1+...+\beta_p x_p}}{1 + e^{\beta_0+\beta_1 x_1+...+\beta_p x_p}} \tag{2.6}$$

When using Scikit learn to train models with logistic regression, a set of parameters can be established as seen in the documentation [5]. To avoid overfitting with a large number of features, the lasso (L1) regularisation, ridge regression (L2) regularisation or a combination called the elastic net, can be used as penalty [42]. L2 is simple and fast and is good to avoid overfitting [43]. L1 has more sparse properties and is suitable for datasets with a large number of features [43]. The solver parameter has to support the penalty. The 'newton-cg', 'sag' and 'lbfgs' solvers support the L2 penalties, and 'elasticnet' is supported by the 'saga' solver [5].

Decision trees

Some methods use a set of splitting rules to segment the predictor space that forms a tree. These methods are called decision tree methods and are applied both in regression and classification problems [38]. In regression trees, the response variable for a given observation is the mean response of the training observations in the same terminal node [38]. A classification tree predict the response for a given observation by finding the most common category among the training data in the region it belongs to [38]. While interpreting a classification tree, the proportion of training observations that fall into each region, is

also important [38]. A split in a regression tree is a result of using residual sum of squares (RSS) as criterion, as shown in Equation 2.7. In classification problems, the Gini index, calculated in Equation 2.8 can be used as criterion for the split [38].

$$RSS = (y_1 - \hat{\beta}_0 - \hat{\beta}_1 x_1)^2 + ... + (y_n - \hat{\beta}_0 - \hat{\beta}_1 x_n)^2 \tag{2.7}$$

$$G = \sum_{k=1}^{\kappa} \hat{p}_{mk}(1 - \hat{p}_{mk}), \tag{2.8}$$

where κ is the number of classes and $\hat{p}_{mk}$ is the proportion of training observations in the mth region which originate from the kth class. A small Gini index means that the node has observations mainly from one class.

Deeper decision trees have a more complex decision boundary and can cause overfitting [15]. In Scikit learn, the maximum depth can be determined [44]. The default settings for decision tree classifier in Scikit learn, is shown in the documentation [5].

Decision trees are easy to visualise graphically, and therefore also easy to interpret [45]. An example of a visualisation of a scikit learn decision tree is showed in Figure 2.1 [15; 44]. A small change in the training data, can make the final tree look totally different, so it is non-robust. There are methods that utilises several decision trees, such as random forest, which will make the trees considerably more robust [38].

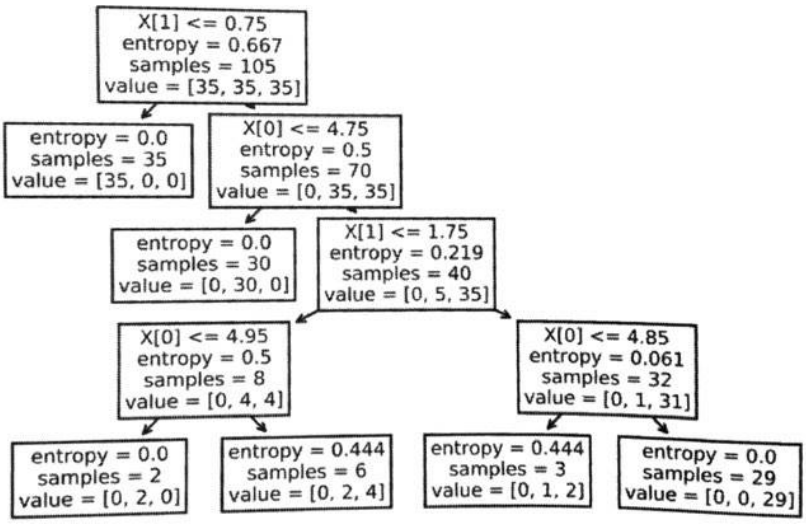


Figure 2.1: An example of a decision tree created by Raschka and Mirjalili from the book *Python Machine Learning* [15]. The example is using the Iris dataset, and different features are used to decide class [16]. The decision criterion is entropy, and the maximum depth is 4 [15].

K-nearest neighbours

The prediction with K-nearest neighbours for an observation x is executed by using the K training observations closest to x [38]. The K-nearest neighbours classifier predicts that the observation belongs to the same category as the plurality of the neighbours [38]. In K-nearest neighbours regression, the average value of the neighbours is what is predicted for the observation [46]. Thus, the predicted function for Y will not assume a shape, and the method is non-parametric. The approach is relatively uncomplicated, but does not specify which variables are most important. When $K = 1$, it will be an overly flexible decision boundary with low bias and high variance [38]. However, with a high K-value the decision boundary will have low flexibility, low variance and high bias. The Python machine learning library Scikit learn has a K-nearest neighbours classifier [5].

Random forest

As mentioned, random forest methods use several decision trees to create models. Building of a decision tree is done by every time a split is made, only a random sample of m of the p predictors are considered candidates for the split [38]. The size of m often is as calculated in Equation 2.9 [38]:

$$m \approx \sqrt{p}. \tag{2.9}$$

The average of several trees is calculated. If all p candidates were considered in all the trees, the trees would have been similar, and the variance would not have been much smaller than for a normal decision tree [38]. Scikit learn has a random forest classifier [5].

Neural networks

In the context of neural networks, a neuron is a simple computational unit [17]. A large number of neurons are interconnected in layers to form highly complex predictions [17; 47]. Besides being flexible, neural networks are also scalable, as the network can maintain their representation confined to finitely number of neurons [17].

Neural networks can be seen as a function f, where the inputs are an observation x and an input vector that is learned from the learning data. For a neuron with index k, Equation 2.10 shows how the a_k is calculated [17].

$$a_k = \rho(\sum_j a_j w_{jk} + b_k), \tag{2.10}$$

where w_{jk} are the weights and b_k is the bias, and they both have to be learned

from the training data. a_j was obtained from the previous layer, and a_k will be its replacement in the next layer, if it is not the final output y. The sum is the weighted sum over all j neurons in the input to neuron k. ρ is an activation function that can be made in different manners. A simple neural network based on a figure from the book *Machine Learning Meets Quantum Physics* is illustrated in Figure 2.2 [17].

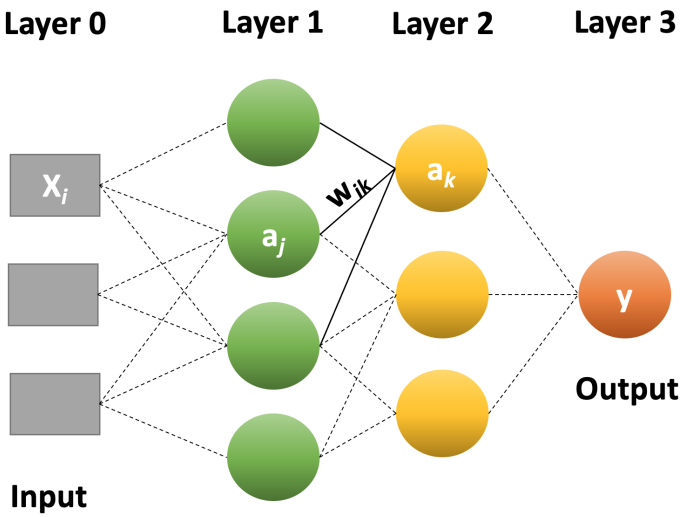


Figure 2.2: An example of a neural network. Neuron a_k in Layer 2 is the result of three inputs including neuron a_j. The figure is based on a figure from the book *Machine Learning Meets Quantum Physics* [17].

Multi layer perceptron (MLP) is a class of neural networks [48]. An MLP has several layers of neurons. The role of the input layer is to pass the input vector to the network [49]. Then there may be one or more layers before the output layer. MLPs are considered fully connected. Each node is connected to every node in the next and previous layer [49]. Scikit learn has an available MLP classifier [5].

2.3.6 Comparison of algorithms

Different machine learning algorithms are suitable for different datasets and areas of use. When the true decision boundaries are linear, linear regression and logistic regression can be a good fit [38]. If the true decision boundaries however are more complicated, non-parametric methods like K-nearest neighbours, will fit better. Decision trees are simple and easy to interpret [38]. However, algorithms composed of several trees, like random forest, will have better prediction accuracy and be harder to interpret [38]. Neural networks are both scalable and flexible [17]. A large neural network can represent a wide class of functions with few errors, but then each training iteration will last longer [17]. Neural networks thus have limitations when it comes to running time.

Performance measure

How well a model predicts a response variable ($\hat{Y}$), often is an important measure. This accuracy is depending on reducible error and irreducible error [38]. The reducible error is the difference between f and the estimate $\hat{f}$ in Equation 2.2 [38]. The irreducible error exists due to the fact that Y also is a function of the random error (ϵ), as seen in Equation 2.1 [38].

The most flexible model is not always the model with the most accurate prediction, because of overfitting [38]. A performance measure much used for regression problems, is the mean squared error (MSE) [38]. The MSE can both be calculated on the training data and the test data. Overfitting is when the flexibility of a model is increased to a level where the MSE of the training data is decreased, but the MSE of the test data is increased. Overfitting is also an issue in classification problems [38].

Accuracy, sensitivity, specificity, precision and F1 are performance measures

that have a score between zero and one, where one is the best possible performance [50]. The measurements are based on true positives (TP), true negatives (TN), false positives (FP) or false negatives (FN). Positives can for example be people tested positives for a disease, which will be used as an example in the following.

Accuracy is how many of the predictions that are correct and is calculated by Equation 2.11 [51]. Sensitivity is calculated in Equation 2.12, and measures how many of the sick people tested that tests positive [50]. Specificity is the amount of healthy people that are tested negative, as calculated in Equation 2.13 [50]. Precision is how many of the people tested positive that is actually sick as calculated in Equation 2.14 [50]. F1 is a combination of sensitivity and precision and the calculation is shown in Equation 2.15 [4]. Matthews correlation coefficient (MCC) in Equation 2.16 is a performance measure that can be used to evaluate binary classification predictions on imbalanced datasets, and has a score between minus one and one [4].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.11}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{2.12}$$

$$Specificity = \frac{TN}{TN + FP} \tag{2.13}$$

$$Precision = \frac{TP}{TP + FP} \tag{2.14}$$

$$F1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{2.15}$$

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}} \tag{2.16}$$

2.4 Genetic algorithms

Genetic algorithms (GA) are initiated with a population and use selection and recombination operators to generate new sample points that are optimal values of a function [23; 5]. John Holland and his students first introduced the genetic algorithm [52]. The computational models are inspired by evolution. When the algorithms have started with a population of chromosomes, different reproductive opportunities are evaluated, and chromosomes with a better solution to the target problem are more likely to be conserved [23]. Sklearn-genetic is a genetic feature selection module from scikit learn [5].

Chapter 3

Method

The aim of this thesis was to create models that predict from methylation profiles whether a person had T1D and identify possible T1D associated genes. The models were created using different ML algorithms, and the training data was a dataset consisting of T1D methylation profiles generated by Rakyan et al. and Bell er al. [3; 22; 53].

3.1 Data

A dataset was composed of methylation profiles generated by Rakyan et al. and Bell et al. [3; 22; 53]. Rakyan et al. made genome-wide DNA methylation profiles out of purified $CD14+$ monocytes from twin pairs where one of the two had T1D, and control pairs where none of them had T1D [3]. $CD14+$ monocytes are a cell type associated with T1D onset. 27,578 CpG sites for 100 individuals were included in the dataset. 68 individuals did not have T1D. Bell et al. made DNA methylation profiles across the same CpG sites from 195 individuals with T1D using the same overall design. In this study the case had T1D and nephropathy

while the controls had T1D and no renal disease.

From both studies, the series matrix text file was downloaded. The text file from Rakyan et al. was saved as 'twin_study.txt' and the heading line, the line with sample titles and the lines with the values were kept unchanged. The same lines were kept from the Bell et al. file, and named 'renal_study.txt'. In this project, the data from the two studies were put together in one dataset using the python file 'create_twinAndRenal.txt2.py' found on a GitHub page [18]. All code referred to is found on this page. This code reads in the two text files with the Python tool Pandas, and writes a new text file named 'twinAndRenal_study.txt'. The 'null' values and the methylation values that were left out were changed to zero. In addition, the code created the text file 'cg.txt' with only one name of a CpG-site on each line in the same order. A flow chart that shows input and output from various files is shown in Figure 3.1.

The new text file 'twinAndRenal_study.txt' was being read in in the code 'linesToLists.py' [18]. This code sorted the individuals such that those with T1D came first, followed by those without. Individuals with more than 5% values of zero were discarded. CpG sites including invalid methylation values, such as '$-3.40 \cdot 10^{38}$', were removed. After this a new text file 'inputML.txt' was written, and included two empty lines followed by 27,006 lines for each CpG site. The names of the CpG sites were not included. The 226 first columns were individuals with T1D, and the 68 last were individuals without T1D. The file was converted to a comma-separated values (CSV) file.

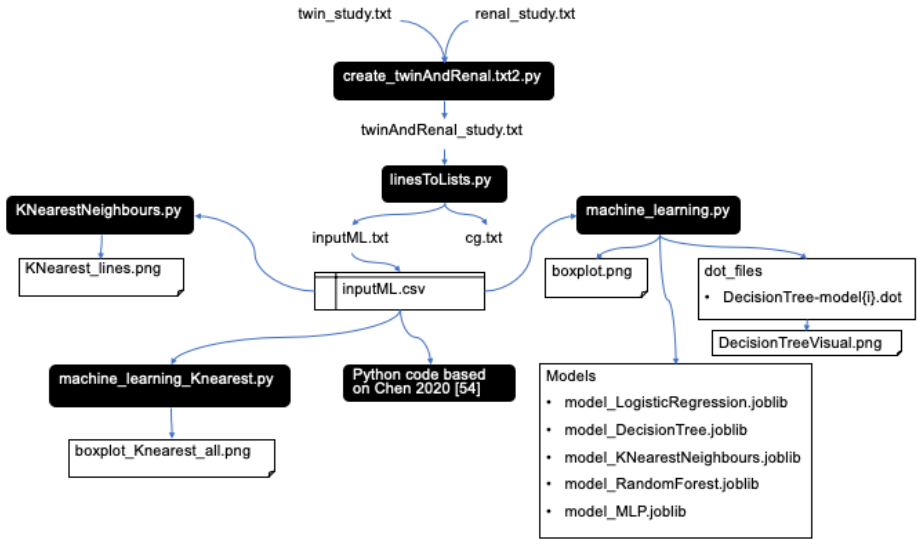


Figure 3.1: A flow chart that shows input and output from various files.

3.2 Methods

The ML algorithms logistic regression, decision tree, K-nearest neighbours, random forest and multilayer perceptron (MLP) were all compared with the CSV file as input data. All algorithms were classifiers, as the response data were binary. In order to decide which K-value to use in K-nearest neighbours, all K-values from 1 to 50 were compared. The Python code 'KNearestNeighbours.py' read the file 'inputML.csv'. KNeighborsClassifier from the scikit learn library for ML in Python was applied [44; 18]. 80% of the data chosen at random were used to train the model, and 20% to test it. The model was run 30 times for each K-value. Every run the model was tested and accuracy, sensitivity, specificity, precision, F1 and Matthews correlation coefficient were calculated. The averages of each K-value were calculated. The score was plotted against K-value with the plotting library for Python Matplotlib. A polynomial curve fitting was also carried out by

using the Python package numpy.

The Python code 'machine_learning_KNearest.py' also used KNeighborsClassifier with 'inputML.csv' as input [18]. It trained and tested models in the same way as the code mentioned previously, with 30 models for each K-value from $1-30$. In addition to include a smaller range of K-values, this approach also created box plots for each of the six performance measures. The box plots were created using the Python data visualisation library Seaborn, together with Matplotlib.

After investigating which K-value resulted in better model performance, all the mentioned ML algorithms were used to train models using the same CSV file. This was executed by the Python code 'machine_learning.py' [18]. Six classifiers from the scikit learn library were used, LogisticRegression, DecisionTreeClassifier, KNeighborsClassifier, RandomForestClassifier and MLPClassifier [44]. The decision tree used the Gini impurity as split criteria. The default parameters were used in all algorithms, except the K-value in K-nearest neighbours was sat to 15. It was trained 30 models of each. A randomly drawn subset covering 80% of the data were used for training, and the remaining data were used for testing. The last of the 30 models for each ML algorithm was saved with the Python tool kit joblib. The last model trained using decision tree, was visualised with export_graphviz from scikit learn. Again, accuracy, sensitivity, specificity, precision, F1 and MCC were calculated. Boxplots of the different performance measures for the ML algorithms were made using the Python data visualisation library Seaborn.

Two methods were used to find feature selections that alone trained the best models. The code used to execute this was based on code by Chen and the original paper by Chicco and Jurman [54; 55]. From Chen's Colab notebook, the

sections Data Preprocessing, Logistic Regression Baseline, Logistic Regression + GA and Logistic Regression + Select From Model, were run with adjustments [54]. The data pre-processing after the imports was adjusted to the dataset, and shown in Listing 1 in Appendix A [54].

With these adjustments, the code read in the file 'inputML.csv' and started by defining variables for which ML algorithm and population size that should be used, as shown in Listing 1 in Appendix A. This can be changed to different ML algorithms and population sizes. The previously described ML algorithms were used, except for the neural network algorithm MLP, because it was time consuming and had not performed better than the other ML algorithms. This time the random forest algorithm had a maximum depth of 3 instead of no maximum, and the solver of logistic regression was 'liblinear' instead of 'lbfgs'.

Sklearn-genetic was the first method used as feature selection module to find which smaller group of parameters that alone trained the best models [5]. GeneticSelectionCV was used with parameters listed in Table 3.1.

Table 3.1: GeneticSelectionCV was used with the parameters listed.

Parameter	Value
crossover probability	0.5
mutation probability	0.2
generations	10
crossover independent probability	0.5
mutation independent probability	0.1
gen. no. change	10
scoring	MCC
ML algorithms	LogisticRegression(solver = "liblinear") DecisionTreeClassifier() KNeighborsClassifier(n_neighbors=(15)) RandomForestClassifier(max_depth=3) MLPClassifier()
Populations	10 1000

The Scikit learn feature selection SelectFromModel was also used. It was done with populations of 10 and 1000, both with a max population of 25.

The features selected by GeneticSelectionCV and SelectFromModel were given as indices. The Python code 'IndexToCg.py' used 'cg.txt' to translate the indices in to CpG sites [18].

Chapter 4

Results and analysis

The created models to predict T1D will be presented in the following. The selections created with GA and the Scikit learn approach for feature selection will also be listed. To identify possible T1D associated genes, further analyses of these results were also conducted.

4.1 Results from Machine Learning

Figures 4.1 and 4.2 show how different K-values in K-nearest neighbours affected the performance score. The results were used to decide that 15 should be used as K-value, as there were K-values in a wide range around it with low variance and bias.

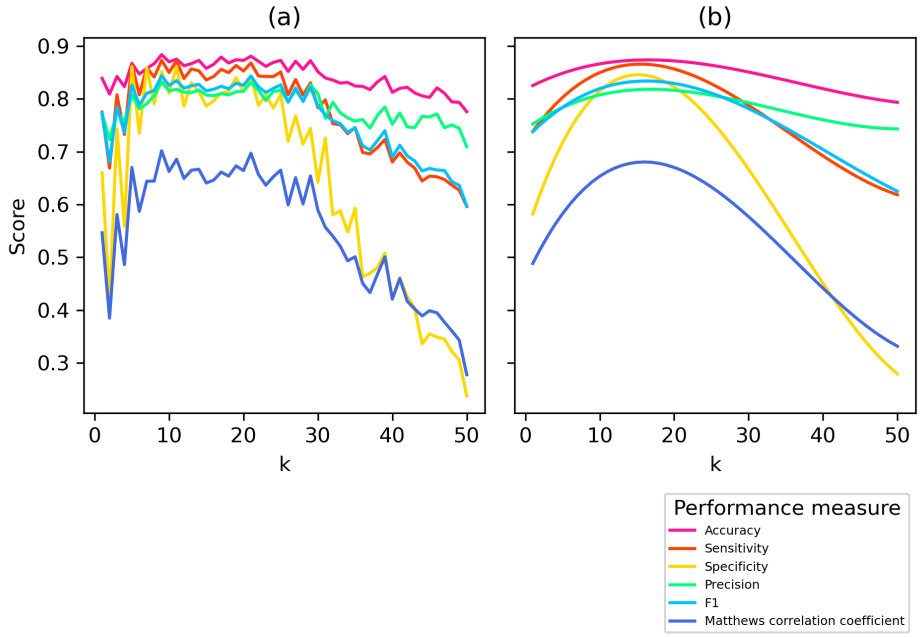


Figure 4.1: Different K-values in K-nearest neighbours affected the performance score. (a) The different performance scores were plotted against K-values between 1 and 50. (b) The same points were fitted to a polynomial curve.

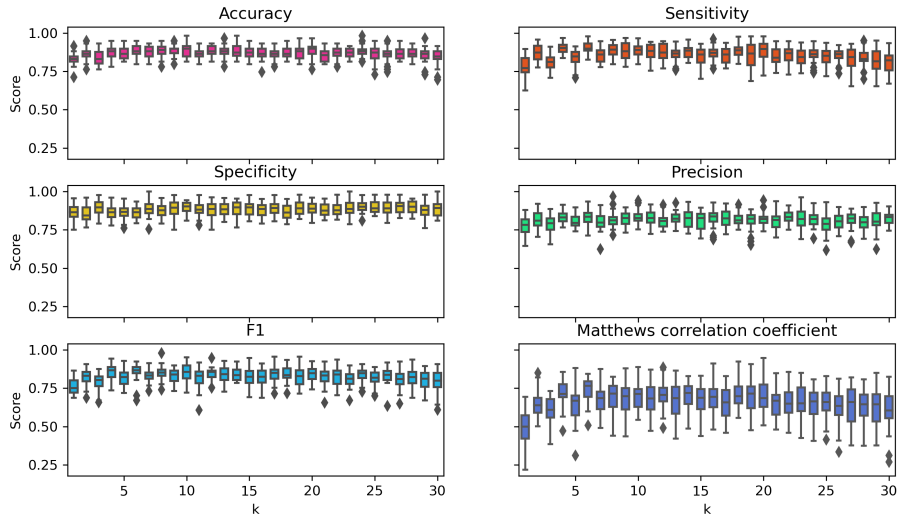


Figure 4.2: Different K-values in K-nearest neighbours affected the performance score. In separate windows, the different performance scores were plotted as box plots against K-values between 1 and 30.

A K-value of 15 was set for the K-nearest neighbours algorithm, and all models trained with 'machine_learning.py' are available at FigShare archives [56; 57; 58; 59; 60]. The model trained using the decision tree classifier is also visualised in Figure 4.3. The box plots created as a result of 30 of each model type is shown in Figure 4.4.

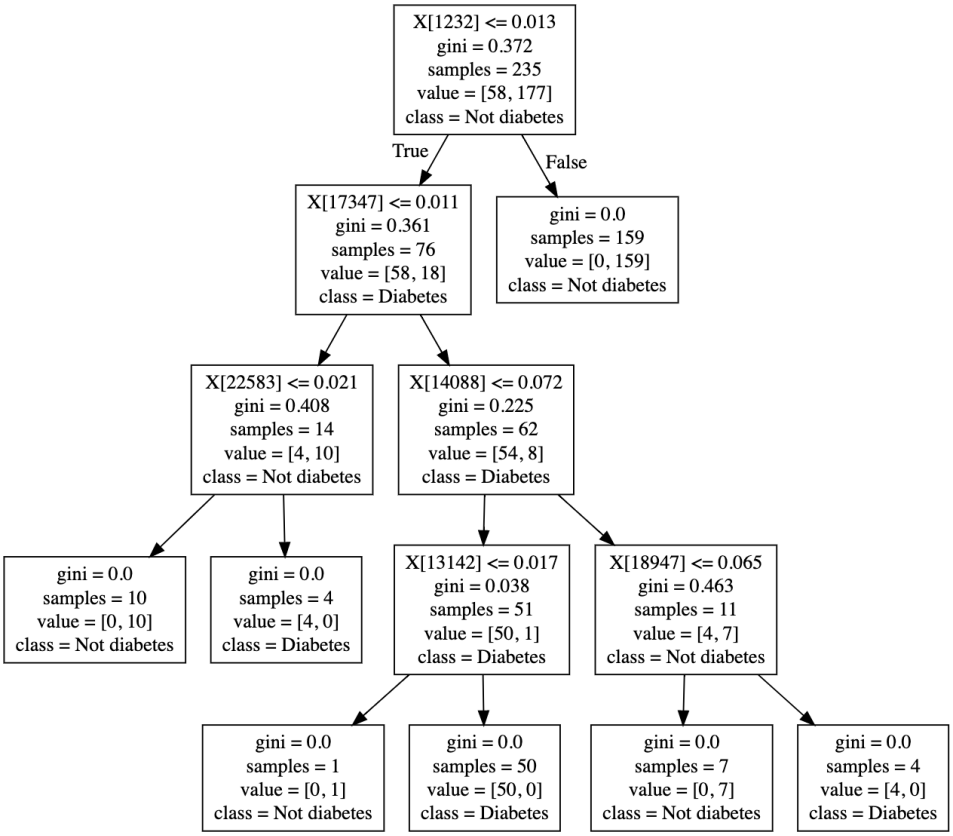


Figure 4.3: Decision tree classifier created with 'machine_learning.py' and visualised using Visual Studio Code [18].

Table 4.1: The corresponding CpG sites to the indices in the Decision tree in Figure 4.3

Index	CpG site
1232	cg01249910
17347	cg17698505
. 22583	cg23090824
14088	cg14377370
13142	cg13467814
18947	cg19346899

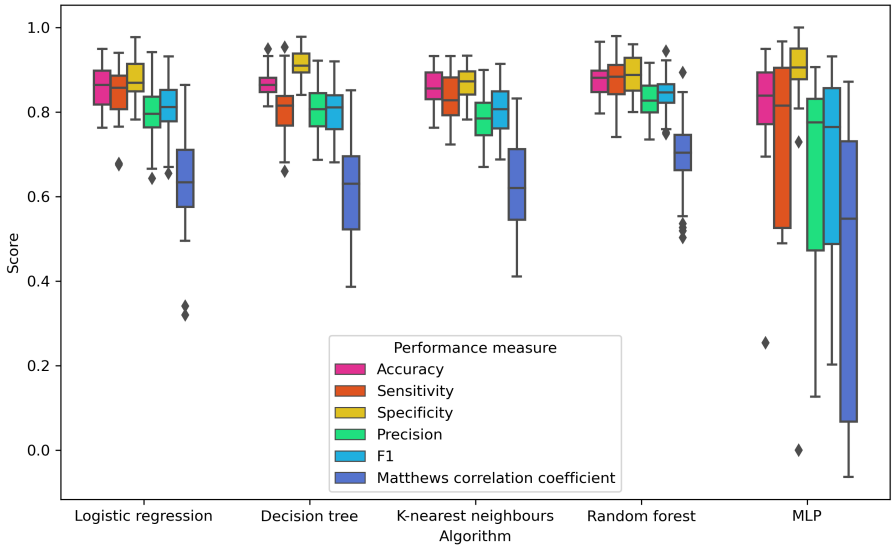


Figure 4.4: The box plots created as a result of 30 of each model type, with different kind of performance measures.

The results of the genetic selection and the select from model features selections are shown in Table 4.3 and 4.4 respectively. The MCC of the various algorithms without selection are shown in Table 4.2. The neural network algorithm MLP was not included as it did not give significantly better performance, and it was more time consuming. The K-nearest neighbours classifier was only included in the genetic selection, because it was not compatible with the Scikit learn feature selection. The K-nearest neighbours classifier missed the attributes 'coef_' and 'feature_importances_'.

Table 4.2: The MCC of the various algorithms without selection.

Without selection	Logistic regression	Decision tree	K-Nearest neighbours	Random forest
MCC	0.640	0.619	0.611	0.695

Table 4.3: The results of the genetic selection. CpG sites which exist in several selections are marked in **bold**. CpG sites which both exists in several selections and have T1D related functions are marked in in **blue**.

Genetic selection	Logistic regression	Decision tree	K-Nearest neighbours	Random forest
population: 10 max features: 4	**cg09814877**	cg10829134, cg13259290	cg06622725, cg07909422, cg13393195, cg22899145	cg04444006, cg06807379, cg11407345, cg22781236
MCC	0.000	0.570	0.745	0.669
population: 1000 max features: 4	**cg06270401**, cg14898639, cg15972617, cg17067005	cg01675895, cg07613278, cg15321195	cg11187508, cg18589858, cg21982518	cg13557178, cg18047509, **cg18830459**, cg23426587
MCC	0.725	0.740	0.787	0.775
population: 10 max features: 9	cg08744769, cg09478478, cg13676215, cg17425224, cg19515518, cg20699736, cg21184174, cg27541515	cg02233559, cg05566397, cg06911113, cg11263296, cg11826116, cg14560895, cg17997329, cg21874193, cg25261329	cg00903375, cg03985657, cg08823182, cg09396217, cg11266874, cg12266551, cg16306115, cg16414852, cg26952662	cg03900104, cg04536922, cg15750102, cg18671950, cg23710218, cg24034289
MCC	0.267	0.629	0.720	0.700
population: 1000 max features: 9	cg02217814, cg02273078, cg06084117, cg09837977, cg11021744, cg14802310, cg16050349, cg18236721, cg22511947	cg05517572, cg17797815, cg20053799, cg21142188, **cg26131019**	cg06501790, cg13235447, cg18084791, cg18674980, cg21295911, cg22800631, cg23055159, cg23307264, cg26128441	cg04177705, cg05501721, cg13232900, cg26306976
MCC	0.648	0.794	0.777	0.788
population: 10 max features: 16	cg03544320, cg04435420, cg04956790, cg11701148, cg15477600, cg15783027, cg20162076, cg24926276, cg26001030	cg02579736, cg03359508, cg03914452, cg05343453, cg08122545, cg08466074, cg09757107, cg12312863, cg15599064, cg19166347, cg20676475, cg20761322	cg00292971, cg02806658, cg04633513, cg13636404, cg15006973, cg18627308, cg21122774, cg22182945, cg24981018	cg00042156, cg04798158, cg06910100, cg06943865, cg07745725, cg08575950, cg10608333, cg10863038, cg17067993, cg18653991, cg23858565, cg26221631
MCC	0.471	0.557	0.712	0.688
population: 1000 max features: 16	cg01193293, cg02089348, cg02537023, cg05071677, cg06268694, **cg06270401**, cg07535475, cg07676849, cg10617171, cg11701148, cg11801374, cg12254515, cg15379858, cg15747133, cg22874560	cg01868128, cg02631957, cg02751839, cg02936468, cg04058169, cg04564646, cg10971346, cg15903282, cg19379303, cg23152667, cg26413355, cg26466094	cg07685869, cg15790852, cg16257040, cg18034859, cg20088913, cg22341310, cg24043192, cg26195577, cg26520371	cg00969271, cg05348272, **cg10762132**, cg21264055, cg23030090, **cg24407308**, **cg26131019**
MCC	0.781	0.758	0.792	0.799
population: 10 max features: 25	cg02153528, cg02254407, cg03430067, cg06114765, cg08977028, cg09238677, cg11884656, cg11887234, cg11928366, cg20485165, cg20913782, cg20925811, cg24965984, cg25577842	cg06796611, cg06849477, cg09924998, cg11429658, cg12622986, cg15134628, cg16512727, cg17018527, cg17612991, cg21784498, cg22064942, cg25459778	cg03160740, cg04413397, cg06071083, cg07009002, cg10185638, cg11943820, cg14643978, cg18943383, cg19786920, cg23050981, cg23123262, cg24603941	**cg02212836**, cg03046445, cg03548857, cg03955296, cg07786760, cg09394600, cg09656405, cg10076009, cg11298616, cg12780322, cg13271951, cg15981753, cg16386158, cg16802152, cg19464252, cg21715963, cg22391400, cg23333306, cg25836159
MCC	0.327	0.614	0.700	0.754
population: 1000 max features: 25	cg02168291, **cg02212836**, cg04115602, cg06728335, cg07873488, cg09288658, **cg09418321**, cg12374721, cg12477119, cg12600197, cg13236107, cg18826520, cg20857947, cg23495733, cg25306927, cg25600606, cg25985103, cg25993152, cg26147338, cg26646411	cg07750111, cg08291000, **cg10762132**, cg10971790, cg11909865, cg16004226, cg16313587, cg16714091, cg20828084, cg21038703, cg27555776	cg02060988, cg04545516, cg05208153, cg05362516, cg05473871, cg05484920, cg08524221, cg12085044, cg19012475, cg19058629, cg25219134, cg26177629, cg27400772	cg01986577, cg03116238, cg03128832, cg07772309, cg07778029, cg07933197, cg11821536, cg13281868, cg17605847, cg17661881, cg20788083, cg22108175, cg22913584, cg26845838
MCC	0.745	0.782	0.790	0.794

Table 4.4: The results of the select from model features selection. CpG sites which exist in several selections are marked in **bold**. CpG sites which both exists in several selections and have T1D related functions are marked in in **blue**.

Select from model	Logistic regression	Decision tree	Random forest
population: 10 max features: 25	cg00066153, **cg00795812**, cg01989224, cg03734783, cg04349727, cg04356968, **cg09814877**, cg10478221, cg13882988, cg14356114, cg14808739, cg14885742, cg14930674, cg17964955, cg17977362, cg20022541, cg21063899, cg22456522, cg23857226, cg24435704, cg24541550, cg25022327, cg26661481, cg27105123, cg27238470	cg01464985, cg02718725, cg04619381, cg13725272, cg19912436, cg21602160, cg25040733, cg25104511, **cg26131019**	**cg00795812**, cg04197051, cg04523589, **cg06270401**, cg06725035, cg06913228, cg09197965, **cg10762132**, cg11564268, cg11856918, cg12998491, cg13451483, cg14781919, cg15039399, cg15379858, cg15690721, cg16538604, cg17091770, cg17655576, cg18085206, cg21235838, cg24594997, cg25788012, **cg26131019**, cg26831968
MCC	0.718	0.891	0.771
population: 1000 max features: 25	cg00066153, **cg00795812**, cg01989224, cg03734783, cg04349727, cg04356968, **cg09814877**, cg10478221, cg13882988, cg14356114, cg14808739, cg14885742, cg14930674, cg17964955, cg17977362, cg20022541, cg21063899, cg22456522, cg23857226, cg24435704, cg24541550, cg25022327, cg26661481, cg27105123, cg27238470	cg01464985, cg02718725, cg11465372, cg17847607, cg19531130, cg25040733, cg25104511, cg26060255, **cg26131019**	cg00824109, cg01718139, cg02047577, cg03157149, cg08242020, cg08335125, cg09076012, **cg09418321**, cg09475757, cg14698961, cg14714578, cg14781919, cg15379858, cg16829154, cg17896097, cg18264687, **cg18830459**, cg18986273, cg20319264, cg21101222, cg22305782, cg23508786, cg24364574, **cg24407308**, **cg26131019**
MCC	0.731	0.911	0.781

4.2 CpG sites that exist in several selections

Nine of the CpG sites were found in more than one selection in Table 4.3 and 4.4. They are listed in Table 4.5 and marked in Table 4.3 and 4.4. It was not included if it was only repeated in the same ML algorithm, same type of selection and same number of maximum features, but with different population sizes. *cg*26131019 was included in six selections, which was the most recurring.

Table 4.5: The table lists the CpG sites found in more than one selection. It was not included if it was only repeated in the same ML algorithm, same type of selection and same number of maximum features, but with different population sizes.

Candidates of important CpG sites based on ML	Number of selections
cg26131019	6
cg09814877	3
cg06270401	3
cg10762132	3
cg00795812	3
cg18830459	2
cg24407308	2
cg02212836	2
cg09418321	2

4.3 Comparison with CSD results

The author's previous project was to use parts of the same dataset to execute a differential co-expression network analysis using software programs called CSD developed by Voigt et al. [20; 8]. The resulting network was analysed in accordance with theory presented and discussed in the report, and possible T1D associated network patterns were identified [8]. Candidates for important CpG sites had more than five neighbours in the network, together with relatively high values of other centrality measures, and were listed in Table 4.6.

Table 4.6: The table lists the nodes (CpG sites) with the highest degrees k_i, which means the highest number of neighbours. The node with the highest degree is found in the biggest component of 212 nodes. The CpG sites cg10031456 and cg09088193 are neighbours, and is found in the second biggest component of 45 nodes [8].

Candidates of important CpG sites based on CSD analysis
cg09736162
cg23173455
cg04542415
cg10031456
cg02946754
cg07588113
cg00067471
cg09088193
cg04348872

The CpG sites used to make the best performing models were compared with the central CpG sites from the CSD analysis. The CpG sites listed in Table 4.6 were searched for among all the selections created by Genetic algorithm, Select from model and those used to create the decision tree classifier. In addition, the CpG sites from the decision tree and those from the selections that had MCC greater than 0.8 were searched for in the entire CSD network. No matches were found.

4.4 Biological function

To access biological functions, CpG sites were first converted to gene names, using a data table published by Illumina Inc. [61]. The Python code 'cgDict.py' was used to execute this conversion [18]. 25,450 out of the 27,006 CpG sites were found in the table. Biological functions and processes of the nine CpG sites repeated in several selections were accessed with the NCBI gene tool and

literature search, as seen in Table 4.7 [9; 10; 6; 11; 12; 7; 13; 14].

Table 4.7: Biological functions and processes of the nine CpG sites repeated in several selections listed in Table 4.5. They were accessed with the NCBI gene tool and literature search [9; 10; 6; 11; 12; 7; 13; 14]. T1D related functions are marked in in blue.

CpG site	Gene name	Function
cg26131019	LRIG1	Enables protein binding Involved in hair cycle process, innervation, otolith morphogenesis and sensory perception of sound
cg09814877	ACPT	Little is known Significance in enamel maturation [10]
cg06270401	DYRK4	Enables kinase activity [6]
cg10762132	SLC20A1	Enables sodium:phosphate symporter activity
cg00795812	PDCD1	Enables protein binding Has demonstrated to play a role in anti-tumor immunity
cg18830459	RNF19B	Enables protein (ubiquitin) binding
cg24407308	DGKZ	Enables ATP binding Enables NAD+ kinase activity Enables protein binding Involved in the pathway of leptin-insulin signaling overlap [11] From pathway presented by Wang et al., DGKZ was involved in the phosphatidylinositol signaling system [12]
cg02212836	LY86	Enables protein binding DNA methylation of the gene is associated with obesity and insulin resistance [7] LY86-AS1 could possibly be used as a diagnostic marker for type 2 diabetes [13]
cg09418321	DYRK4 [14] (Same as gene further above)	See further above

Wang et al. created the disease-associated lncRNA-mRNA-pathway network with a weighed gene coexpression network approach as seen in Figure 4.5 [12].

The network suggested that FASN gene was part of the insulin signalling pathway and DGKZ gene was part of the phosphatidylinositol signalling system, and that both genes were coronary artery disease (CAD) progression-associated [12]. The DGKZ gene is one of the genes repeated in several ML feature selections.

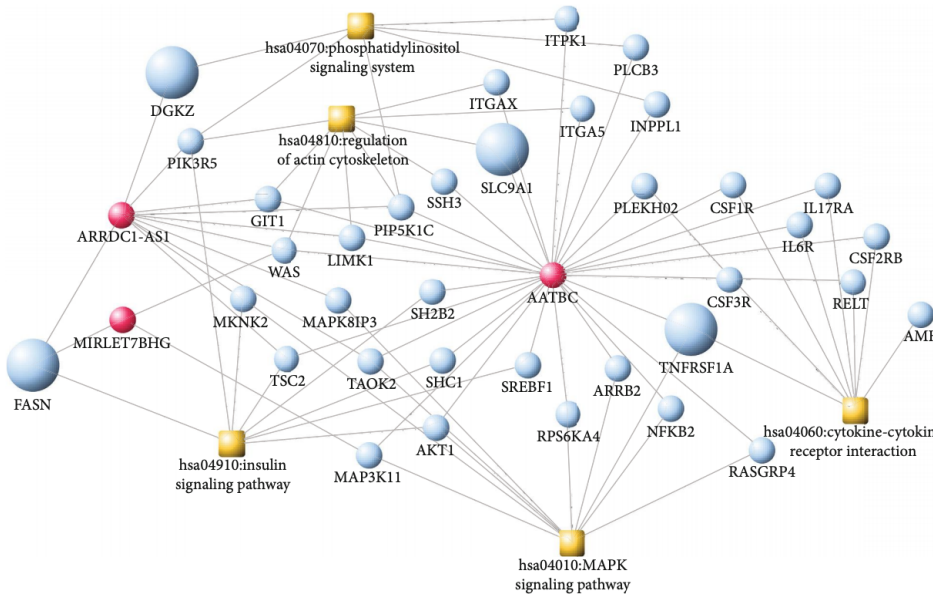


Figure 4.5: Wang et al. created the disease-associated lncRNA-mRNA-pathway network with a weighed gene coexpression network approach [12]. The red dots were lncRNAs, the blue dots were mRNAs, the orange square were the disease pathway, and the bigger nodes were the disease genes [12].

Chapter 5

Discussion

Models were trained using several ML algorithms both for prediction and inference. It was a goal to predict whether a person had T1D or not given his or her methylation profile. Additionally, which CpG sites that were key predictors, was interesting. In both cases, it should be discussed which kind of performance measure that is relevant.

MCC is a performance measure that can be used to evaluate binary classification predictions on imbalanced datasets, and therefore a proper measure [4]. A dataset consisting of 226 individuals with T1D and then 68 without T1D can be considered imbalanced. Even though MCC may be the most relevant overall measure, sensitivity and specificity may be useful to see how many of the sick people tested that tests positive, and the amount of healthy people that are tested negative. Nonetheless, MCC will cover and balance both considerations.

The algorithms performed evenly when looking at MCC, as seen in Figure 4.4. However, the random forest algorithm scored slightly higher, meaning the other algorithms had some more bias. The MLP algorithm had the highest bias, con-

sidering MCC. The MLP algorithm also had the highest variance. This algorithm scored higher at specificity than sensitivity, thus if sensitivity is important, the MLP algorithm should be avoided using this dataset.

As the ML algorithms were about equally good, the input dataset may be more important. This is in accordance with the fact that there is often improvement in scaling up training data sets in current ML systems [31]. The selections results were also supporting the assumption that input data is more important than the algorithm. With genetic selection and selection from model, MCC was higher on selections with a population of 1000 than of 10 with same parameters except that.

However, the MCC was not always higher with the highest number of maximum features, as seen in Table 4.3. This may be so because finding some few features that were good predictors, reduced the problem of overfitting. That did not mean that a maximum number of features of 4 resulted in higher MCC than a maximum number of features of 25. At some point it would turn over to underfitting.

In Table 4.3 and 4.4 it appears that selections which include CpG sites also included in other selections, did not necessarily have a high MCC. This may indicate that important T1D associated CpG sites also exist in the selections that did not provide the best predictions.

The LY86 gene was found in two selections created with genetic selection. In literature search for biological functions, it was found that DNA methylation of the gene is associated with obesity and insulin resistance according to Su et al. [7]. They studied genes associated with diabetes, and observed methylation of these genes, namely forward genetics. Reverse genetics is when the functional study of a gene starts with the gene sequence and not the phenotype [62]. In

the current study the genes that were associated with diabetes were not picked out in advance, and the study was more in the direction of reverse genetics. In spite of that, the dataset was labelled and the aim was not to identify what was the phenotype of a given structure, but rather if some of the structures were associated with T1D. The current results may strengthen the discovered result by Su et al. that DNA methylation of the LY86 gene is associated with insulin resistance. However, T1D patients have insulin deficiency, they are not insulin resistant. Both scenarios may be associated with DNA methylation of the LY86 gene.

The DGKZ gene was found in two of the selections. The gene was involved in the pathway of leptin-insulin signalling overlap [11]. Leptin has a role in the appetite regulation, but also in the control of the peripheral insulin and glucose responsiveness [63]. Leptin gene therapy on insulin-deficient diabetes in obesity animal models such as T1D mice, has shown good results [63]. Additionally, the DGKZ gene was close to the insulin signalling pathway, in Figure 4.5, which may indicate a correlation [12].

Several of the nine CpG sites that were in multiple selections, were associated with T1D. It may be investigated further whether even more of them are associated with T1D. $cg06270401$ and $cg09418321$ that both were among the nine, were associated with the same gene, namely DYRK4. They may be candidates for T1D relevance. $cg26131019$ which is associated with the LRIG1 gene existed in six selections and may also be associated with T1D.

Nine of approximately 27,000 CpG sites were picked out in the current study. Two of those had biological functions that could be critical for developing T1D [12; 11; 7]. The approach has been suitable for detecting T1D associated CpG sites. Selections found when using random forest and to some extend also lo-

gistic regression, seemed to give the best results, because most of the candidates for T1D associated genes were found using these algorithms. To reach the full potential of the approach, optimisation of various parameters are further discussed.

The logistic regression algorithm did perform approximately equal to 'liblinear' as solver as seen in Table 4.3 and 'lbfgs' as seen in Figure 4.4. The default is 'lbfgs', and previously it was 'liblinear' [5]. The penalty L1 has more sparse properties than L2 and is suitable for datasets with a large number of features [43]. It is therefore suggested to use L1 if no feature selection is completed with for example genetic algorithms.

A parameter that might increase the performance of the decision tree classifier, could be the maximum depth of the trees. No maximum depth was fixed on the decision tree classifiers in this study. Therefore, nodes expanded until all leaves were pure, or all leaves contained less than the minimal samples split, which was sat to two. To avoid overfitting, smaller integers can be tested as the maximum depth. Random forest algorithms consists of several decision trees, and therefore the same applies for maximum depth in random forest algorithms. No maximum depth was compared to a maximum depth of 3, but resulted in little change in MCC. An even smaller maximum depth might have given higher performance.

The CpG sites used to make the best performing models were compared with the central CpG sites from the CSD analysis, but no matches were found [8]. The results may not be comparable, because the CSD method detects correlations. The inference of the ML algorithms has not been studied beyond finding which predictors are associated with the response. The CSD method may be better in detecting patterns that characterise disease than finding genes that alone can

be used as diagnostic markers for a disease.

The CSD method divides a dataset in conditions, which for example can be individuals with and without diabetes. If the dataset is not labelled, unsupervised ML trained models can be used to divide the data in conditions.

Chapter 6

Conclusion

The aim of the thesis was to create models to predict from methylation profiles whether a person had T1D and identify possible T1D associated genes. The training of models was executed by ML algorithms.

The ML classifiers forms of logistic regression, decision tree, K-nearest neighbours, random forest and MLP were compared. After pre-processing, the dataset was a CSV file consisting of 27,006 lines of CpG sites, 226 columns representing T1D individuals and 68 columns representing individuals without T1D. The values were degree of methylation, which was a value between zero and one. By comparing trained K-nearest neighbours models with different K-values, a K-value of 15 was considered optimal. The ML algorithms performed evenly high with an average MCC at around 0.65. The even results indicated that training data may be more important than the choice of model. It seems that it is best to have a large training dataset and spend most of the time and resources on pre-processing data. However, in order to avoid overfitting in parametric methods it can be an advantage to create feature selections or adjust ML algorithm param-

eters that determine how many features to use in the model. Feature selections were created using the GA Scikit learn-genetic and the Scikit learn approach for feature selection [5; 44]. Models trained with a feature selection only, tend to score higher. Nine CpG sites were found in more than one feature selection. Among these, two were T1D associated. The approach has been suitable for detecting T1D associated CpG sites. One of the T1D associated CpG sites is associated with the LY86 gene. DNA methylation of the LY86 gene is associated with insulin resistance [7]. Given the identified results, DNA methylation of the LY86 gene appears to be associated with insulin deficiency as well. All nine CpG sites were considered candidates for T1D relevance.

Bibliography

[1] Mobasseri M, Shirmohammadi M, Amiri T, Vahed N, Fard HH, Ghojazadeh M. Prevalence and incidence of type 1 diabetes in the world: A systematic review and meta-analysis. Tabriz University of Medical Sciences; 2020. Available from: https://doi.org/10.34172/hpp.2020.18.

[2] Dabelea D, Mayer-Davis EJ, Saydah S, Imperatore G, Linder B, Divers J, et al. Prevalence of type 1 and type 2 diabetes among children and adolescents from 2001 to 2009. JAMA - Journal of the American Medical Association. 2014. Available from: https://doi.org/10.1001/jama.2014.3201.

[3] Rakyan VK, Beyan H, Down TA, Hawa MI, Maslau S, Aden D, et al. Identification of type 1 Diabetes-associated DNA methylation variable positions that precede disease diagnosis. PLoS Genetics. 2011;7(9):1–9. Available from: https://doi.org/10.1371/journal.pgen.1002300.

[4] Chicco D, Jurman G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evalua-

tion. BMC Genomics. 2020 jan;21(1). Available from: https://doi.org/10.1186/s12864-019-6413-7.

[5] Calzolari M. manuel-calzolari/sklearn-genetic: sklearn-genetic 0.4.0 (Version 0.4.0). Zenodo; 2021. Available from: https://doi.org/10.5281/zenodo.4661178.

[6] Papadopoulos C, Arato K, Lilienthal E, Zerweck J, Schutkowski M, Chatain N, et al. Splice variants of the dual specificity tyrosine phosphorylation-regulated kinase 4 (DYRK4) differ in their subcellular localization and catalytic activity. Journal of Biological Chemistry. 2011;286(7). Available from: https://doi.org/10.1074/jbc.m110.157909.

[7] Su S, Zhu H, Xu X, Wang X, Wang X, Dong Y, et al. DNA methylation of the LY86 gene is associated with obesity, insulin resistance, and inflammation. Twin Research and Human Genetics. 2014;17(3). Available from: https://doi.org/10.1017/thg.2014.22.

[8] Mello A. A network modelling approach to investigate methylation profiles for type 1 diabetics; 2020. Available from: https://github.com/amaliemello/Masters-thesis/blob/main/AmalieMelloSpecialisationProject.pdf.

[9] NCBI. Gene;. Available from: https://www.ncbi.nlm.nih.gov/gene/.

[10] Mu Y, Huang X, Liu R, Gai Y, Liang N, Yin D, et al. ACPT gene is inactivated in mammalian lineages that lack enamel or teeth. PeerJ. 2021;9. Available from: https://doi.org/10.7717/peerj.10219.

[11] WikiPathways. PubChem Pathway Summary for Pathway WP3935, Leptin-

insulin signaling overlap. National Center for Biotechnology Information. 2021.

[12] Wang L, Hu J, Zhou J, Guo F, Yao T, Zhang L. Weighed Gene Coexpression Network Analysis Screens the Potential Long Noncoding RNAs and Genes Associated with Progression of Coronary Artery Disease. Computational and Mathematical Methods in Medicine. 2020;2020. Available from: https://doi.org/10.1155/2020/8183420.

[13] Saeidi L, Ghaedi H, Sadatamini M, Vahabpour R, Rahimipour A, Shanaki M, et al. Long non-coding RNA LY86-AS1 and HCG27_201 expression in type 2 diabetes mellitus. Molecular Biology Reports. 2018;45(6). Available from: https://doi.org/10.1007/s11033-018-4429-8.

[14] Neumeyer S, Popanda O, Edelmann D, Butterbach K, Toth C, Roth W, et al. Genome-wide DNA methylation differences according to oestrogen receptor beta status in colorectal cancer. Epigenetics. 2019;14(5). Available from: https://doi.org/10.1080/15592294.2019.1595998.

[15] Raschka S, Mirjalili V. Python Machine Learning: Machine Learning & Deep Learning with Python, Scikit-Learn and TensorFlow 2, Third Edition. January 2010; 2019.

[16] Fisher RA. Iris Data Set; 1936. Available from: https://archive.ics.uci.edu/ml/datasets/Iris.

[17] Montavon G. Introduction to Neural Networks. In: Lecture Notes in Physics. vol. 968; 2020. Available from: https://doi.org/10.1007/978-3-030-40245-7{_}4.

[18] Mello A. github.com/amaliemello/Masters-thesis; 2021. Available from: https://github.com/amaliemello/Masters-thesis.

[19] Kitano H. Computational systems biology; 2002. Available from: https://doi.org/10.1038/nature01254.

[20] Voigt A, Nowick K, Almaas E. A composite network of conserved and tissue specific gene interactions reveals possible genetic interactions in glioma. vol. 13; 2017. Available from: https://doi.org/10.1371/journal.pcbi.1005739.

[21] Tibshirani R. Regression Shrinkage and Selection Via the Lasso. Journal of the Royal Statistical Society: Series B (Methodological). 1996. Available from: https://doi.org/10.1111/j.2517-6161.1996.tb02080.x.

[22] Bell CG, Teschendorff AE, Rakyan VK, Maxwell AP, Beck S, Savage DA. Genome-wide DNA methylation analysis for diabetic nephropathy in type 1 diabetes mellitus. BMC Medical Genomics. 2010;3. Available from: https://doi.org/10.1186/1755-8794-3-33.

[23] Whitley D. A genetic algorithm tutorial. Statistics and Computing. 1994 jun;4(2):65–85. Available from: https://doi.org/10.1007/BF00175354.

[24] Atkinson MA, Eisenbarth GS, Michels AW. Type 1 diabetes; 2014. Available from: https://doi.org/10.1016/S0140-6736(13)60591-7.

[25] Usher-Smith JA, Thompson MJ, Sharp SJ, Walter FM. Factors associated with the presence of diabetic ketoacidosis at diagnosis of diabetes in chil-

dren and young adults: A systematic review; 2011. Available from: https://doi.org/10.1136/bmj.d4092.

[26] Rorsman P, Renström E. Insulin granule dynamics in pancreatic beta cells; 2003.

[27] Kerner W, Brückel J. Definition, classification and diagnosis of diabetes mellitus; 2014. Available from: https://doi.org/10.1055/s-0034-1366278.

[28] Daneman D. Type 1 diabetes. In: Lancet; 2006. Available from: https://doi.org/10.1016/S0140-6736(06)68341-4.

[29] Strober BJ, Elorbany R, Rhodes K, Krishnan N, Tayeb K, Battle A, et al. Dynamic genetic regulation of gene expression during cellular differentiation. Science. 2019;364(6447). Available from: https://doi.org/10.1126/science.aaw0040.

[30] Ravasi T, Suzuki H, Cannistraci CV, Katayama S, Bajic VB, Tan K, et al. An Atlas of Combinatorial Transcriptional Regulation in Mouse and Man. Cell. 2010;140(5):744–752. Available from: https://doi.org/10.1016/j.cell.2010.01.044.

[31] Zhu H, Wang G, Qian J. Transcription factors as readers and effectors of DNA methylation. Nature Reviews Genetics. 2016;17(9):551–565. Available from: http://dx.doi.org/10.1038/nrg.2016.83.

[32] Bestor TH. DNA methylation: evolution of a bacterial immune function into a regulator of gene expression and genome structure in higher eukaryotes.; 1990. Available from: https://doi.org/10.1098/rstb.1990.0002.

[33] Bird A. Epigentic Memory. Genes and Development. 2002;16:16–21. Available from: http://www.genesdev.org/cgi/doi/10.1101/gad.947102.

[34] Harvey S, Harvey R. An introduction to artificial intelligence. Appita Journal. 1998;51(1).

[35] Rouse M, Tucci L, Burns E, Laskowski N. Definition Artificial intelligence; 2020. Available from: https://searchenterpriseai.techtarget.com/definition/AI-Artificial-Intelligence.

[36] Iniesta R, Stahl D, McGuffin P. Machine learning, statistical learning and the future of biological research in psychiatry. Cambridge University Press; 2016. Available from: https://doi.org/10.1017/S0033291716001367.

[37] Haddaway NR, Callaghan MW, Collins AM, Lamb WF, Minx JC, Thomas J, et al. On the use of computer-assistance to facilitate systematic mapping. Campbell Systematic Reviews. 2020 dec;16(4). Available from: https://doi.org/10.1002/cl2.1129.

[38] James G, Witten D, Hastie T, Tibishirani R. An Introduction to Statistical Learning with Applications in R (older version); 2013.

[39] Badillo S, Banfai B, Birzele F, Davydov II, Hutchinson L, Kam-Thong T, et al. An Introduction to Machine Learning. Clinical Pharmacology and Therapeutics. 2020 apr;107(4):871–885. Available from: https://doi.org/10.1002/cpt.1796.

[40] Schmidt AF, Finan C. Linear regression and the normality assumption. El-

sevier USA; 2018. Available from: https://doi.org/10.1016/j.jclinepi.2017.12.006.

[41] Wu S, Flach Pa. Feature Selection with Labelled and Unlabelled Data. ECML/PKDD'02 workshop on Integration and Collaboration Aspects of Data Mining, Decision Support and Meta-Learning. 2002.

[42] Friedman J, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software. 2010;33(1):1–22. Available from: https://doi.org/10.18637/jss.v033.i01.

[43] Zeng S, Gou J, Deng L. An antinoise sparse representation method for robust face recognition via joint l1 and l2 regularization. Expert Systems with Applications. 2017;82. Available from: https://doi.org/10.1016/j.eswa.2017.04.001.

[44] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research. 2011 oct;12:2825–2830.

[45] Le TT, Moore JH. treeheatr: an R package for interpretable decision tree visualizations. Bioinformatics (Oxford, England). 2021 apr;37(2):282–284. Available from: https://doi.org/10.1093/bioinformatics/btaa662.

[46] Nguyen B, Morell C, De Baets B. Large-scale distance metric learning for k-nearest neighbors regression. Neurocomputing. 2016 nov;214:805–814. Available from: https://doi.org/10.1016/j.neucom.2016.07.005.

[47] Vadla PK, Ruwali A, Prakash KB, Lakshmi MVP, Kanagachidambaresan GR. Neural Network. In: EAI/Springer Innovations in Communication and Computing. Springer Science and Business Media Deutschland GmbH; 2021. p. 39–43.

[48] Burger HC, Schuler CJ, Harmeling S. Image denoising: Can plain neural networks compete with BM3D? In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 2012. p. 2392–2399.

[49] Gardner MW, Dorling SR. Artificial neural networks (the multilayer perceptron) - a review of applications in the atmospheric sciences. Atmospheric Environment. 1998;32(14-15). Available from: https://doi.org/10.1016/S1352-2310(97)00447-0.

[50] Bowers AJ, Sprott R, Taff SA. Do We Know Who Will Drop Out?: A Review of the Predictors of Dropping out of High School: Precision, Sensitivity, and Specificity. The High School Journal. 2013;96(2).

[51] Hung CW, Li WT, Mao WL, Lee PC. Design of a chamfering tool diagnosis system using autoencoder learning method. Energies. 2019 sep;12(19). Available from: https://doi.org/10.3390/en12193708.

[52] Holland JH. Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence; 1975.

[53] Teschendorff AE, Menon U, Gentry-Maharaj A, Ramus SJ, Weisenberger DJ, Shen H, et al. Age-dependent DNA methylation of genes that are suppressed in stem cells is a hallmark of cancer. Genome Research. 2010 apr;20(4):440–446.

[54] Chen P. Heart Failure; 2020. Available from: https://colab.research.google.com/drive/17NqqAoSm24N9a6nXLN2vzPxXkP8AlygM#scrollTo=gH6CKF5A5HU8&line=3&uniqifier=1.

[55] Chicco D, Jurman G. Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. BMC Medical Informatics and Decision Making. 2020;20(1). Available from: https://doi.org/10.1186/s12911-020-1023-5.

[56] Mello A. Logistic regression; 2021. Available from: https://doi.org/10.6084/m9.figshare.15074505.v1.

[57] Mello A. Decision tree; 2021. Available from: https://doi.org/10.6084/m9.figshare.15074832.v1.

[58] Mello A. Random forests; 2021. Available from: https://doi.org/10.6084/m9.figshare.15074922.v1.

[59] Mello A. K-nearest neighbours; 2021. Available from: https://doi.org/10.6084/m9.figshare.15074901.v1.

[60] Mello A. MLP; 2021. Available from: https://doi.org/10.6084/m9.figshare.15074925.v1.

[61] NCBI. Illumina HumanMethylation450 BeadChip; 2011. Available from: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GPL13534.

[62] Girard L, Chan J, Kenny E, Sternberg P, Stein L, Chalfie M. WormBook- An Online Review of C. elegans Biology. West Coast Worm Meeting. 2004:247–252.

[63] Amitani M, Asakawa A, Amitani H, Inui A. The role of leptin in the control of insulin-glucose axis. Frontiers in Neuroscience. 2013;(7 APR). Available from: https://doi.org/10.3389/fnins.2013.00051.

Appendix A

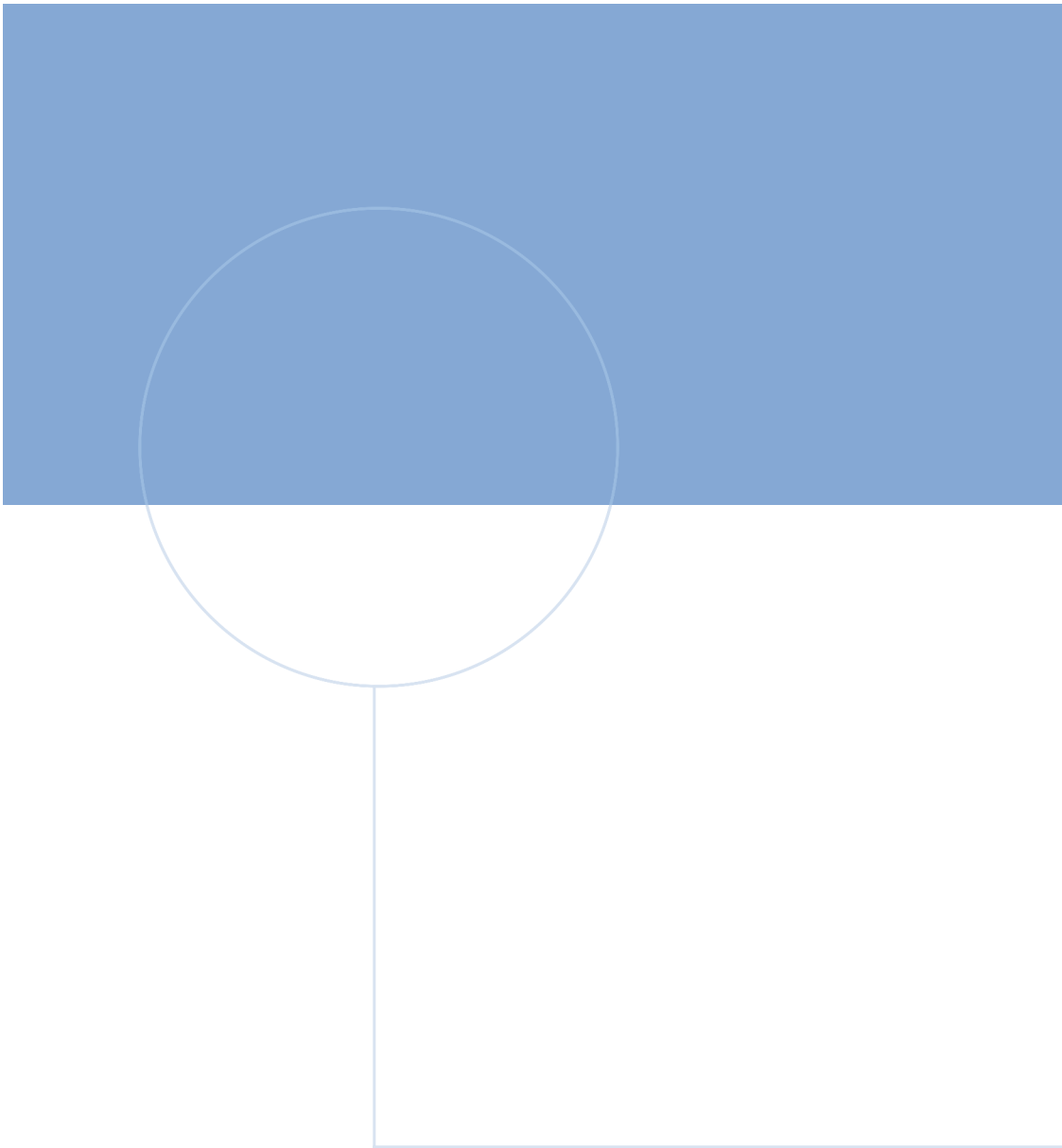
```
warnings.filterwarnings("ignore")
#Choose clf to be one of:
 ↪  #LogisticRegression(solver = "liblinear")
 ↪  #DecisionTreeClassifier()
 ↪  #KNeighborsClassifier(n_neighbors=(15))
 ↪  #RandomForestClassifier(max_depth=3)
 ↪  #MLPClassifier()
clf = RandomForestClassifier(max_depth=3)

#population can for example be 10 or 1000
population = 1000

mcc = make_scorer(matthews_corrcoef)
D = pd.read_csv("inputML.csv", delimiter=';')
D = D.transpose()
allfeats = D.columns
allfeats = list(allfeats)
allfeats.remove(0)
numcols = set(allfeats)
numcols = list(numcols)
D[allfeats].dtypes
```

```
D[allfeats]
X = D[allfeats]
Y = D[0].values
Y = pd.DataFrame(Y)
Y = np.array(Y)
Y_binary = []
for i in Y:
    if i =='Diabetes':
        Y_binary.append(1)
    else:
        Y_binary.append(0)
Y = np.array(Y_binary)
```

Listing 1: The data pre-processing after the imports in Chen's code was adjusted to the dataset [54].



NTNU

Norwegian University of
Science and Technology