CHALMERS
UNIVERSITY OF TECHNOLOGY

Causal Relationships Between Food Intake and Stomach Issue

An Algorithmic Detection Using Machine Learning

Master's thesis in
Biomedical Engineering

OSKAR KARNBLAD,
NILS NORDEMAN

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2017

Master's thesis EX030/2017

Causal Relationships Between Food Intake and Stomach Issues

An Algorithmic Detection Using Machine Learning

OSKAR KARNBLAD
NILS NORDEMAN

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Division of Signal Processing and Biomedical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2017

Causal Relationships Between Food Intake and Stomach Issues
An Algorithmic Detection Using Machine Learning
Oskar Karnblad, Nils Nordeman

Supervisor: Anders Elfving, Tummy Lab
Examiner: Tomas McKelvey, Department of Electrical Engineering

Master's Thesis EX030/2017
Department of Electrical Engineering
Division of Signal Processing and Biomedical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Several models fitted to data points. Designed by: Thijs Keesenberg

Typeset in LaTeX
Gothenburg, Sweden 2017

Causal relationships between food intake and stomach issues
An algorithmic detection using machine learning
Oskar Karnblad, Nils Nordeman
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Irritable Bowel Syndrome (IBS) is the most common functional disease related to the bowel and is classified as one of the most common diseases in the world with 11.2% of the global population suffering. An accurate tool as an aid will therefore have a major societal impact. In this thesis, algorithms for identifying causal relationships between food intake and stomach issues from synthetically generated data and patient's self-recorded journals were investigated as the primary aim. The thesis was confined to an investigation of algorithms appropriate for small datasets. Algorithms considered appropriate were members from the following families of algorithms: regression analysis, ensemble learning, support vector machines and Bayesian statistics. The results were obtained by running each algorithm on the same datasets and performing averaging. The study found that the beta-binomial hierarchical model acquired the highest average performance for all metrics considered when selecting symptom intolerances from synthetic data. However, due to the unknown symptom generating behavior of users, the limitations of the model may affect the performance significantly. We believe that utilizing the hierarchical model in combination with another algorithm may be useful for analysis of the available datasets.

Keywords: IBS, machine learning, beta-binomial hierarchical model, intolerance detection

Acknowledgements

Firstly, we would like to thank everyone at Tummy Lab for all the help and support throughout the thesis work and for accepting us as thesis workers. Thanks to our supervisor Anders Elfving for the many hours of discussion and for always having the door open to answer uncertainties. Thanks to Anders Carling and Thijs Keesenberg for the computer-related support and the help with the graphics respectively. Not to forget the many waves of laughter. Thanks to the dietitian and Ph.D. Lena Böhn for providing clarifications within the field of digestion and the gastrointestinal tract.

We would also like to express our gratitude to David Fendrich at Crawlica for all the supervision and assistance in the field of machine learning and statistical analytics and especially for introducing us the field of Bayesian statistics and hierarchical models. Moreover, we would like to show our appreciation to postdoc Francesco Gatto for sharing his knowledge and thoughts within the field of, especially regression and ensemble learning. Our examiner at Chalmers, professor Tomas McKelvey, deserves huge thanks for all the support and discussions and for always pointing us in the right direction. We always left motivated after our meetings.

Finally, we would like to show our deepest gratitude to our family and friends for their support throughout our time at Chalmers.

Oskar Karnblad, Nils Nordeman, Gothenburg, May 2017

Contents

List of Figures

List of Tables

1
Introduction

In Sweden, over one million people are suffering from chronic stomach issues diagnosed as Irritable Bowel Syndrome (IBS) with symptoms such as diarrhea, gas, pain and bloating. Individuals suffering from IBS often keep a paper-based journal of food intake and symptoms, and then manually look through the data to find patterns. A majority of the individuals say they want more help investigating what foods to avoid in order to ease the symptoms. Since the disease is classified as one of the most common ones in the world with 11.2% of the global population suffering, an accurate tool as an aid will have a major societal impact [10].

The possible benefits of an accurate tool for people with IBS are primary to increase suffering person's Quality-of-Life (QoL) but also to reduce direct and indirect health costs related to IBS for the society. In the year 2000, [37] conducted an SF-36 health survey, which estimates a person's QoL based on 36 measures and concluded that IBS patients had similar QoL as patients with end-stage renal disease and diabetes mellitus. Moreover, the QoL of IBS patients was lower than for people suffering from gastroesophageal reflux disease.

From a health economic point of view, IBS leads to a substantial amount of indirect and direct costs. [33] has shown that the yearly cost directly connected to IBS in the United States is approximately 1.3 billion US dollar. However, this figure does not include the indirect costs of e.g. productivity losses at the workplace and off-work days. Moreover, [14] has estimated the average off-work or school days for IBS patients to be 13.4 days/year which is three times higher than for a non-sufferer.

The health system can not only increase the cost-benefit with a well-performing aid for IBS patients but also reduce the pressure on dieticians and gastrointestinal experts. With the many sufferers in Sweden and approximately only 800 specialized dietitians on the subject, this limits the possibilities of receiving help and support for sufferers [37].

1.1 What is IBS?

IBS is the most common functional disease related to the bowel. The cause is still unknown but factors such as diet, stress, depression, gut dysbiosis and prior infections have been acknowledged as confounding elements [8]. Women are twice as often diagnosed with IBS compared to men and 80% of the individuals with IBS suffer due to dietary and nutritional factors [28][5]. Persons with the disease often experience pain due to visceral hypersensitivity, a condition where the sensory nerve

endings in the bowel have an unnaturally strong response to stretching of the bowel.

IBS can be divided into 4 subtypes, namely IBS-C, IBS-D, IBS-M and unsubtyped IBS (IBS-U). The subtypes are based on the form and consistency of the stool. IBS-C stands for IBS with constipation and indicators are often bloating, delayed or sporadic movement of the bowel and hard stool. IBS-D is IBS with diarrhea and comes with abnormally frequent bowel movement together with watery stool. Finally, mixed IBS is a type which alternates between IBS-C, IBS-D and IBS-U and does consequently not fulfill the specifics for neither IBS-C, IBS-D or IBS-M [5].

1.2 Related Work

IBS has been a subject of medical research for the past three decades and is partly understood. Substantial research of the effects of food intake and nutrition has been performed. Examples are [5], [8] and [36]. Other explored topics have been on susceptibility for IBS and genetically contributing factors as in e.g. [15]. The societal and health economic impact of IBS has also been charted out as mentioned earlier by, for example, [35], [14] and [33].

A study on the topic of food intake that has been conducted by performing dietary modifications. The study showed that avoiding certain short-chains of carbohydrates (FODMAP), which are absorbed poorly by the small intestine, eased the symptoms for approximately half of the persons undergoing the trial [5]. However, the outcome from certain food intakes was shown to be highly individual. FODMAPs are further introduced in section 2.1.

To our knowledge, no similar research has been carried out using a dataset as large as the one available for this thesis. In addition, the focus of the previous studies of IBS has been from a medical and anatomical point of view whilst this report investigates the disease using machine learning and statistical analytics.

The field of incorporating machine learning techniques in the field of medicine and especially medical diagnosis is however not unexplored. Common classifiers introduced in this thesis e.g. naive Bayes classifier, support vector machines and decision trees have been fed with patient records and medical images for labeling purposes in the healthcare sector since the 1990s [25].

1.3 Aim

The aim of this master thesis is to identify available techniques to algorithmically identify causal relationships between food intake and stomach issues from patient's self-recorded journals, utilizing machine learning and statistical analytics. Other actions considering data usage which can result in increased accuracy for the end-user might also be taken. Finally, the aim is to identify which of the investigated algorithms that is the most optimal for the datasets available.

1.4 Research Questions

In order to reach the aim of this thesis work the following research questions will be considered. The questions are listed in prioritized order, that is, the most emphasis will be placed on answering question 1.

1. What algorithms are applicable to the problem of finding causal relationships between food intake and stomach issues for the data available? This question is divided into sub-questions for:
   - Classifiers
   - Regressors
   - Statistical Inference
2. What would be the most suitable algorithm to use for extracting intolerance ingredients for individual users?
3. Can the data be modified in a more beneficial way?

1.5 Scope

Due to the relatively few data points available for analysis, deep machine learning methods, for instance, artificial neural networks or similar algorithms which require large sets of data was not considered. Furthermore, the emphasis of this thesis work did not lie in developing algorithms but rather apply already existing algorithms from Python machine learning packages. Moreover, this thesis work did not put any effort in collecting more data but instead use the available data in a more advantageous way. Parameters used to generate synthetic data was, if possible, approximated to its furthest extent in order to emulate individual's behavior. However, if impossible, these were subject to guesses made by the authors of this thesis.

1.6 Thesis Outline

The beginning of the next chapter will introduce the connection between FODMAP and IBS more thoroughly. Thereafter, two approaches for finding relationships are represented, namely the machine learning and the statistical approach. In addition, methods for evaluating the performance of the procedures are presented and constitutes together with the three previous topics the theory chapter of this thesis. Thereafter, the method used for testing and pre-processing data is presented. The result conducted from the method is presented and also analyzed in the subsequent chapters.

2
Theory

This chapter presents the theoretical framework which the method and result chapters are based upon. The theory chapter begins with a brief introduction to the collection of short-chain carbohydrates and sugar alcohols called FODMAP and their connection to the disease IBS. Thereafter, a description of the machine learning and statistical algorithms for identifying relationships is presented. Finally, designated methods for performance evaluation for the algorithms are described.

2.1 FODMAP and IBS

Fermentable oligo-, di-, mono-saccharides and polyols (FODMAP) are a set of short-chain carbohydrates and sugar alcohols. Fructose, fructans, galacto-oligosaccharides, lactose, sorbitol and mannitol are examples of the most common FODMAPs. Foods including high level of FODMAPs are typically apples, pears, cauliflowers, leek, onion, milk and honey [40].

During digestion, carbohydrates are decomposed into monosaccharides by the use of enzymes which are secreted from e.g. the stomach, salivary glands and from cells in the small intestine. Glucose is a common monosaccharides which is easily absorbed by the small intestine. However, depending on the coincident absorption of glucose, the ability for the small intestine to absorb fructose is limited. In addition to fructose, the sugar alcohol sorbitol may also be malabsorbed by the small intestine.

The malabsorbed carbohydrates, including but not limited to fructose, act as solutes and draws fluid into the interior of the gastrointestinal walls. The excessive water stretches the lumen of the gastrointestinal tract which causes pain due to visceral hyperactivity for individuals with IBS. The fluid may also cause the muscles around the gastrointestinal tract to spasm which may cause diarrhea. Moreover, the non-digested carbohydrates are metabolized by the bacterial strain in the intestine which causes the production of gas. The gas amplifies the symptoms of pain, muscle spasms and diarrhea [5].

2.2 Machine Learning for Relationship Identification

The field of data analysis and machine learning is an important part of modern science. The aim is to discover data patterns which might be invisible for the human

eye. The increase in generated data and widespread availability of processing power enables machine learning to be implemented within numerous fields. The objective of predictive machine learning is to, based on training data, predict the output from new introduced data points.

A machine learning problem with the aim to make decisions or predictions based on data can be divided into two essential categories, regressors and classifiers. Regressors are commonly used for quantitative problems while the classifiers are used in qualitative problems. However, the regression techniques can often be converted to a classifier by feeding the continuous output to an activation function and perform thresholding in order to obtain discrete values. There exists several groups of activation functions such as identity, binary step and the logistic function. The logistic function, shown in 2.1 below, is most commonly used in machine learning and is also known as "softstep" due to its S-shaped characteristics [42].

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.1}$$

2.2.1 Imbalanced Datasets

Most datasets that are non-synthetical and unprocessed are often unbalanced. This is a well-known problem in the field of machine learning and has been a highly researched topic for the last 20 years. Imbalanced data occasionally yields biased results and returns misleading predictive power for the machine learning algorithms. An example of this issue is within detection of fraudulent transactions where, for example, 1 out of 100 transactions are fraudulent and the rest are legitimate. A classifier always predicting non-fraudulent will yield an accuracy of 99% which becomes misleadingly good prediction power [41]. Suitable additional classifier performance metrics for imbalanced data are presented in Section 2.4.1. An imbalance ratio 1:4 between rarest and most common class is considered to be an imbalanced dataset [22].

2.2.2 Regression Analysis

Linear regression techniques are probably the most fundamental family of machine learning algorithms. The essential idea is most easily introduced by linear regression. Linear regression is the process of estimating values for coefficients for a model based on observed data points. The model is estimated through fitting the coefficients, β, for the model by using a minimization method, often Residual Sum of Squares (RSS). A visualization of linear regression can be seen in Figure 2.1 below. Assume an input-matrix $\mathbf{X} = (X_1, ..., X_N)$ and a corresponding prediction of the output Y_j where $i = 1, 2..., N$. Then the regression model takes the following form:

$$f(X) = \beta_0 + \sum_{j=1}^{N} X_j \beta_j \tag{2.2}$$

In the situation presented in Figure 2.1, N in the above equation is set to 1 and the formula for describing a line in $\mathcal{R}^2$ space is obtained.

Least Absolute Shrinkage and Selection Operator (LASSO) is a machine learning method performing penalized, linear regression through regularization. The regularization performed by LASSO aims to minimize $\sum_i |\beta_i|$ and is also known as L1-regularization. Regularization minimizes the complexity of the underlying linear regression model and therefore reduces the risk of overfitting [39][23, pp. 43-51].

The estimate from LASSO, $\hat{\alpha}$ and $\hat{\beta}$ where $\hat{\beta} = (\hat{\beta_1}, .., \hat{\beta_N})$, is calculated as shown in 2.3 below. The observed input data is $\mathbf{x}^i = (x_{i1}, ..., x_{iM})^T$ and the observed output data is y_i where $i = 1, 2..., N$.

$$(\hat{\alpha}, \hat{\beta}) = \arg\, min \left(\sum_{i=1}^{N} (y_i - \alpha - \sum_j \beta_j x_{ij})^2 \right) \text{subject to} \sum_j |\beta_j| \leq t \tag{2.3}$$

$t \geqslant 0$ is the shrinkage parameter regulating the amount of shrinking applied to the model [39].

Figure 2.1: The principle of linear regression visualized

The L1-regularization is however not the only regularization term used in the field of regression. L2-regularization, compared to L1-regularization, aims to minimize $\sum_i |\beta_i|^2$. The different constraints are presented graphically in Figure 2.2. $||\omega_1||$ and $||\omega_2||$ are the weights for L1- and L2-regularization respectively. If one considers the two-dimensional case, the constraint for LASSO takes the form $|\beta_1| + |\beta_2| < t$ and is represented by the diamond-shape in the Figure 2.2 a. The shape of the constraint is the cause of the feature selection performed by the LASSO algorithm. The level curve of the objective function is more likely to hit a corner of the constraint region, resulting in that certain components in the β-vector is set to zero.

Ridge regression is an algorithm that applies L2-regularization. The two-dimensional constraint of the ridge regression is presented in Figure 2.2 b. The constraint takes

the form $\sqrt{\beta_1^2 + \beta_2^2} < t$ and therefore has the shape of a circle. In contrast to the LASSO algorithm, the ridge regression constraint does not have sharp edges. Therefore, a lower likelihood that the level curves of the objective function hit a corner is obtained, resulting in a higher number of nonzero elements in the β vector compared to the LASSO algorithm [34]. If both L1- and L2-regularization is utilized, the regression is called elastic net regularization.

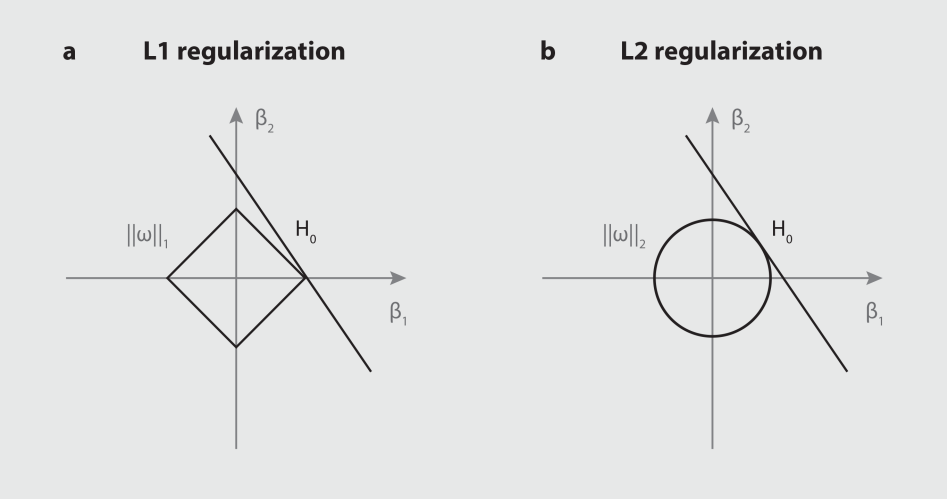


Figure 2.2: Constraint areas for the different regularization constraints

The counterparts of linear regression and LASSO in the classification case are called logistic regression and logistic LASSO. The elementary idea of these classifiers is the same as presented in the begin of this section. The continuous output values from linear regression and LASSO are fed through an activation function thus discretizing the output data. The activation function is in these cases is the logistic function. Logistic regression is widely used in many application areas [24].

2.2.3 Ensemble Learning

The general approach of ensemble techniques is to create numerous sets of replicates of the original data. The replicates are retrieved by sampling from the original data set with replacement. The replicates are used for training models to each set. After the training phase, each model prediction is aggregated to make a majority prediction. This general procedure is also known as **b**ootstrap **agg**regat**ing** (bagging) and has been the subject of extensive research, and implemented in various fields such as medical diagnostics the last decades of the 20th century [23, pp. 282-288].

A fundamental part of ensemble learning is decision trees. Decision trees can be divided into classification trees and regression trees (CART) depending on the format of the input to the trees. A decision tree takes a numeric value X as input to the trees so called root node. From the root node, the data is split into subnodes. The splitting continues until reaching the end of the tree, also referred to as the leaves. How the split is carried out is depending on which type of decision tree algorithm that is being used. However, how much information that is necessary to describe

the data and how much variance each split introduces are two common aspects considered [16].

One of the most widely used decision tree techniques is random forest. The essential idea of random forest for classification purposes is very similar to the approach of bagging. A committee of de-correlated decision trees cast individual votes for a prediction class. The votes for each tree are generated by splitting each node and follow the split that maximizes the information retrieval until reaching the leaves of the tree. This split is often called the optimal split. Decision trees are known to introduce a great amount of variance in their output votes. To reduce the variance introduced by decision trees, the result from the votes is averaged and used as the final output [23, pp. 587-604].

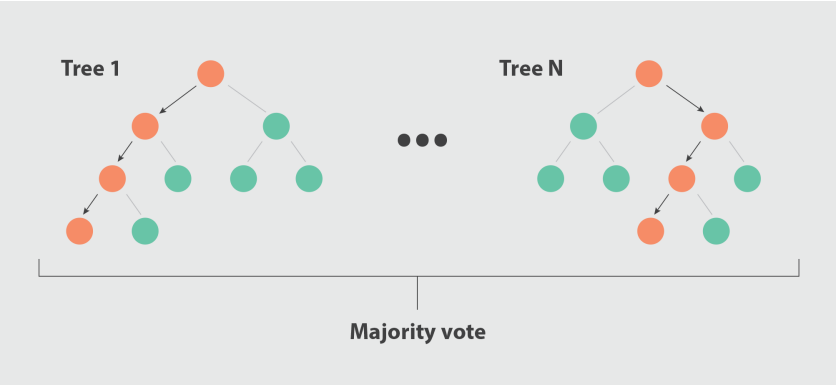


Figure 2.3: The simplified idea of random forest

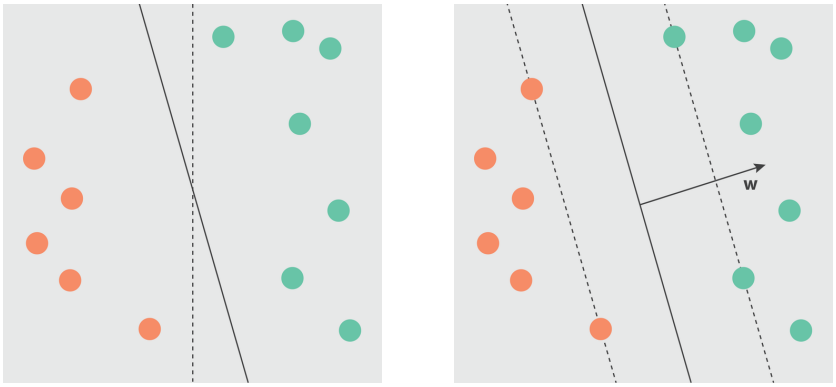
Random forest, just as the bagging technique where it originates from, has a low-bias since each individual tree has equivalent expected value as an average of a set B trees. This is, in addition to the reduced variance and the simplicity of the method, the main reasons why random forest is a popular classification alternative. Moreover, tree algorithms perform well on imbalanced datasets since both classes in data often are addressed in the creation of trees and make random forest a strong classification candidate when imbalance between classes is present. Since the variable selection in each tree is random, random forest does not introduce overfitting [23, pp. 587-604].

Boosting is yet another branch derived from the principle of bagging. [23, pp. 337-387] states that this learning method is one of the most powerful created in the past two decades. Boosting, in contrast to random forest, uses a collection of weak learners that each can cast a weighted vote in contrast to the equal-influence votes featured in random forest algorithm. The weighting of the votes is obtained by running the training of the algorithm multiple times and redistributing weights between learners depending on the training error retrieved from each run. The most well-known boosting technique is ADABOOST [23, pp. 337-387].

2.2.4 Support Vector Machines

Support Vector Machines (SVM) is a supervised machine learning method with a broad span of applications, ranging from face recognition to classification of biologi-

cal data [44][9]. SVM is efficient on high dimensional data and is therefore commonly used in high feature problems such as language processing and classification of gene expression data [44]. There exists a variety of different types of SVM. However, the description in this section is limited to the most basic SVM which utilizes a linear hard margin.



(a) (b)

Figure 2.4: Possible linear boundaries for separating classes using SVM

Figure 2.4 presents a binary classification problem in which the SVM can be applied in order find a linear boundary which separates the classes. As can be seen in the left visualization in Figure 2.4, several possible boundaries separate the classes. However, the SVM aims to fit a hyperplane, which maximizes the separation between the classes. The hyperplane can be expressed as $\boldsymbol{w} \cdot \boldsymbol{x}_i + b = 0$ where $\boldsymbol{x_i}$ is the set of input vectors located within the hyperplane, each having a label which takes the binary values -1 or +1 depending on class. The weights, $\boldsymbol{w}$, are the normal to the hyperplane and determine the orientation while b is the offset to the origin in input space [23, pp. 418-420]. The hyperplane which maximizes the separation is presented in the right visualization in Figure 2.4 and can be expressed by solving the following optimization problem:

$$min\frac{1}{2}||\boldsymbol{w}||^2 \text{ subjected to constrains } y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \geq 1, \forall i \tag{2.4}$$

The constraint ensures correct classification while the minimization maximizes the margin, i.e the separation of the classes. However, in many situations, complete separation of classes is not possible and therefore the equation need to be modified in order to allow some misclassification. This is achieved by introducing a variable to the constraint called the slack variable which measures the relative distance overlap, with respect to the margin. After introducing the slack variable, optimization problem above can be expressed as:

$$min\frac{1}{2}||\boldsymbol{w}||^2 + C\sum\zeta_i \text{ under the constrains } \begin{cases} y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \geq 1 - \zeta_i \\ \zeta_i \geq 0 \end{cases} \tag{2.5}$$

Which is the usual way the SVM is defined for non-separable classes [23, pp. 418-420]. The constant C is a tunable parameter which regulates the trade-off between

the training classification error and margins maximization [27]. The previous description of the SVM performs well on balanced datasets. However, as described in section 2.2.1, one rarely has this type of data. Therefore, in order to improve the performance on imbalanced datasets, equation 2.5 can be modified in the following manner to penalize misclassification of classes differently as described by equation 2.6:

$$min\frac{1}{2}||\boldsymbol{w}||^2 + C_+ \sum_{i:y_i=+1} \zeta_i + C_- \sum_{i:y_i=-1} \zeta_i \text{ with } \begin{cases} y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \geq 1 - \zeta_i \\ \zeta_i \geq 0 \end{cases} \tag{2.6}$$

C+ and C- are the weights which regulate the misclassification penalty for the different classes. If the weights C is chosen as C-= C+ one ends up with the ordinary SVM expressed by equation 2.5. In order to penalize misclassification of minority classes more heavily, the values of C- and C+ can be set to be inversely proportional to the class frequency thus allowing better performance on imbalanced data [27].

When using the SVM in practice, a kernel function, also referred to as the similarity function, is often applied. The aim of a kernel function is to transform the data x into a higher dimensional space in order to gain linear separation between classes. According to [13], a kernel suited for high feature problem with few data points is the linear-kernel. A linear-kernel utilizes the the expression in equation 2.5 and is therefore equivalent to not applying any kernel at all. Other kernels are also used in combination with SVM but will not be presented in this report.

2.3 Bayesian Statistics for Relationship Identification

Bayesian inference allows incorporation of prior information for parameters which may, in some situations, be advantageous, especially when limited amount of data is available. In the initial phase of Bayesian inference, one develops a model for the joint probability $p(\theta, y)$, where θ is unobservable vector quantities or population parameters of interest and y is the observed data, in order to express the conditional probability $p(\theta|y)$ in an appropriate manner [20]. This is achieved by applying Bayes' rule, which states that the posterior probability distribution $p(\theta|y)$ of a parameter θ, given the data, is equal to the likelihood $p(y|\theta)$ multiplied with the prior belief $p(\theta)$ of the parameter divided by the probability of the data [4]. Bayes' rule is presented in equation 2.7 below.

$$p(\theta|y) = \frac{p(\theta, y)}{p(y)} = \frac{p(\theta)p(y|\theta)}{\int p(\theta)p(y|\theta)d\theta} = \frac{p(\theta)p(y|\theta)}{p(y)} \tag{2.7}$$

For discrete value of θ, $p(y)$ becomes $\sum_\theta p(\theta)p(y|\theta)$, where all possible values of θ are included in the summation. The integral in the denominator is occasionally intractable. However, since $p(y)$ does not depend on θ it can be seen as a constant, yielding the unnormalized posterior distribution:

$$p(\theta|y) \propto p(\theta)p(y|\theta) \tag{2.8}$$

The posterior distribution represents the uncertainty in the unknown parameter values after data has been observed while the prior distribution represent the uncertainty in the unknown parameter values before the data is observed. The likelihood is a function of the model parameters, given the observed data [18].

2.3.1 Naive Bayes Classifier

The naive Bayes classifier is the least complicated approach to Bayesian inference. The naive Bayes classifier applies Bayes' theorem which for one variable is unproblematic. However, as the number of variables increases, the math becomes more complicated. Assume that the features held by the input-vector $\mathbf{x} = x_1, x_2, \ldots, x_N$ is to be classified to the classes G_j where j is the possible outcomes. The conditional probability of the problem can then be formulated with Bayes' theorem as:

$$p(G_j|\mathbf{x}) = \frac{p(G_j)p(\mathbf{x}|G_j)}{p(x)} \tag{2.9}$$

The denominator of the equation becomes constant due to the lack of dependence to G_j. The numerator of the expression is the same as $p(G_j, x_1, \ldots, x_N)$ which is obtained by the applying the chain-rule to the definition of the conditional probability repeatedly. The result from this is non-trivial and difficult to evaluate. Therefore, the naive Bayes approach to the problem is to see every variable as independent to the other variables and is also the reason it is called "naive". Assuming independence and replacing the denominator of 2.9 with a scaling factor, K, the final equation can be rewritten as:

$$p(G_j|\mathbf{x}) = \frac{1}{K}p(G_j)\prod_{i=1}^{N} p(x_i|G_j) \tag{2.10}$$

The class, G_j, that yields the highest probability are the class which the variables are classified as. However, the approach of assuming each feature independent of the other does not apply well to reality since variables rarely are completely independent of each other [31].

2.3.2 Beta-Binomial Model

The beta-binomial model utilizes, as the name suggests, the beta and the binomial distribution in order to infer a posterior distribution on the binomial parameter. The binomial distribution is often used for describing dichotomous data which has been generated through a sequence of Bernoulli trials with aim to estimate unknown population proportions. Due to exchangeability, the data can be expressed in terms of number of successes y, over n trials. By allowing the parameter θ to describe the proportion of success in a population, or probability of success in each trial, the formulation of exchangeability can be converted to one using independent and identically distributed random variables [18]. The binomial distribution, used as the likelihood function in the beta-binomial model, can thus be written as described by the equation below.

$$p(y|\theta) = Bin(y|n, \theta) = \begin{pmatrix} n \\ y \end{pmatrix} \theta^y (1-\theta)^{n-y} \tag{2.11}$$

In order to make Bayesian inferences, a priori distribution has to be specified. If one chooses a priori distribution with the same shape as the likelihood, a posteriori with the same form will be obtained. This property, where the posterior has the same parametric form as the prior distribution is called conjugacy. Conjugacy is mathematically desirable since the posterior will follow a known parametric form [18]. It is achieved by choosing the beta distribution as the prior distribution, which has the following density:

$$p(\theta) \propto \theta^{\alpha-1}(1-\theta)^{\beta-1} \; \theta \sim Beta(\alpha, \beta) \tag{2.12}$$

Where α and β parameterize the beta distribution and is often referred to as hyperparameters. The value of α and β model the prior belief regarding the parameter θ. If no prior knowledge is available, the values of α and β is chosen in a manner such that a non-informative prior distribution is achieved. Examples of such values could be α, $\beta = 1$ or α, $\beta = 0.5$ [18][21]. The posterior distribution of θ, including the hyperparameters α and β, can be expressed as in equation 2.13 below. By inspecting the equation, one realizes that the parameter values of α and β, will have a larger impact in situations when the number of data points, n, are small.

$$P(\theta|y) \propto \theta^{y}(1-\theta)^{n-y}\theta^{\alpha-1}(1-\theta)^{\beta-1} = \theta^{y+\alpha-1}(1-\theta)^{n-y+\beta-1} = Beta(\theta|\alpha+y, \beta+n-y) \tag{2.13}$$

The probability of success in future draws can be expressed as the posterior mean value of θ. The mean value will lie between the sample mean and and prior mean and can be expressed as by 2.14.

$$E(\theta|y) = \frac{\alpha+y}{\alpha+\beta+n} \tag{2.14}$$

2.3.3 Hierarchical Models

A hierarchical model can be viewed as a mathematical description of dependencies between variables where the credibility values of some variables are dependent on other variable values. There are several situations which can be modeled using a hierarchical structure. In the following subsection, the concept of hierarchical modelling is presented with support from an example. Moreover, the example will be referred to throughout the chapter.

Consider a factory that produces a coin used for coin flipping, where the sequence of successful flipping outcomes y_j follows an independent binomial distribution, parameterized as $y_j \sim Bin(n_j, \theta_j)$. Each coin distribution is parameterized individually by θ_j which represent the bias of the coin and n_j which represent the number of performed flips. Since the coins are produced in a similar manner, one can assume that the there exists a common factory distribution, describing the general coin bias, from where the values of θ_j, are independently sampled from. Since the observed values y_j follows the binomial distribution, the beta distribution will provide conjugacy as described in previous section. The hierarchical structure can graphically be interpreted as presented in Figure 2.5 [18].

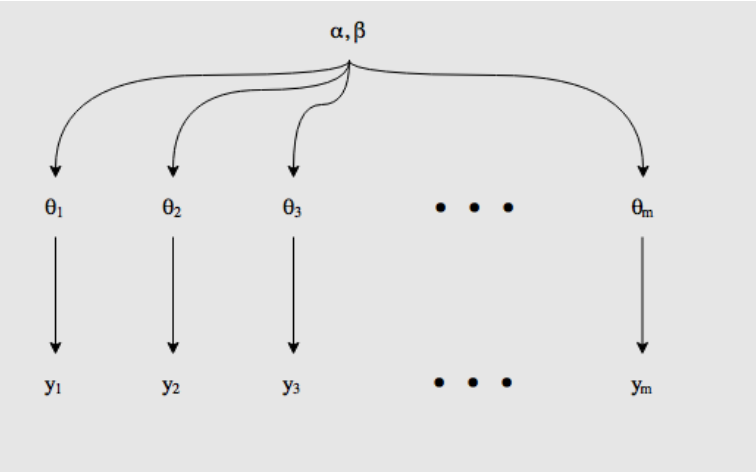


Figure 2.5: An example of a hierarchical model's parameters structure

Moreover, the hierarchical structure can further be described with the use of Bayes' rule as by equation 2.7.

$$p(\theta_j, \alpha, \beta|y_j) \propto p(\alpha, \beta)p(\theta_j|\alpha, \beta)p(y_j|\theta_j, \alpha, \beta) = p(\alpha, \beta)p(\theta_j|\alpha, \beta)p(y_j|\theta_j) \quad (2.15)$$

The right-hand side of the equal-sign in the equation above states that the observed data y_j is only dependent on the variable value of θ_j, which is in turn only dependent on the variable value of α and β. The observed data is considered as independent of all parameters except θ while the θ distribution is conditionally independent of all parameters except α and β.

The dependencies in the hierarchical structure enables more informed estimates of parameters and is one of the advantages with hierarchical models. The relationships between the higher level parameters α, β and lower level parameter θ are called shrinkage. In addition to the advantages described, the shrinkage reduces the impact of random sampling noise in the lower level parameter space since the θ_j is shrunk toward the mode of the prior distribution [26].

2.3.3.1 Empirical Bayes for Estimating Beta Parameters

The empirical Bayes method utilizes point estimates for the parameters which parameterize the population distribution rather than joint distributions and is therefore not strictly Bayesian. In the empirical Bayes method, one considers the data as random samples from a common distribution. In the beta-binomial model, exemplified in the previous section by the coin factory experiment, one would estimate the beta parameters α, β by calculating the sample mean and sample variance from the binomial parameters $\theta_{1:n}$, estimated by flipping the individual coins. By knowing the sample mean and variance, one can utilize the method of moment in order to estimate the parameters alpha and beta, as described by the equations below.

$$\alpha + \beta = \frac{\mathbf{E}(\theta)(\mathbf{1} - \mathbf{E}(\theta))}{var(\theta)} - 1 \quad (2.16)$$

$$\alpha = (\alpha + \beta)\mathbf{E}(\theta)\alpha = (\alpha + \beta)(1 - \mathbf{E}(\theta)) \quad (2.17)$$

$$var(\theta) < \mathbf{E}(\theta)(1 - \mathbf{E}(\theta)) \tag{2.18}$$

$E(\theta)$ is the sample mean of θ and $var(\theta)$ is the sample variance [18][30]. Estimating the beta parameters in this manner is a more rapid procedure compared to fully Bayesian methods, which is often require computationally heavy procedures like the Markov Chain Monte-Carlo sampler [18].

To account for differences in the collected data, the variance can be inflated. For the beta-binomial model, this would correspond to decreasing $\alpha + \beta$ while keeping the ratio $\frac{\alpha}{\beta}$ constant. To associate this with the introduced example for this section, such a situation could be if there were two factories producing coins in likewise manner or if there exists an uncertainty regarding a change in the manufacture process [18].

2.3.3.2 Bayes Factor for Model Comparison

The above model describes a scenario in which the θ_i distributions is considered to follow one common distribution. However, in some situations, there can be more than one model competing to describe the data. Consider once again the coin factory, but now, the factory produces two types of coins, one that is biased toward heads and one that is biased toward tails. Given a coin, one has to have the ability to decide from which of these two distribution the coin originates. The decision can be made by the use of Bayes factor (BF) [26].

The BF measure is gained by comparing the posterior probability of models. By applying the Bayes' rule to a specific model one gets following expression:

$$p(m|y_j) = \frac{p(y_j|m)p(m)}{\sum_m p(y_j|m)p(m)} \tag{2.19}$$

Where $p(y_j|m)$ is the probability of the data given the model, marginalized across all parameter values. By comparing the posterior distribution of two models one gets following expression:

$$\frac{p(m=1|y_j)}{p(m=2|y_j)} = \underbrace{\frac{p(y_j|m=1)}{p(y_j|m=2)}}_{\text{BF}} \frac{p(m=1)}{p(m=2)} \underbrace{\frac{/\sum_m p(y_j|m)p(m)}{/\sum_m p(y_j|m)p(m)}}_{=1} \tag{2.20}$$

BF describes the ratio of the probability of the data in the different models. According to [43], a BF of 3 indicate substantial evidence for model 1 while a BF of 1/3 indicate substantial evidence for model 2. With the use of these values, one can make a decision regarding the choice of model [26].

2.3.3.3 Change-Point Modelling

The change-point model connected to the beta-binomial model in this thesis is not considered as a Bayesian method. However, due to its relation, it is featured in this section. Change-point models are most commonly used to find abrupt changes in time series and are applied in, for example, medical condition monitoring, speech recognition and human activity analysis [3]. The abrupt changes can, for example, be

in mean value or in variance. The changes can be modeled in various ways, however, in this section, a description of the change in a binomial probability parameter θ will be presented [3][12]. The estimation of the change-point, k, will be described using a maximum likelihood estimate (MLE) method.

Consider a number of variables y_i, each following a binomial distribution, $y_i = Bin(n_i, \theta_i),\ i\exists[1, c]$, where y_i is the number of ones or successes in n trials for variable i. The null hypothesis H_0 can be written as in equation 2.21 and describe a situation where no variation between the θs exists:

$$H_0 : \theta_1 = \theta_2 = ... = \theta_c = \theta_0 \tag{2.21}$$

If one wants to examine if there exist 2 distributions, separated at index k, within the variables y, one can write the hypothesis as:

$$H_1 : \theta_1 = \theta_2 = ... = \theta_k = \theta_k^- \neq \theta_{k+1} = ... = \theta_c = \theta_k^+ \tag{2.22}$$

As before, if one assumes independence between the variables y, one can write the likelihood functions for the hypotheses L_0 and L_1 as:

$$L_o(\theta_0) = \prod_{i=1}^{c} \binom{n_i}{y_i} {\theta_0}^{y_i} (1-\theta_0)^{(n_i - y_i)} \tag{2.23}$$

$$L_1(\theta_{k^-}, \theta_{k^+}) = \prod_{i=1}^{k} \binom{n_i}{y_i} {\theta_{k^-}}^{y_i} (1-\theta_{k^-})^{(n_i - y_i)} \prod_{j=k+1}^{c} \binom{n_j}{y_j} {\theta_{k^+}}^{y_j} (1-\theta_{k^+})^{(n_j - y_j)} \tag{2.24}$$

By taking the logarithm and deriving the likelihood above with respect to θ_0, θ_{k-} and θ_{k+}, following MLE of the parameter θ is achieved:

$$\theta = \frac{\sum_{i=1}^{c} y_i}{\sum_{i=1}^{c} n_i} = \frac{M}{N}, \theta_{k^-} = \frac{\sum_{i=1}^{k} y_i}{\sum_{i=1}^{k} n_i} = \frac{M_k}{N_k}, \theta_{k^+} = \frac{\sum_{j=k+1}^{c} y_j}{\sum_{j=k+1}^{c} n_j} = \frac{M_k'}{N_k'} \tag{2.25}$$

The logarithmic maximum likelihood ratio between the two hypotheses can be calculated as:

$$\log \frac{L_0(\theta_0)}{L_1(\theta_1, \theta_2)} = \log(L_0(\theta_0)) - \log(L_1(\theta_1, \theta_2)) \tag{2.26}$$

Which can be rewritten as the -2log maximum ratio which is described in the equation below:

$$L_k = -2 \log \frac{L_0(\theta_0)}{L_1(\theta_{k^-}, \theta_{k^+})} = 2[l(N, M) - l(N_k, M_k) - l(N_k^{'}, M_k^{'})] \tag{2.27}$$

Where $l(N, M)$ is defined as $l(n, m) = m\log(m) + (n-m)\log(n-m) - n\log(n)$. The $-2L$ log likelihood ratio has a chi-square asymptotic distribution and therefore the position of the change point k can be estimated such that $L = L_k^- = max_{l \leq k \leq l-1} L_k$ [12].

2.4 Performance Evaluation

There is an abundance of methods for evaluating the performance of machine learning techniques. Given the data structure and how the evaluation of the predictor is carried out, certain choices are more relevant than others. In this section, commonly used metrics for evaluation of both classifiers and regressors are presented.

2.4.1 Classifier Performance

In this chapter, properties and gained information from common classifier evaluation metrics are presented.

2.4.1.1 Confusion Matrix and Accuracy

A perspicuous and common way of visualizing a classifier's performance is with a confusion matrix (also sometimes referred to as contingency matrix). The confusion matrix is constructed by calculating the number of cases that the classifier has predicted a correct and incorrectly label respectively for the possible outcomes of the classifier. If the classifier is binary (meaning that class outcomes from the classifier are either 0 or 1), the confusion matrix consists of 4 bins, namely True Positives (TP), False Negatives (FN), True Negatives (TN) and False Negatives (FN) [7]. The TP and TN are defined as the correctly classified labels by the algorithm for the positive and negative class respectively and constitutes the diagonal of the confusion matrix as can be seen in table 2.1. The anti-diagonal of the table is constituted by the wrongly classified negatives and positives, FN and TN respectively.

Table 2.1: Confusion or contingency matrix

		Predicted Class	
		P	N
Actual Class	P	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

By use of the confusion matrix, accuracy of an algorithm can be calculated. The accuracy is the most general performance metric for machine learning applications and measures the fraction of correctly classified labels as shown in 2.28. However, accuracy may be a misleading metric to use for e.g. imbalanced datasets in which high accuracy will be obtained even for algorithms only labelling or predicting data to the majority class.

$$Accuracy = \frac{TN + TP}{TN + FP + FN + TP} \tag{2.28}$$

2.4.1.2 Precision, Recall and F1-score

It is often insufficient to use accuracy as the only performance metric and therefore, more informative metrics are often used. Precision and recall are two measures used for evaluating the classifier's performance for different classes. Recall measures the

relation between TP and FN and can interpret as the number of positive classes that were detected (recalled) as such by an algorithm. Precision measures the relation between TP and FP and can be interpreted as the percentage of correctly classified positives by an algorithm. The arithmetic definitions based on the terms introduced from the confusion matrix are shown below [11].

$$Precision = \frac{TP}{TP + FP} \tag{2.29}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.30}$$

It is desired to achieve a high precision and a high recall for a classifier coincidently. A metric which considers the trade-off between recall and precision is the F1-score. The F1-score is the harmonic mean of precision and recall meaning that a classifier yielding high precision and low recall or vice versa is penalized compared to a classifier that yields an intermediary result for both [7].

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{2.31}$$

2.4.1.3 ROC Analysis and Cohen's Kappa Coefficient

Receiver Operating Characteristic (ROC) analysis is a useful graphing method when handling data with unbalanced classes. The ROC curve is constructed by plotting the true positive rate (recall) against corresponding false positive rate over a sequence of cut-off points, with equal scales in the range [0, 1]. The false positive rate is the percentage of incorrectly classified negatives (often called false alarm rate). Assuming that the X-axis holds the false positive rate and the Y-axis holds the true positive rate, an ideal classifier results in an Area Under Curve (AUC) equal to 1. A diagonal ROC curve indicates an absence of predictive power for a classifier which is equivalent to randomly guessing a class [17].

In addition to the ROC-curve, the Cohen's Kappa Coefficient metric can be utilized in order to evaluate the inter-rater agreement of an algorithm. The coefficient is calculated by measuring to which extent two independent raters (interrater reliability) assign scores to the same variable. The inter-rater reliability can be measured by several means. However, the Cohen's Kappa has the advantage that interweaves the probability of common agreement by chance.

$$\kappa = \frac{p_o - p_{ca}}{1 - p_{ca}} \tag{2.32}$$

Where p_o and p_ca is the observed probability for common agreement and the probability for chance agreement respectively. The Kappa coefficient ranges between -1 to +1 where -1 is inverse classification, 0 is random classification and +1 perfect classification [29].

2.4.2 Regressor Performance

Since the output for a regressor takes continuous-values, the confusion matrix cannot be created and consequently, the metrics depending on the matrix are not available for evaluation. Therefore, specific measures for evaluating regressors needs to be applied.

2.4.2.1 Mean Squared Error and Median Absolute Error

The Mean Squared Error (MSE) evaluates the quality for a predictor performing regression. The quality is measured in the average magnitude of the error which is calculated as:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{Y}_i - Y_i)^2 \tag{2.33}$$

Where N is the total number of predictions, $\hat{Y}_i$ is the predicted value and Y_i the true value. A related metrics is the Median Absolute Error (MedAE) which is more robust to outliers. MedAE metric evaluates the performance of the regressors as the median deviation for all predictions between predicted and true value as shown in equation 2.34 [2].

$$MedAE = \text{median}(|\hat{Y}_1 - Y_1|, ..., |\hat{Y}_N - Y_N|) \tag{2.34}$$

2.4.2.2 Coefficient of Determination

The coefficient of determination (R^2) is a regression metric that answers the question *"How many % of the total variation in y is described by the variation in x?"* and is a common goodness-of-fit measure. This is evaluated as:

$$R^2 = 1 - \frac{SSE}{SS_{tot}} \tag{2.35}$$

Where $SSE = \sum_i (y_i - \hat{y}_i)^2$ is the Sum of Squared Errors and $SS_{tot} = \sum_i (y_i - \bar{y})^2$ is the total Sum of Squares which is the variance if normalized. $\hat{y}_i$ is the estimated value by the regression line at sample i and $\bar{y}$ is the mean of the true values for y. A R^2 score equal to 0 mean equal goodness-of-fit as the simple average while +1 is a perfect fit. The R^2 score can also be negative if the regression line is fitted worse than the simple average of the data [2].

2.4.3 Cross-Validation

Cross-validation is a commonly used method for estimating prediction error. The elementary purpose of cross-validation is to evaluate how well the result of an algorithm generalizes to an independent set of data. Moreover, cross-validation is commonly used for reduction of overfitting, in which the error for the training set remains low but the validation error successively increases. Cross-validation is performed by dividing one dataset into two subsets of data called training and validation data. The classifier is trained using the training data and the performance measurement of the trained classifier is obtained using the validation data [23, pp. 241-249].

2.4.3.1 K-Fold Cross-Validation

K-fold cross-validation (KFCV) divides the dataset into K roughly equally-sized subsets (folds) where one of the subsets is used for validation and the rest $K - 1$ sets are used as training data.

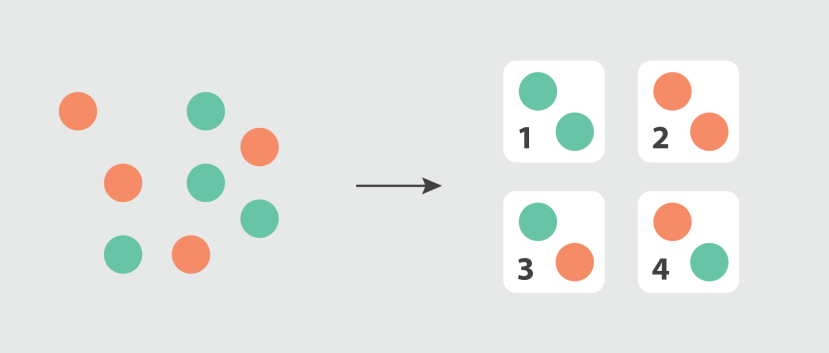


Figure 2.6: Unstructured data points divided into folds.

When the classification has been performed, one of the other K folds is used for validation while the other $K - 1$ sets are used for training. This procedure is repeated (rotated) until all folds have been used for validation. The classification from the previous folds are stored and after all folds have been evaluated, the result is averaged.

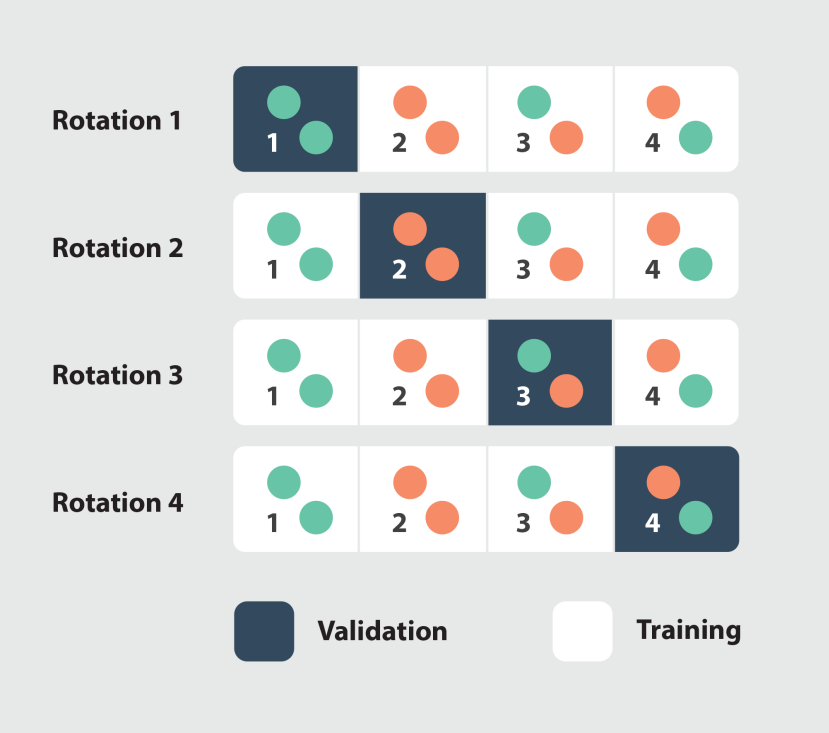


Figure 2.7: The basic idea of training and validation rotation for KFCV based on figure 2.6

After the rotation of which of the subset k is used for validation, and which ones are being used for training, the averaged prediction error can be obtained from the stored outcomes of the evaluation for the previous folds. If $K = N$ where N is the number of data points in the dataset, the cross-validation is called leave one out cross-validation (LOOCV). However, K is an unfixed parameter but are often set to 5 or 10 folds [23, pp. 241-249].

3
Methods

This chapter begins with a description of the data structure and the approach to the general problem. The subsequent sections will focus on explaining the format of the anonymized user data and how the data was preprocessed in terms of data selection, data aggregation and data point combination. Thereafter follows two sections explaining the modelling of the model person and the hierarchical model. The final section gives a detailed description on how the comparison of the different techniques were addressed. This includes which techniques that were used and a justification why these were chosen.

3.1 Data Structure and General Approach

The data collected from users contain information about intakes of food, beverages, supplements, medicals, activities performed and symptom responses. Since the logging of data is manually done by the users of the app and are not automatically logged as in other typical machine learning fields such as automotive, economics and marketing, the data introduces several challenges. Firstly, a considerable amount of noise is expected to be present in the data in forms of delayed, missed or inaccurate data logs.

Since the general data problem is high-dimensional with few data points, a delayed or missed log has a great impact on the possibility to classify if a food log is the source to the symptom outcome or not. However, these aspects are not considered in means of improvement for this thesis since outlier-classification and anomaly detection is a broad area which would be time-consuming to implement properly. Emphasis was instead put on modifying inaccurate logs to reduce the number of dimensions for the problem and increase the number of data points for each dimension. How this was carried out is further discussed in Section 3.2.2.

In addition to the high-dimensionality and noise in the data, the datasets are imbalanced. The imbalance may be due to the fact that users avoid food that they have noticed to give rise to outcomes. The imbalanced dataset problem could have been tackled through introducing over- and undersampling. However, in this case, this could result in loss of information or redistributing the statistical model erroneously which may be counteracting. Instead, performance metrics taking the imbalance into account were applied and some machine learning techniques addressing both classes were included in the comparison.

The research conducted regarding the IBS disease has not yet been able to find

common trends for all sufferers since the triggering factor varies individually. Consequently, there is no ground-truth data available for testing the performance of a machine learning technique. This challenge is bypassed by creating a model of a person's behaviour including several random variables. The model of a person is further discussed in Section 3.3. Using the model person approach to the problem introduces the possibility to evaluate the algorithms on synthetic data.

The general approach toward the problem was that meals consumed by users were considered to be independent random samples from the collection of consumed meals at an individual level. Therefore, the aspect of time is not considered and consequently, the ingredient intolerances for the individuals are assumed to be constant. Moreover, individuals are further considered to either be tolerant or intolerant towards an ingredient. Therefore, if a intolerance ingredient is consumed, the individual will experience a symptom response. Due to the distinctiveness of intolerances for individuals suffering from IBS, one model is trained per individual.

3.2 Data Preprocessing

This section aims to present the format of the raw data in more detail prior to any preprocessing. Thereafter, the steps of selecting, aggregating and combining data will be described in detail which altogether constitutes the applied preprocessing.

3.2.1 Format and Selection

The food and beverage items are composed hierarchically and were, therefore, flattened in order to construct an input-matrix described in subsection 3.2.3. The user logs the symptom response from their food, beverage, supplement and medicine intake. An outcome describes how a user experiences the intensity of a symptom. A typical description of an outcome is "Bloating", "Diarrhea" and "Gases". Moreover, supplement, medicine and activity logs are also composed of single-layer logs. Only symptom responses directly connected to stomach issues were used for benchmarking.

The users of the application have generated different amount of data. Therefore, in order to select individuals with significant number of logs, a threshold based on file size was selected. The user of the application has generated different amount of data. Therefore, in order to select individuals with significant number of logs, a threshold based on file size was selected. Limiting the selection to few users would result in a bias towards these user's specific lifestyle. Therefore, a lower threshold was chosen in order to put emphasis on a quantitative selection as the foundation for the benchmarking. Figure 3.1 below visualizes the distribution of the user data file sizes and which interval of files that was used for benchmarking. The interval corresponds to roughly the top 10% datasets sorted by size and corresponds to the shaded area in the figure.

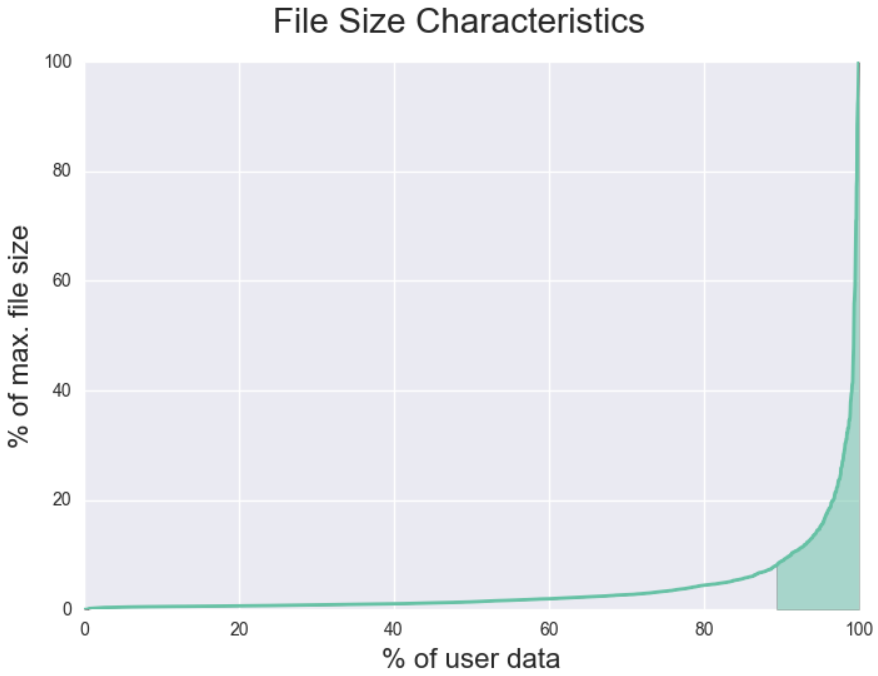


Figure 3.1: Datasets sorted by file size and the portion used in this thesis

3.2.2 Aggregation and Point Combination

The data generated by the users includes a lot of features with many unique instances of food, beverages, supplements and medications. However, the number of times each feature has been logged varies. For instance, common beverages such as “Water”, “Coffee” and “Tea” are logged frequently while more atypical intakes such as “pickled pears” are more seldomly logged. The most commonly logged food intakes can be seen in Figure 3.2 below.

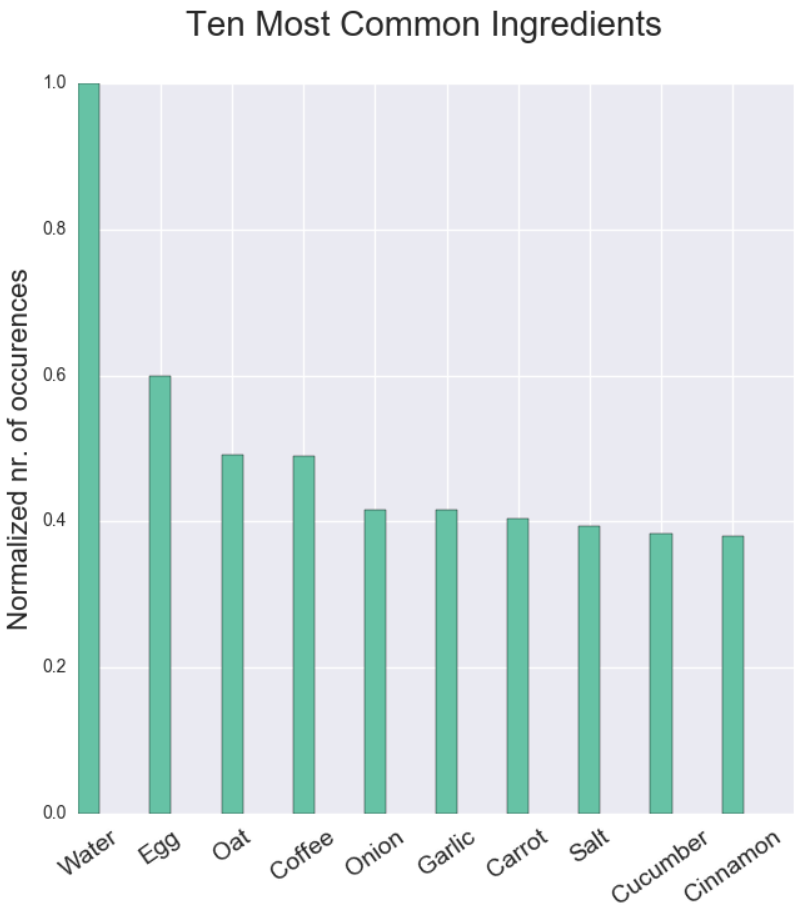


Figure 3.2: 10 most common food ingredients for the users after point combination with normalized number of occurrences

In order to reduce the number of features, similar features were merged. Moreover, the data for the merged features was combined, resulting in higher number of data points per dimension. As an example, user logs with the food item "Butter", were often logged with corresponding fat content (often 60% or 80%) of the butter. Such items were merged, resulting in no separation with respect to the percentage of fat.

The data point combination was done by going through the logged food items manually for the 10 users which have been using the app most frequently. Information in the food items considered excessive was written in a csv-file. An example of information considered superfluous was "Extra Virgin", "Frozen", "Chopped" och "Passed". Moreover, the combination was carried out case-insensitively. After the manual scan, the file of unnecessary words contained approximately 85 entries. This could probably be carried out by using machine learning similar to spam-mail recognition. However, the effort of e.g. creating labeled training data and evaluating the performance was considered as outside the scope of this thesis due to the time budget.

The next step included extracting all the food items and their corresponding unique identifier from datasets of users iteratively. Each food item was scanned to see if any of the words in the csv-file were in the entry. If they were present, they were

removed and the items were given identical identifiers. Moreover, all numerical entries and percent-signs in the food item were removed.

There are however certain drawbacks by using the explained combination method. The performance of the method might differ between users due to individual user behavior. However, the method modifies, on an average computed for 10 individuals, 30.6% of the entries. The data was further rigorously reviewed after the combination method and a negligible part of the food instances were wrongly processed. The method was therefore added as a preprocessing stage when performing benchmarking tests on the anonymized user data.

3.2.3 Construction of Arrays

The construction of arrays starts with combining meals and symptoms into pairs. Thereafter, ingredients which have been connected to a symptom are stored with the corresponding identifier and consumed amount n. Ingredients connected to symptoms at least once are considered as interesting. Ingredients not connected to symptoms are considered as non-intolerances and are therefore removed. Moreover, such procedure further reduces the number of features.

In the next step, an empty matrix is created of size $M \times N$ where M is the number of meals eaten and N is the number of ingredients consumed by the user. The consumed meals are iterated through and if an ingredient i has been eaten in the current meal pair j, the consumed amount is placed in the cell-position (i, j). When all interesting ingredients have been assigned to their corresponding cell, column-wise normalization is performed on the matrix.

In parallel, the symptom intensity experienced after each meal is stored in a column vector. The matrix is further referred to as input-matrix and the column vector as the output-array. Based on if binary classification or regression is to be performed, the output-array is binarized or left unchanged. Binarization intends here that all positions in the output-array that are non-zero are set to one or else, not modified.

3.3 Model Person

In order to evaluate algorithms in aspects of symptom prediction ability and ingredient selection accuracy, synthetic data was generated by a model which mimic the behavior of a user. The model generates data with respect to aspects such as time, meal variance, the number of ingredient allergies and symptom response intensity. Moreover, in order to model the noise in the user data, aspects such as random symptom appearances, omitted events and delayed symptom logs were also considered. The aspects are presented in the following order: time, meal, symptom and noise. For each section, a table is provided to summarize the parameters considered connected to each field of the modeling.

3.3.1 Time Modelling

The start date was selected as the actual day and time the simulation was initiated. The parameter *StartOfDay* was used to model the variations of the awakened state of an user. Analogously, the parameter *EndOfDay* was used to model variations of the sleeping state. A third parameter, *TimeBetweenMeals*, was used to control the time between meals.

Table 3.1: Parameters connected to time modelling

Name	**Description**
StartOfDay	Deciding from which time the model person is awake.
EndOfDay	Deciding from which time the model person is asleep.
TimeBetweenMeals	Parameter for modelling the time between meals.

3.3.2 Meal Modelling

During daytime, the model generates events which are either a logged meal or a logged symptom. However, during nighttime, no events are generated. Food consumption behavior was assumed to coincide with the behavior of the average population. The variance in eaten meal was modeled by *MealChoiceVariance*. *MealChoiceVariance* was expected to mimic the behavior of some meals to be more frequently logged than others. This yielded that 68% of the meals consumed are the *MealChoiceVariance* most commonly occurring meals.

Each meal consists of a number of different ingredients in the range [1, *MaxMealComponents*] which is selected by *ComponentChoiceVariance*. The parameter *MaxMealComponents* was selected to generate similar behavior as a number of ingredients in each meal as for real users visualized in Figure 3.3.

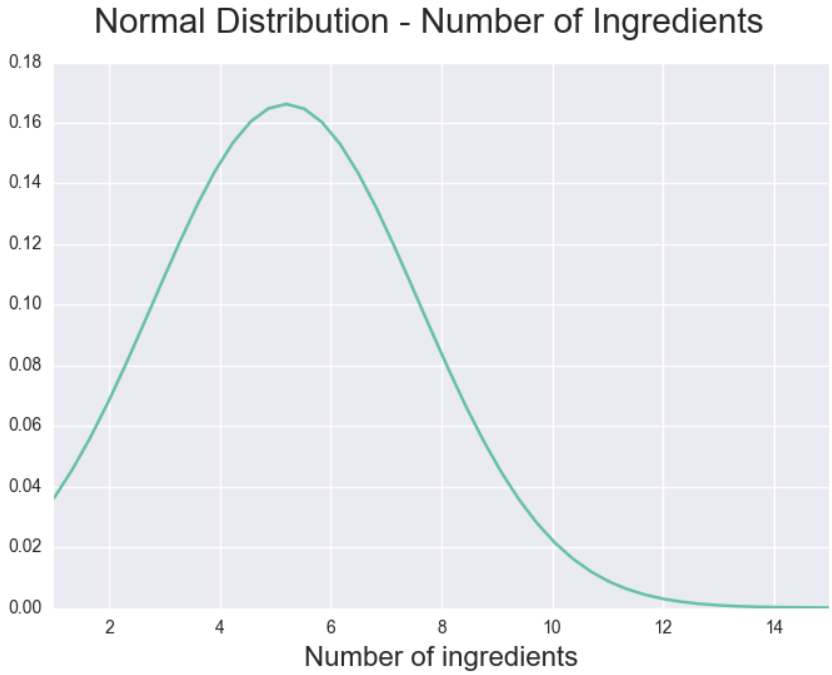


Figure 3.3: Normal distribution generated from user data showing the distribution of number of ingredients in each meal

The amount of each ingredient consumed was modeled in a similar way as the selection of ingredients. The parameter controlling the eaten amount of an ingredient was selected through studying the standard deviation of the amount of food eaten in meals by real users as shown in Figure 3.4 below. First obvious outliers and erroneous logs were removed (abnormally high amounts of food intakes). Moreover, the rolling standard deviation was computed over 50 samples and then used to compute a mean rolling standard deviation. This mean rolling standard deviation was then used to choose the value for the parameter *EatenAmountVariance*.

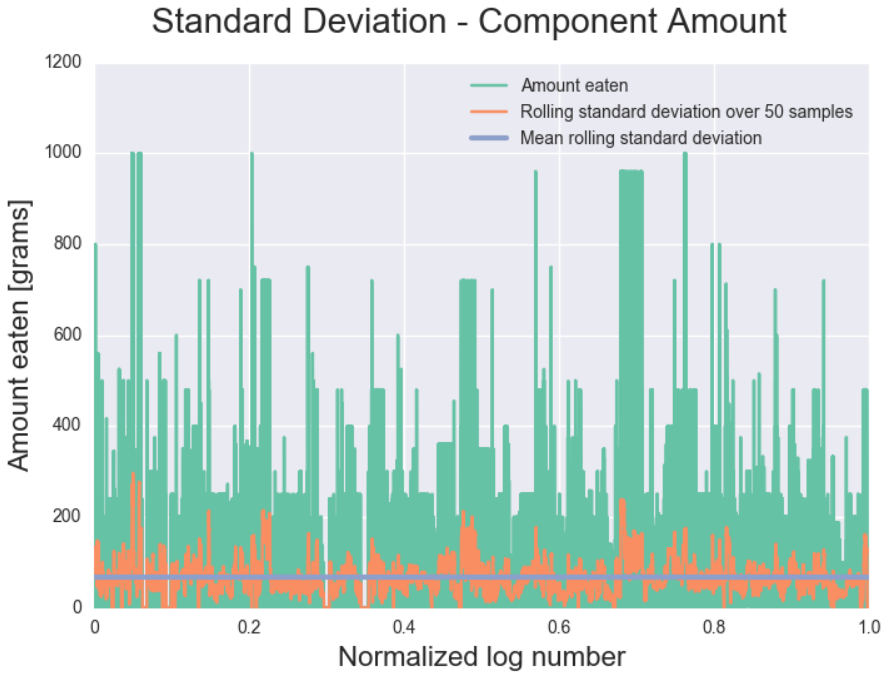


Figure 3.4: Standard deviation of the amount of food intake logged

Table 3.2: Parameters connected to meal modelling

Name	Description
MealChoiceVariance	Amount of variance when picking the meal eaten by the model person.
MaxMealComponents	The upper limit of how many components one meal holds.
ComponentChoiceVariance	Amount of variance when picking the component eaten by the model person.
EatenAmountVariance	The variance in the amount of food eaten from Figure 3.4.

3.3.3 Symptom Modelling

To model the symptom response, the meal consumed was searched through for an intolerance ingredient. The probability that the specific model person is intolerant to an ingredient is modeled by randomly assigning the model person to a probability distribution with either a high or low expected value. If a symptom generating ingredient was found, a symptom intensity response was drawn proportional to the linear combination of the amount consumed of the symptom generating ingredient multiplied with a tolerance factor. The variance, *IntoleranceVariance*, was gathered from real user data similarly as for *EatenAmountVariance* and is shown in Figure 3.5 below. The drawn symptom response was then scaled into the range $(0, 1]$.

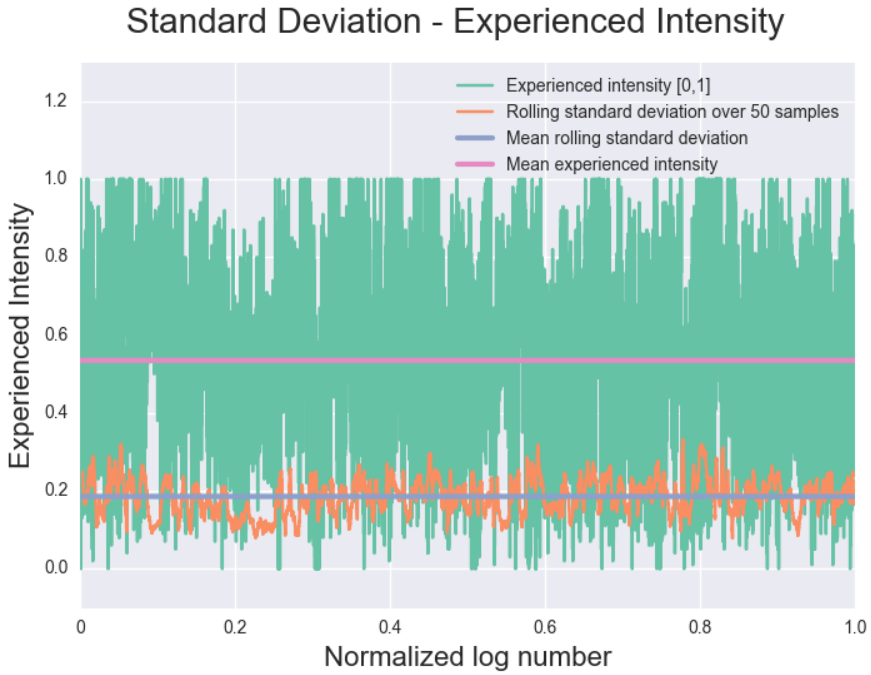


Figure 3.5: Standard deviation of the experienced intensity logged

The number of unique intolerance ingredients of a model person was chosen by the parameter *MaxIntolerants*. A component is modeled to consist of several subcomponents. The number of subcomponents a user is intolerant to determined by a range constituted by *MinIntolerantsSub* and *MaxIntolerantsSub*. A component could be for example "Pasta" while its subcomponents would be for example "Wheat", "Egg" and "Water".

If the timestamp for the occurrence of a symptom is greater than the time of the end of the day, the timestamp of the symptom is postponed to the beginning of the next day. If no symptom generating ingredient were found the intensity was neglected. The maximum delay between food intake and symptom, *DelayOfSymptom*, was set to the time sufferers of IBS claims that it takes for the symptom to arises. This parameter is highly dependent on the type of food consumed, stress levels and other parameters that are difficult to model [5].

Table 3.3: Parameters connected to intolerance and symptom modelling

Name	Description
MaxIntolerants	The maximum amount of components the model person are intolerant to.
IntoleranceVariance	Amount of variance when picking the intensity experienced by the intolerant component.
MinIntolerantsSub	The minimum amount of sub-components in the components that the model person are intolerant to.
MaxIntolerantsSub	The maximum amount of sub-components in the components that the model person are intolerant to.
DelayOfSymptom	The delay between food intake and symptom outcome.

3.3.4 Noise Modelling

In order to mimic the behavior of noise and erroneous logs in the data, five parameters were modeled shown in table 3.4. The first parameter, *MissMealLogRisk* and *MissSymptomLogRisk* probably constitute the most common source of noise. However, both parameters are difficult to approximate from real user data. Consequently, the *MissMealLogRisk* was set to a higher value than *MissSymptomLogRisk* since a symptom has a high impact on the well-being and consequently, the users are assumed to log these occurrences to a higher degree.

The probability that random symptoms are appearing is also included as a noise parameter. The parameter is called *RiskOfRandomSymptom* and simulate the occurrence of symptoms not connected to food intake. The chosen probability of generating a random symptom is based on guessing. The timestamp was selected as a random number contained in the day time interval.

Experienced symptoms might be delayed until the upcoming morning. This is assumed to happen if food is consumed within *DelayOfSymptom* minutes from the time for sleep. By examining at what time real users log data and assuming that average time to go to sleep is 11 PM, one can study the fraction between meals consumed within *DelayOfSymptom* and not. Consequently, the parameter *OvernightDelayOfSymptom* can be estimated as this fraction. The distribution of logging times for meals can be seen below in Figure 3.6.

Figure 3.6: Time distribution for logging food intake among user data

Table 3.4: Parameters connected to noise modelling

Name	Description
MissMealLogRisk	The probability that a meal is not logged.
MissSymptomLogRisk	The probability that a symptom is not logged.
OvernightDelayOfSymptom	Probability that a meal is not logged until the upcoming morning.
RiskOfRandomSymptom	Probability that a random symptom will occur each day. Not assumed to be connected to food intake.

3.4 Hierarchical Model

This section presents the data input structure, followed by a description of the implemented hierarchical model.

3.4.1 Structure of Input Data

As described in Section 3.2.3, the logged data is structured in an output vector and an input matrix which was either binarized or scaled in the range [0, 1]. The hierarchical model utilized the binary version of the matrix and the vector. The data structure for the model was achieved by separating the row indexes in the output array by value, where a value of 1 represented a symptom response and a value of 0 represented no response for each individual. The two sets of indexes were used to extract the rows in the input matrix, connected to symptom response and no symptom response, respectively. The binary structure in the input matrix enabled column-wise summation for the separated rows in order to gain the final input result.

The final data structure for each individual was a $2 \times N$ matrix, containing a number of times the ingredients were connected to a symptom and to no symptom. The above restructuring of data was performed for all individuals used for the construction of priors and was subsequently separated by ingredient and added to a data frame. Therefore, one data frame per ingredient was achieved where each row contained the restructured data result for an individual. Moreover, the data frame was further sorted in ascending order by the ratio: symptom connection/
(symptom connection + no symptom connection).

As described in the symptom modeling section, the users were considered to have either high or low tolerance toward an ingredient. The difference in tolerance between individuals was considered to be represented as abrupt changes of the ratio in the data-frames. The change-point was estimated using the likelihood-ratio procedure presented in the theory section. The data on rows with lower index than the change-point were used for estimating the high-tolerance distribution while the data on rows with higher index were used for estimating the low-tolerance distribution.

3.4.2 Estimating Prior Beta Parameters

As described in the previous section, the data frame for each ingredient, containing the responses from several individuals, were separated into two datasets. For each of the sets, mean value and variance were calculated and subsequently used to estimate the parameters of the two beta distributions. The number of individuals used for estimating priors and symptom ingredient distributions was 300. The estimate of the prior α and β were gained by using the equations described in Section 2.3.3.4. If the criterion of variance was not fulfilled, α and β were set to 1 to represent the ignorance toward the priori knowledge. Moreover, the sum $\alpha + \beta$ was confined to an upper limit of 20 to allow for detection of symptom responses which differ from the general population response. If $\alpha + \beta$ exceeded 20, the procedure presented in Section 2.3.3.4, were applied. After generating priors, the parameter values were stored in csv files.

3.4.3 Usage of Beta Priors for Individual Inferences

The process of making inferences regarding the parameter θ was as follows. For each individual, the data was structured in a $2 \times N$ matrix, where N is the number of individual ingredients consumed by individual I, as described in Section 3.2.3. For each of the N ingredients, the posterior distribution of θ was calculated by the use of equation 2.13, where α and β were the priors estimated as described in the previous section. Since the data was assumed to be random samples from two separate beta distributions, one for high-tolerance and one for low-tolerance, the BF was used to select the model which described the data most accurately. The BF were calculated as described in Section 2.3.3.2, where the probability $P(m_{1,2})$ was set to 0.5. However, due to a large number of individual ingredients, the criterion for selecting distribution was simplified such that only the most likely model was selected.

The final result of the inference was N posterior distributions, one for each ingredient consumed by an individual. However, in order to classify or select ingredients as harmful, a threshold was constructed. The threshold was compared with the mean value of the posterior distribution of the θs. If the mean value of the posterior distribution of θ exceeded the threshold, the ingredient was selected as an intolerance. Different values of the threshold were tested and tuned manually. Examples of tested values were 0.4, 0.5 and 0.6.

3.5 Performance Comparison

This section describes how the evaluation and comparison of the machine learning and the statistical analytics algorithms were conducted. First, a pipeline is introduced explaining the steps performed in order to develop a benchmarking suite for the algorithms. Secondly, the choice of algorithms are presented and motivated.

3.5.1 Benchmarking Pipeline

The benchmarking was performed by running each algorithm on the same datasets in order to obtain comparable results since the class imbalance and number of logged events in the different segments varies. The dataset used is further divided into validation data and training data by splitting it into K consecutive folds according to the KFCV principle. Ten folds were used which is the most common choice according to [23, pp. 241-249].

A list summarizing the full pipeline of the benchmarking is given below:

1. Model Person Evaluation
    - Meal Prediction Ability
        - Classifiers
        - Regressors
    - Symptom Ingredients Classification
    - Dependence on Intolerance Occurrences
2. User Data Evaluation
    - Meal Prediction Ability
        - Classifiers
        - Regressors
    - Effects of Point Combination

3.5.1.1 Model Person

In order to attain an understanding of the algorithms performance in selecting correct intolerance ingredients, the implemented techniques were evaluated with the use of synthetic data which was generated by a model person. The number of unique intolerance ingredients assigned to the model person were 1, 3 and 5. This set was used in order to evaluate region-specific performance for each of the algorithms. The minimum number of generated intolerance occurrences was varied to constitute a 10:1 ratio compared to the number of unique intolerance ingredients assigned to the model person.

For evaluation using the model person, 2000 averaging rounds were performed where each round contributed with a new dataset, generated from a new model person. The number of averaging rounds was determined based on the Law of Large Numbers (LLN) which states that the average result from the experiment tends to be closer to the expected value as the number of trials increases. 2000 trials were considered to achieve a sufficiently converged result as visualized in Figure 3.7 below where the averaged MedAE for LASSO was studied as an arbitrarily chosen constellation.

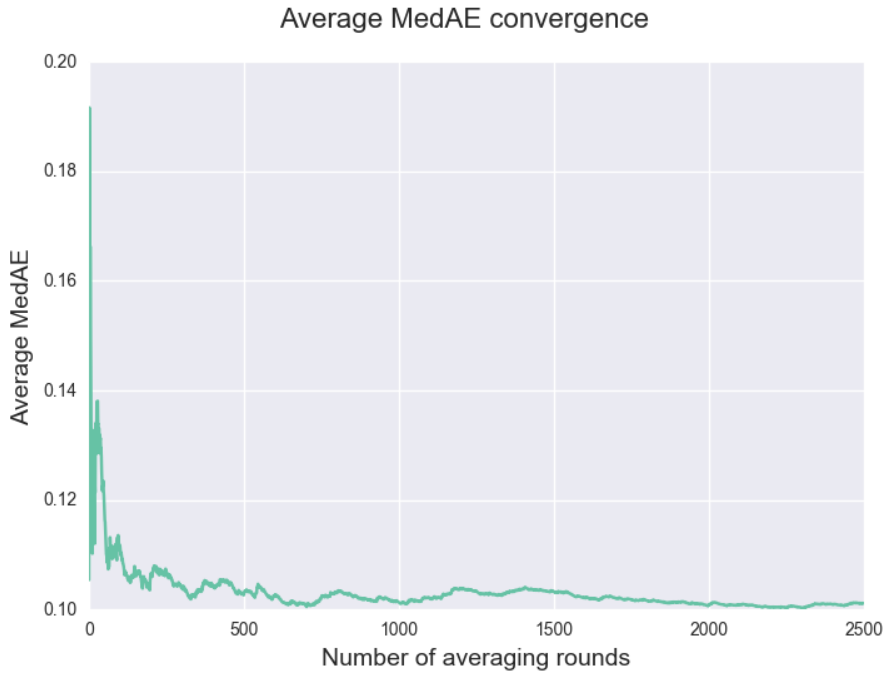


Figure 3.7: Average MedAE convergence for LASSO for varying number of averaging trials

The first part of the evaluation with the model person was performed in order to get a sense of how well the different techniques might perform on answering the question *"Given that I have eaten these meals and reacted accordingly, will I experience a symptom when eating this meal?"*. This could also, indirectly, be a measure on how well the pattern in the dataset has been learnt by the algorithms. This part was implemented by training the algorithms on a training dataset and then evaluate their ability to predict the outcome of a consumed meal featured in the validation dataset. Consequently, this part of the benchmarking does not extract individual ingredients contribution to a symptom but rather the predicted outcome of a full meal. However, the algorithms meal predictive power might correlate with the accuracy of the ingredient classification.

The model person evaluation was initially considered as a classifier problem and answers the question: *"Will the user experience a symptom if eating this meal? Yes/No"*. The classifier evaluation metrics used were accuracy, precision, recall, F1-score, Cohen's Kappa Coefficient, ROC analysis and AUC in order to capture the effects of class imbalance. The classifiers tendency to perform better than random by chance and if there were equal classification ability for majority as for minority classes as described in Section 2.4.1.3.

In the second part of the model-person evaluation, a regressor-approach was taken and consequently, more information was interweaved in the evaluation. This part of the benchmarking aimed to answer the question: *"How high intensity of a symptom will the user experience if eating this meal?"*. Metrics used for this evaluation was MSE, MedAE and the goodness-of-fit measure R^2.

The meal prediction part of the benchmarking aimed to determine which of the classifiers and regressors, separately, that yielded the best prediction power given a certain amount of training data. However, since the ability of selecting correct intolerance ingredients is of greater importance compared to accurate meal predictions, emphasis is put on the algorithms performance on such tasks. This benchmarking is therefore limited to using the model person since it is for real-user data, unknown, which component that is actually contributing to the symptom outcome. However, if the behaviour of the model person mimic real user behavior, the result obtained can be assumed to be similar.

The aim of the next part was to evaluate the algorithms ability to identify intolerance ingredients. The question can be formulated as *"What ingredients are causing a symptom?"*. Through feature importances, the most significant features from each algorithm can be classified as an intolerance ingredient. The intolerance identification allows regressors and classifiers to be benchmarked together and the effects on the result for the different groups can be examined.

In order to select the feature considered as significant, the features importance obtained after training on the data were extracted. A robust method to select these features was coveted since the selection has a great impact on the benchmarking output. A feature is only classified as an intolerance if the feature importance value exceeds a fixed lower threshold which functions as a safeguard. Moreover, the feature importance has to be a part of the 75th percentile of the total number of features for an individual. The generous percentile range was chosen to put emphasis on classifying intolerance ingredients correctly.

The outcome from the intolerance identification is used to construct a second confusion matrix. For each averaging round, the resulting confusion matrix is normalized and stacked position-wise. Each position of the stacked confusion matrix is then divided by the number of averaging rounds. The reason for the normalization in each round is to make allowances for confusion matrices with different number of total elements. The second normalization is a macro-average approach to the problem.

Macro-averaging is sometimes useful. However, in this case, information regarding fluctuation from each result is lost. Therefore, in addition to the macro-averaged confusion matrix, variance is measured to be able to involve fluctuations in the benchmarking. The macro-averaged confusion matrix was used in order to generate the metrics accuracy, precision, recall and F1-score. The Cohen's Kappa Coefficient, ROC curve and AUC is dropped in this part of the evaluation since the characteristics of the methods has already been obtained in previous parts of the analysis.

Finally, an algorithm's ability to classify correct outcomes given as few observed intolerance occurrences as possible were studied. This was achieved by sweeping the number of occurrences a symptom ingredient was consumed in the range [5, 30] with a fixed step size of 1. For each sweep, 2000 averaging rounds were used and plots for the averaged precision and recall were generated for benchmarking. The number of unique ingredient intolerances generated were kept at 3.

3.5.1.2 User Data

The second part of the evaluation was performed on real user data and consequently, no ground truth data was available. Therefore, the meal prediction ability of the algorithms was the only part evaluated. This part of the benchmarking could be argued to be superfluous. However, by cross validating the performance compared to the the meal prediction ability run on the model person, insights on how accurate the model person has been modelled can be obtained. In analogy to the model person evaluation, the structure and performance metrics implemented are the same.

The last part of the evaluation aimed to investigate the effect of the point combination and consequently answer the research question, *"Could the data be used in a more beneficial way?"*. Arbitrarily chosen metrics and classifiers were used to evaluate the performance before and after the point combination.

3.5.2 Algorithms Included

This section intends to give the reader a better understanding and overview of the algorithms used in this thesis. Moreover, an intuition why certain choices were made is also included. A 3-letter abbreviation for each algorithm is introduced.

3.5.2.1 Regression Analysis

The possible success of a linear regression approach is not unwarranted. Each meal S registered by a user constitutes a new row in the input-matrix A. The positions i corresponding to a logged ingredients are updated with the amount eaten of the ingredient. The output-array B, holds values for the experienced intensity of the outcome after eating the ingredients on the position i. The problem can therefore be viewed as the matrix equation $A\mathbf{x} = \mathbf{b}$. Where the linear combinations held by A are weighted through the weighting-vector $\mathbf{x}$, computed through linear regression in LASSO. The LASSO is also well-suited since the shrinkage operator extracts, in this case, the most significant features for each meal and handles the noise in the data.

- **Logistic Regression (LRC)**: The discretized counterpart to linear regression was featured as it is one of the most basic classifiers and its strengths and weaknesses compared to more complex classifiers was interesting to study.

- **Logistic LASSO (LLC)**: Strong theoretical support, the differences and similarities between the regressor and classifier is of interest.

- **Linear Regression (LRR)**: The linear regression was featured to examine if the penalty introduced by regularization methods did result in loss of important information and if a better performance could be obtained without regularization.

- **LASSO Regression (LAR)**: The method with probably the strongest theoretical support. However, the regularization term might not practically fit the problem optimally.

- **Elastic Net (ENR)**: Even though the nature of the problem theoretically is a linear one, the effect of introducing an additional regularization parameter was considered interesting to study.

3.5.2.2 Ensemble Learning

The divide-and-conquer algorithms of ensemble learning does not have the same theoretical support for small datasets as the more tolerant approach of linear regression and LASSO. Consequently, there was a risk that the complexity of the model could not be learned by the ensemble methods. Nevertheless, ensemble algorithms were included in the benchmarking for this thesis. This due to that both classes in a dataset is often addressed under the creation of voters and since the dataset used in this thesis is highly imbalanced, the ensemble approach could still be promising. Two different approaches of random forests and one boosting method were included.

- **Random Forest Classifier (RFC), Random Forest Regression (RFR):** Random Forest was implemented due to the fact that it does not introduce overfitting but also since all classes are addressed during tree creation and therefore does not get affected by the imbalance in the data set.

- **Extremely Randomized Trees Classifier (ETC), Extremely Randomized Trees Regression (ETR):** Extremely Randomized Trees or Extra-Trees is a modification of the RFC and RFR. The split node used in RFC and RFR is the one expected to yield the best result. In ETC however, this split node is chosen randomly. This can introduce marginally better result than for RFC and RFR in some applications.

- **ADABOOST Classifier (ABC), ADABOOST Regressor (ABR):** The ADABOOST Classifier and ADABOOST regressor was implemented due its praised performance by [23, pp. 337-387] which is also mentioned in the theory chapter of this thesis. Moreover, the difference in results from other ensemble learning algorithms is interesting to study.

3.5.2.3 Support Vector Machines

- **SVM Classifier (SVC), SVM Regression (SVR):** The SVM classifier and regressor was implemented to evaluate if the classes in the data could be separated properly with a hyperplane.

3.5.2.4 Bayesian Statistics

The possibility to include prior knowledge to the parameters in order to make inferences could be a possible advantage due to the few data points available at an individual user level.

- **Naive Bayes Classifier (NBC):** Often used in the field of automatic medical diagnosis. The naivety of assuming that each event is independent is often a poor assumption for real-life scenarios, however in many cases, the method performs better than more complex solutions [31]. The simplicity and low computation time of the method was the motivation for including naive Bayes classifier for the evaluation before investigating more complex Bayesian approaches.

- **Hierarchical Model (HMC):** The motivation regarding the use of hierarchical models was based on the assumption that there exist similarities in user responses toward a specific ingredient. Such information can be incorporated in the hierarchical structure, resulting in more informed parameter estimates as well as reduced random sampling noise. In addition, individuals with few data points will to a greater extent be affected by the shrinkage effect. Therefore, hierarchical models were considered as a possible technique for providing users with early traces of ingredients contributing to symptom.

4
Results and Analysis

The previous chapter presented the methods used for reaching the aim and answering the research questions addressed. It consisted of a detailed description of the data preprocessing method, the model person, the hierarchical model and the structure of the performance comparison. This chapter will present the results obtained from the steps described in the method chapter. Moreover, an analysis of the obtained result are included for each of the constellations of model parameters and algorithms. Question formulations from the previous chapter are occasionally repeated to clarify the analysis. For each subsection, significant information from the result is highlighted and discussed in the following paragraphs.

4.1 Result from Model Person

The results presented in this section has been acquired through averaging over 2000 different model persons, a number which was selected based on the the converging performance metric presented in the methodology. The number of meals generated was in the range 150-700 with a mean value of 304. The class imbalance, including the noise, varied in the range 0.16-0.45 with a mean value of 0.3. The first section presents the performance of the 7 classifiers evaluated on the model person with 1, 3 and 5 as set values for the number of unique ingredient intolerances. The reason for the different constellations on the number of unique intolerances is to evaluate each algorithm in different regions of data. The dataset retrieved from using 1 unique symptom generating ingredient will be referred to as the first data region, 3 the second and finally 5 as the third.

4.1.1 Meal Predictive Performance

The predictions presented below aims to classify the outcome of an eaten meal and not an individual ingredient. This approach is examined to obtain an overview of how well the patterns of data has been learned by the algorithms. It is also relevant to identify differences and similarities when, subsequently, performing classification of individual ingredients. This section is deliberately left narrow on discussion since a more exhaustive discussion is carried out in Section 4.1.4 on common metrics and algorithms.

4.1.1.1 Classifiers

This subsection introduces evaluation of the least complicated scenario of classifications considered. Classifiers are compared on how well, given a training dataset, each are able to predict the label of the data points in the validation dataset. The values in a cell correspond to 1, 3 and 5 generated intolerances respectively. The minimum number of generated intolerance occurrences was varied to constitute a 10:1 ratio compared to the number of unique intolerance ingredients generated. The classifier with the highest average value for each metric has been marked out. In this subsection, the result is presented in the first paragraph followed by the analysis. In the end of the analysis, the general findings are summarized.

Table 4.1: Classifier performance on model person for 1, 3 and 5 generated allergies averaged over 2000 model persons

Classifier	**Accuracy**	**Precision**	**Recall**	**F1-score**	**AUC**	**Kappa**
Logistic Regression	0.722, 0.690, 0.685	0.459, 0.682, 0.699	0.396, 0.656, 0.698	0.398, 0.659, 0.693	0.725, 0.748, 0.772	0.270, 0.368, 0.365
Logistic LASSO	0.714, 0.699, 0.702	0.408, 0.692, 0.718	0.388, 0.668, 0.707	0.376, 0.669, 0.707	0.635, 0.742, 0.773	0.250, 0.387, 0.398
Random Forest Classifier	0.752, 0.704, 0.704	0.636, 0.707, 0.722	0.574, 0.654, 0.706	0.570, 0.668, 0.709	0.779, 0.765, 0.783	0.397, 0.395, 0.404
Extra Trees Classifier	0.757, 0.709, 0.707	0.654, 0.734, 0.748	0.541, 0.615, 0.665	0.557, 0.657, 0.698	0.780, 0.759, 0.779	0.396, 0.402, 0.411
ADABOOST Classifier	0.715, 0.679, 0.674	0.552, 0.666, 0.685	0.483, 0.656, 0.697	0.480, 0.651, 0.686	0.670, 0.728, 0.758	0.292, 0.348, 0.343
SVM Classifier	0.718, 0.691, 0.698	0.568, 0.670, 0.731	0.490, 0.693, 0.670	0.487, 0.672, 0.693	0.675, 0.724, 0.748	0.306, 0.374, 0.393
Naive Baye's Classifier	0.646, 0.625, 0.631	0.472, 0.592, 0.647	0.631, 0.679, 0.650	0.508, 0.623, 0.642	0.712, 0.716, 0.729	0.684, 0.250, 0.260

By reviewing the table above in a column-wise manner, it can be seen that the accuracy metric tends to be higher for one intolerance compared to three and five intolerances, for which the accuracy remains approximately constant. The classifiers which achieved the highest average accuracy was the ETC which also gained the highest accuracy in each data region. However, the average accuracy performance between the top-achieving classifiers is approximately the same. The ETC has the highest precision performance in all data regions and does consequently, gain the highest average precision. The NBC attains the greatest average recall. However, it

is outperformed by other classifiers in the second and third data region. The highest average harmonic mean of precision and recall, F1-score, is acquired by RFC which has the highest performance in the first and third data region. However, the F1-score for several classifiers in the second and third region are similar. Furthermore, the top average score for the AUC metric is gained by RFC which is the top-achiever in all regions except for the first. Finally, the marginally highest value for the Cohen's Kappa Coefficient is obtained by ETC which perform best, in relation to the other classifiers, in the second and third data region.

The accuracy score might, as presented in the theory chapter, be misleading due to the accuracy paradox. Moreover, in addition to the effect of the class imbalance, noise in the data further complicates the interpretation of the accuracy result. In our case, there are several random variables introduces to model the noise in user logs. An accuracy of 1.0 might therefore not be optimal since it would correspond to fitting the noise, which is not desireable. Instead, the metrics precision, recall and F1-score need to be examined since they are less affected by the class imbalance.

As introduced in the theory chapter, the precision can be thought of as the percentage of correctly classified positives by an algorithm. In this part of the benchmarking, the precision measure can, therefore, be interpreted as the ratio between the correctly classified intolerance meals and the total number of intolerance meals selected by the algorithm. Moreover, the recall measure in this part of the benchmarking is the percentage of the intolerance meals which were classified correctly as such.

To summarize the results of the meal prediction evaluation, it has been found that tree algorithms perform well in most of the examined regions. The highest average accuracy, precision and Cohen's Kappa was achieved by ETC whilst RFC achieved the highest average, F1-score and AUC. In addition, ETC and RFC further acquired the highest and second highest average value for Cohen's Kappa, indicating that more accurate data patterns have been found. Consequently, these algorithms will be assumed to perform accurately when classifying ingredient intolerances.

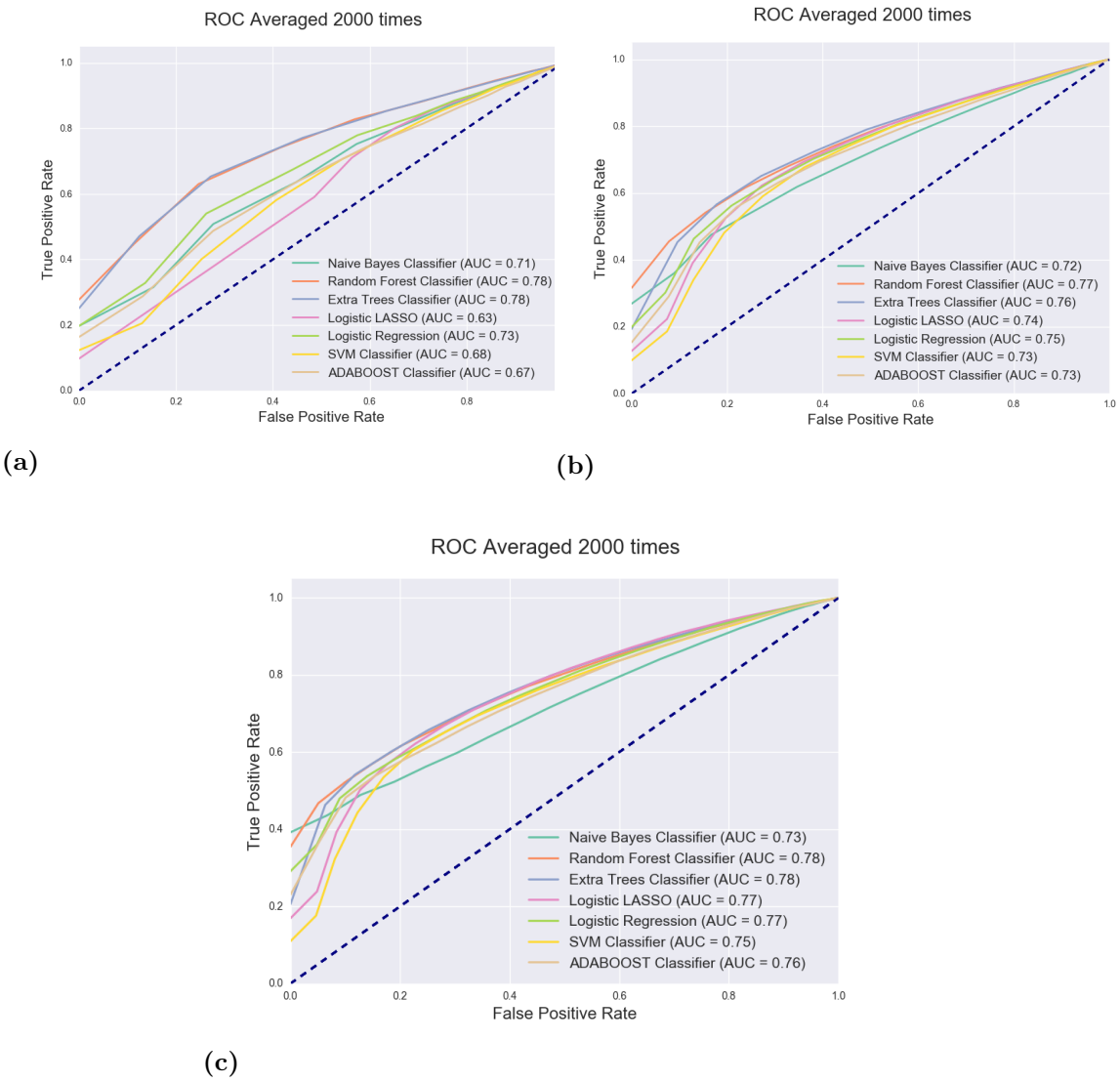


Figure 4.1: Generated ROC curves for all classifiers evaluated for 1, 3 and 5 generated ingredient intolerances

In Figure 4.1, the ROC curves for the classifiers are presented for the first data region in a), second data region in b) and third data region in c). As can be seen, the ETC and RFC achieves similar AUC from their corresponding ROC curves for the first data region. Moreover, the shape of these curves are more desirable compared to the curves obtained by the other classifiers. No classifier lacks predictive power since all curves are above the diagonal. By investigating the next data region presented in Figure 4.1 b, it can be seen that the performance of the classifiers converge to a similar result. The result obtained for region three, presented in Figure 4.1 c, is similar to the second region. However, the AUC measure has marginally increased for all classifiers.

As mentioned in the theory chapter, the FPR on the X-axis can be rewritten as $1 -$ precision. Consequently, the ROC curve can be interpreted as *"How much precision do we sacrifice to achieve the following recall?"* and vice versa. For example, from

Figure 4.1 A, if a recall of 0.9 is desirable for the ETC, a poor precision of 0.25 is obtained. On the contrary, if a precision of 0.9 is preferable, a recall of approximately 0.4 is obtained. However, since parts of the meals are labeled positively due to noise, an ideal ROC-curve with an AUC equal to 1 would indicate overfitting.

4.1.1.2 Regressors

The ability to predict quantitative outcomes given newly introduced data points are evaluated in this subsection. Each cell holds the values for 1, 3 and 5 number of intolerance ingredients as in the previous subsection. The highest average R^2 score has been marked out while the lowest average MSE and MedAE score has been marked out since these indicates higher performance. The results are presented in table 4.2 below.

Table 4.2: Performance of regressors of 1, 3 and 5 symptom ingredients generated by model person

Regressor	**MSE**	**MedAE**	**R^2**
Linear Regression	0.183, 0.168, 0.169	0.176, 0.255, 0.304	0.804, 0.319, 0.184
LASSO	0.157, 0.159, 0.185	0.111, 0.274, 0.369	0.185, 0.036, 0.016
Elastic Net	0.170, 0.142, 0.158	0.108, 0.293, 0.355	0.173, 0.073, 0.154
Random Forest Regression	0.153, 0.134, 0.152	0.109, 0.217, 0.273	0.251, 0.167, 0.233
Extra Trees Regression	0.110, 0.158, 0.156	0.114, 0.204, 0.272	0.355, 0.114, 0.093
ADABOOST Regression	0.205, 0.155, 0.180	0.120, 0.364, 0.443	0.309, 0.122, 0.132
SVM Regression	0.138, 0.139, 0.174	0.115, 0.197, 0.268	0.403, 0.158, 0.218

By reviewing the table 4.2 in column-wise manner, it can be seen that the ETR achieves the lowest average MSE and MSE in the first data region. The top-achiever based on the average MedAE is the SVR, which attains the lowest score for the second and third data region. The highest average value for the R^2 score is achieved by LRR which outperforms the other regressors in the first and second data region.

The noise and the class imbalance in the data may affect the regressor's performance metrics in a similar manner as the accuracy for the classifiers. The analysis

was therefore only based on the relative performance values between the algorithms since no metric considering the noise or class imbalance was used.

Examining table 4.2, one cannot obtain obvious suggestions of algorithm or family of algorithms that achieves higher performance than others. The results are, therefore, in general, non-informative. However, the LRR attains a remarkable R^2 score in the first data region suggesting that the regression line describes the variance in y accurately.

Comparing the obtained R^2 score for all regressors in the first data region with the score obtained in the third data region, a reduction in R^2 score can be seen. This indicates that the predictions of the regressors are more likely to approach the mean of the observed data points as the number of uniquely generated intolerances increases. Therefore, reduced ability to predict variation in y is attained for increasing number of unique intolerances.

4.1.2 Classification of Symptom Generating Ingredient

This section presents the results obtained from the identification of ingredient intolerances. The feature importances were calculated and selected as intolerances if a value above the lower threshold were obtained while belonging to the 75th percentile (third quartile). A generous lower threshold was set as a safeguard. The results are obtained from 2000 averaging rounds with macro-averaged confusion matrices. Due to the macro-averaging, the cell-wise variance was studied in order to detect abnormally high fluctuations. However, the obtained variance was left out since no large fluctuations were detected. In addition to the previously introduced algorithms, the hierarchical model is evaluated in this subsection as well. Based on the result from previous subsections, the tree algorithms from the classifier approach and the LRR from the regressor approach were expected to achieve the highest overall performance.

Table 4.3: Performance of classification for 1, 3 and 5 symptom ingredients generated by model person

Algorithm	**Accuracy**	**Precision**	**Recall**	**F1-score**
Logistic Regression	0.615, 0.572, 0.663	0.727, 0.859, 0.881	0.368, 0.172, 0.377	0.488, 0.287, 0.528
Logistic LASSO	0.662, 0.727, 0.663	0.782, 0.884, 0.852	0.451, 0.523, 0.394	0.572, 0.658, 0.539
Linear Regression	0.774, 0.745, 0.801	0.673, 0.905, 0.881	0.993, 0.547, 0.697	0.799, 0.682, 0.778
LASSO	0.665, 0.697, 0.659	0.252, 0.935, 0.900	0.289, 0.423, 0.358	0.269, 0.582, 0.512
Elastic Net	0.568, 0.631, 0.649	0.668, 0.907, 0.894	0.271, 0.293, 0.339	0.385, 0.443, 0.491
Random Forest Classifier	0.622, 0.664, 0.673	0.766, 0.856, 0.887	0.353, 0.396, 0.396	0.483, 0.541, 0.547
Random Forest Regression	0.774, 0.950, 0.934	0.688, 0.909, 0.883	0.543, 0.816, 1.000	0.615, 0.752, 0.938
Extra Trees Classifier	0.673, 0.645, 0.651	0.808, 0.842, 0.874	0.453, 0.357, 0.354	0.581, 0.501, 0.503
Extra Trees Regression	0.478, 0.611, 0.637	0.402, 0.894, 0.886	0.092, 0.253, 0.313	0.149, 0.394, 0.463
ADABOOST Classifier	0.552, 0.631, 0.624	0.662, 0.830, 0.853	0.213, 0.328, 0.300	0.322, 0.471, 0.444
ADABOOST Regression	0.523, 0.695, 0.659	0.574, 0.935, 0.900	0.182, 0.420, 0.358	0.276, 0.579, 0.512
SVM Classifier	0.487, 0.597, 0.653	0.433, 0.795, 0.875	0.084, 0.262, 0.357	0.140, 0.394, 0.507
SVM Regression	0.529, 0.663, 0.645	0.588, 0.923, 0.892	0.193, 0.355, 0.329	0.290, 0.513, 0.481

Naive Baye's Classifier	0.548, 0.632, 0.713	0.603, 0.831, 0.896	0.279, 0.330, 0.482	0.381, 0.473, 0.626
Hierarchical Model (threshold = 0.4)	0.912, 0.795, 0.812	0.883, 0.819, 0.845	0.960, 0.940, 0.954	0.919, 0.871, 0.896

Reviewing table 4.3 in a column-wise manner, one finds that the HMC acquire the highest average values for all metrics. However, the method is outperformed by RFR in the second and third data region for the accuracy metric. Moreover, the HMC achieves a lower precision compared to all other algorithms except for SVC in the second data region. By inspecting the recall column, it can be seen that the majority of the algorithms attain lower recall scores compared to the HMC. However, the LAR and LRR achieves higher recall score in the first data region while the RFR obtain a higher score in the third. The only algorithm outperforming HMC for the F1-score considering all data regions are RFR which achieves higher F1-score for the last data region.

By separately analyzing the results obtained from linear regression family presented in table 4.3, one finds that LRR has the highest average accuracy, recall and F1-score. For the ensemble learning algorithms, ETC gained the highest average precision and the RFR attains the highest average accuracy, recall and F1-score. However, the difference in average precision for RFR, RFC and ETC was marginal. The SVR approach outperforms the SVC for all average metrics considered. The most superior Bayesian approach evaluated was the HMC which outperformed the NBC for all metrics.

Differences and similarities to meal predictive performance

Since no ground-truth was available for real user data, an aspect considered was to evaluate if the performance of intolerance identification could be derived from the performance of the meal prediction. The best achieving algorithm in the regression family was the LRR. By inspecting the table 4.2, the LRR indicated that the algorithm attained a higher average R^2 score compared to other linear regression algorithms. However, by inspecting the R^2 score obtained for the LRR in all data regions, a decrease in the metric can be seen with increasing intolerances. The same relationships are not visible in table 4.3 when inspecting the obtained F1-score for the LRR. Therefore, a confident conclusion regarding the performance of the ingredient intolerance selection can not be obtained by independently studying the performance of the meal predictions.

For the family of ensemble learning algorithms, the RFR achieved the highest average accuracy, recall and F1-score in the intolerance ingredient extraction. Moreover, the measures improved with increasing number of unique intolerance ingredients. However, the improvement in intolerance identification is not reflected in the meal prediction performance. Therefore, it is difficult to evaluate the performance of the ingredient intolerance extraction based on meal predictive performance for the RFR algorithm.

General comparison of precision and recall
By analyzing the precision in table 4.3, one can see that the metric tends to be higher for three and five intolerances. However, since only 1, 3 and 5 positive examples were available in total, the precision values indicate restrictiveness when selecting ingredients as intolerances. The tendency becomes more apparent if the denominator in equation 2.29 is rewritten as $TP + FP = TP + (\#PP - TP) = \#PP$ resulting in the following fraction $TP/\#PP$. $\#PP$ is the sum of the elements in the left column of the confusion matrix i.e. the total number of ingredients labeled as an intolerance by the algorithm.

In order to attain a higher precision, one could increase the safeguard threshold for selecting intolerance ingredients. Moreover, additional improvements in precision may be gained if the selection window, i.e the third quartile, were narrowed such that only the 90th percentile was included. However, by limiting the selection window, the risk of not detecting the true intolerance ingredients increases which will be reflected in the recall measure as well as the precision measure.

An inspection of the recall measure shows that the performance between algorithms is varying. However, the recall values are in general low for all algorithms except for LRR, LAR, RFR and HMC. Low recall values indicate that the intolerance identification requirements were too restrictive. Another explanation is that the selected algorithms perform poorly when extracting ingredients as intolerances. The tendency becomes more distinctive if the denominator of the recall metric is rewritten as $TP + FN = TP + (\#intolerances - TP) = \#intolerances$ resulting in the following fraction $TP/\#intolerances$. $\#intolerances$ is the total number of unique ingredient intolerances and is attained by adding the elements in the upper row in the confusion matrix.

The recall measure can be improved by reducing the safeguard threshold for selecting intolerance ingredients as well as by expanding the selection window. However, such changes may influence the precision measure negatively since it would allow for more $\#PP$. Moreover, such behavior is not desirable since it lowers the confidence requirements for ingredient selection, resulting in an excessive amount of ingredients marked as intolerances by the algorithm.

Family-wise comparison of precision and recall
A justification for the higher performance achieved by the LRR compared to the other algorithms in the linear regression family could be due to the effect of the regularization. The regularization limits the number of possible positives, resulting in a higher precision. Moreover, the low recall score achieved by the LLC further supports the explanation. However, by comparing the differences in performance between the LAR and the LRR, it can be seen that gains in performance are expected if the regularization is decreased. By studying the linear regression classifiers, the same relationship could not be detected.

Based on the result and discussion in this subsection, we consider the LRR as the most promising of the tested linear regression-based algorithms since it has the highest average accuracy, recall and F1-score.

The reason for RFR achieving higher average scores for accuracy, recall and F1-score than the other ensemble algorithms might be explained by the selection of splitting-points. The splitting-points are chosen to yield the optimal split in RFR and RFC compared to the ETC which chooses the splitting-points randomly. The random choice of splitting-point might be counteracting for small datasets since it increases the variance for the individual trees. A possible explanation for the poor results retrieved for boosting algorithms may be due to the lack of adequate weighting.

Highest performing algorithms
An algorithm which obtains high precision and recall simultaneously is always desirable. The F1-score is, therefore, a measure of great importance in this analysis. The algorithms with good performance with respect to F1-score was RFR and HMC. However, HMC has a higher performance for one and three intolerance while the RFR has higher performance in the last data region.

Due to the chosen ratio between intolerance occurrences and the number of ingredient intolerances, the average number of consumed meals are likely to be higher when five intolerances are used compared to one and three. Therefore, one may assume that the RFR outperforms the HMC algorithm on a larger dataset. However, by inspecting the figures in the following subsection, it is visible that such relationship does not exist. However, the RFR has the ability to capture linear and nonlinear relationships in the data, which the HMC has not.

The highest average score for all metrics was achieved by the HMC. One contributing factor may be the procedure utilized by the model person in order to generate symptom responses. As expressed in the Section 3.3.3, a symptom response was obtained if an intolerance ingredient was consumed. This concept, where single ingredients generate symptoms is supported by the hierarchical model. Therefore, the HMC is likely to find the majority of intolerances as the number of meals consumed increases since it counts the number of times an ingredient has been connected to a symptom. However, an issue emerging from viewing ingredients as independent features is the inability to find the linear and nonlinear relationship between combinations of ingredients. Therefore, symptoms which arise from such connections will not be detected. In addition, the HMC does not take the consumed amount into consideration and consequently intolerances thresholded based on the amount will not be captured accurately.

4.1.3 Dependence on Intolerance Occurrences

To answer the question *"What would be the most suitable algorithm to use based on benchmarking?"*, an important aspect is to consider the algorithm's ability to classify correct intolerance ingredients given a few number of observed intolerance occurrences. In this section, the minimum amount of generated intolerance occurrences is swept in the range [5, 30] with fixed step size equal to 1. For each step, 2000 averaging rounds were used when generating the plots for benchmarking, displaying the averaged precision and recall. The number of unique ingredient intolerances generated was held at 3 as this constitutes a 10:1 relationship towards the maximum

number of intolerance occurrences.

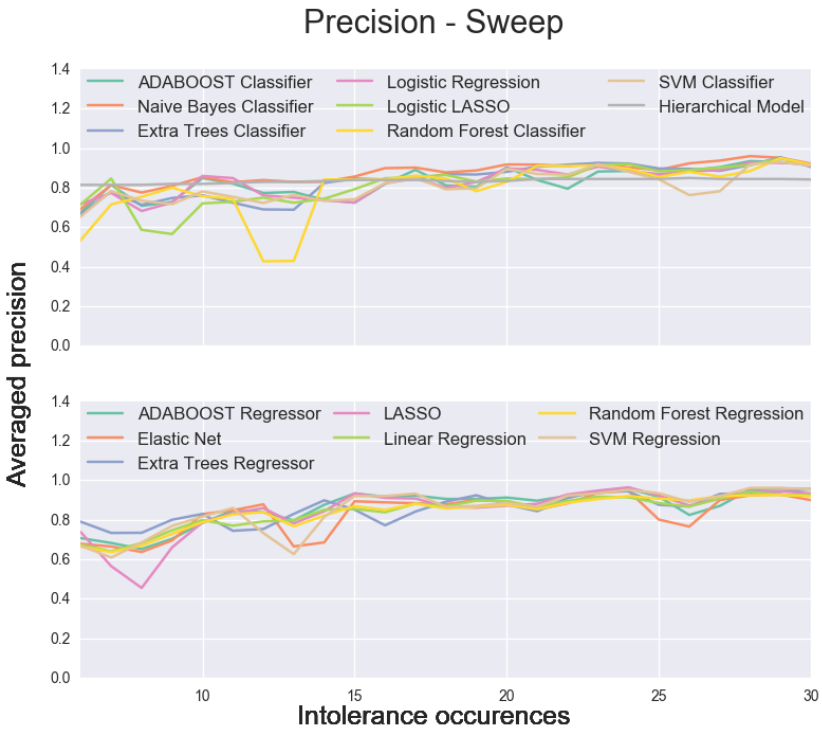


Figure 4.2: Precision averaged 2000 rounds and swept in the range between [5, 30] number of intolerance occurrences

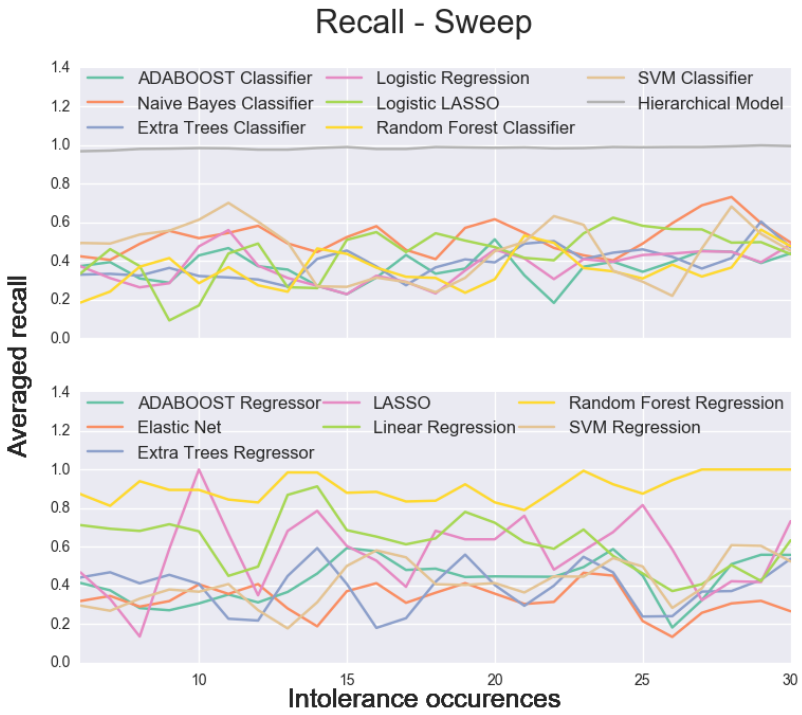


Figure 4.3: Recall averaged 2000 rounds and swept in the range between [5, 30] number of intolerance occurrences

The classifier and regressor achieving the highest precision given the lowest amount of intolerance occurrences is HMC and ETR as can be seen in Figure 4.2. Moreover, fluctuations in precision for different algorithms is also visible in the figure. The general trend is an increasing precision with increasing number of intolerance occurrences.

For the recall sweep, the HMC outperforms the other algorithms and achieves almost an optimal recall in all data point regions. The regressor achieving the highest recall given few intolerance occurrences is RFR. The recall performance for RFR remains high throughout the span of data regions but are more fluctuating than the HMC.

The most obvious finding to emerge from the results is the high performance of the HMC for both precision and recall. HMC's performance is however not unwarranted since the method has before the data points from the sweep are introduced, obtained a prior knowledge from a general population which coincides well with the behavior of a single model persons's behavior.

The general trend of the increasing precision with increasing number of intolerance occurrences may be explained by a decrease in incorrect labeling of the positive class. This could also be interpreted as that the algorithms are more confident when making positive class predictions. Moreover, these results are likely to be related to the increase in meal variance, which increases with the number of consumed meals. Therefore, ingredients consumed in meals with the intolerance ingredient is also more likely to be consumed in absence from it in other meals.

One unanticipated finding was the negligible increase for the recall metric considering all algorithms. This may, as previously mentioned, be explained by the fact that the selection window of intolerance ingredients is too narrow or that the safeguard threshold were set to high. This finding might also indicate that selecting the 75th percentile might not be suitable. Instead, a fixed value could have been used for thresholding. This, in addition to the lower safeguard, may also explain the fluctuations for the curve to some extent since the recall might be restricted to increase for particular regions. Another explanation could be that the algorithms do not find the patterns in the data or that too little data has been generated.

4.1.4 Summary of the Model Person Evaluation

To summarize the findings of this section, differences and similarities between meal prediction performance and ingredient classification performance were identified. The impression of similarities in meal predictive power and ingredient classification was obtained for some regressors including RFR and LRR. However, this correlation is difficult to confirm since the variations in R^2 score for meal predictions was not reflected in e.g. the F1-score for the intolerance selection.

The precision and recall comparison for the ingredient intolerance detection indicated that an increased selection window size might improve the recall measure while simultaneously affecting the result of the precision negatively.

The family-wise comparison between the algorithms showed that the LRR, RFR, SVR and HMC had the highest performance for their respective family. Moreover, HMC achieves the highest average score for all metric considering all algorithms. The sweep over intolerance occurrences further supported the high performance of the HMC.

Before drawing conclusions for the performance of the algorithms on user data, the results presented in this section need to be interpreted with caution. There are several sources of uncertainties which may alter the implications of the obtained result. An example is the process of generating synthetic data which may not capture the behaviour of user data accurately.

4.2 Result from User Data

For this part of the evaluation, no ground truth data was available. Therefore, the aim was to evaluate how realistically the model person has been modeled by highlighting the differences or similarities from the previous section. This may also indicate on how trustworthy the result and discussion from the previous section is. Finally, the effect of the point combination is evaluated in this section.

4.2.1 Meal Predictive Performance

The results presented in this subsection has been generated from anonymized user data. The number of logs used were the number of logs corresponding to the 90th

percentile of the logs ordered by file size and consequently a number of data files. The result has also been preprocessed by point combination of data. The result achieved without point combination of data will be presented in the following subsection. The aim in this section is in analogy to Section 4.1.1 i.e. to predict symptom outcomes from meals and not ingredient intolerances.

4.2.1.1 Classifiers

Table 4.4: Classifier performance on user data with data point combination

Classifier	Accuracy	Precision	Recall	F1-score	AUC	Kappa
Logistic Regression	0.810	0.118	0.116	0.106	0.287	0.025
Logistic LASSO	0.814	0.107	0.124	0.111	0.290	0.035
Random Forest Classifier	0.805	0.160	0.176	0.160	0.298	0.066
Extra Trees Classifier	0.808	0.153	0.166	0.152	0.294	0.060
ADABOOST Classifier	0.779	0.131	0.134	0.124	0.258	0.015
SVM Classifier	0.722	0.143	0.228	0.167	0.259	0.035
Naive Baye's Classifier	0.781	0.183	0.214	0.173	0.317	0.071

The general behavior in table 4.4 is differing from the result obtained for the model person in table 4.1. A significant difference is a major drop in all measures except for accuracy. The marginally highest accuracy is achieved by the LLC. The NBC achieves the highest precision, F1-score, AUC and Kappa score while the SVC obtained the highest recall. However, all algorithms have a low AUC measures, indicating a low probability of ranking a randomly chosen positively labeled meal higher than a randomly selected meal with a negative label. Furthermore, all algorithms obtained low kappa score, indicating that the p_o in equation 2.32 is approximately equal to the probability of agreeing by chance.

The obvious difference in result compared to Section 4.1.1.1 indicates that the model person does not emulate the 90th percentile of the user's data accurately. Therefore, the results obtained in the Section 4.1.1.1 should not be considered to be fully correct for real users. However, the 90th percentile of the user data might still be biased towards these selected individual's behavior and lifestyle and consequently, the result should be interpreted cautiously.

4.2.1.2 Regressors

Table 4.5: Regressor performance on user data with data point combination

Regressor	MSE	MedAE	R^2
Linear Regression	0.141	0.170	0.223
LASSO	0.048	0.131	0.029
Elastic Net	0.047	0.132	0.027
Random Forest Regression	0.061	0.098	0.076
Extra Trees Regression	0.065	0.091	0.068
ADABOOST Regression	0.056	0.128	0.087
SVM Regression	0.058	0.132	0.069

Data from table 4.5 can be compared with the data in table 4.2 which indicates differences that could also be seen for the classifier comparison. Moreover, LRR does, similarly, as for the benchmarking of the model person achieve the highest R^2 score. Such a consistency was not detected for the classifiers and supports the performance presented for LRR for the model person as well as for the 90th percentile user data.

4.2.2 Effect of Point Combination

This section presents the effect of the point combination described in the method chapter. The aim is to answer the research question *"Can the data be modified in a more beneficial way?"*. The evaluation is based on observing the difference in predictive meal performance attained when point combination is applied. The regressor performance with point combination is shown in table 4.5 and the performance without is shown below in table 4.6.

Table 4.6: Regressor performance on user data without data point combination

Regressor	MSE	MedAE	R^2
Linear Regression	0.178	0.232	0.144
LASSO	0.060	0.166	0.019
Elastic Net	0.059	0.163	0.017
Random Forest Regression	0.080	0.128	0.049
Extra Trees Regression	0.082	0.113	0.058
ADABOOST Regression	0.071	0.144	0.023
SVM Regression	0.069	0.156	0.047

By comparing the tables column-wise, it is noticeable that a lower MSE is obtained for all regressors with the point combination. This is expected since all regressors benefit from reducing the number of dimensions while increasing the number of data points for each dimension. The same scenario is observed for the MedAE and R^2 metric due to the same reasons. The R^2 score is for some regressors twice as high for the point combination than without. This means that the regression line fitted

by the regressors are twice as good fit as the simple average. This is an interesting finding and indicates the importance of preprocessing, dimensionality reduction and influence of more data points.

4.2.3 Summary of the User Data Evaluation

The implications from the user data evaluation showed that the classifier's performance varied compared to the classifier's performance for the model person. This gave the impression that the model person did not mimic the general behavior of the 90th percentile of the users. Moreover, the results gained from the regressors further supports this impression. However, some parity might be visible in the R^2 of the LLR. Finally, this section implied that the data can be more beneficially used by e.g. point combination. However, the improvement in performance was not substantial.

4.3 Possibilities for Improvement

There is abundant room for improvements in order to obtain the most optimal analysis engine. During this thesis, the focus has been on evaluating pre-implemented machine learning algorithms with no notable parameter tuning or optimization in mind. This parameter tuning would be a natural next step to examine in the future.

The predictions done by all algorithms in this thesis were based on that there were no synergy effects between food components and consequently, symptom appearances caused by combinations of ingredients were not considered nor detectable. The effect of extending the analysis to finding synergy between food components would be an interesting future implementation. Moreover, improving the model person to support such behavior may increase the similarity between the synthetic and user data.

An interesting future study for the research within the field of IBS would be to perform clustering of individuals in order to bundle individuals with similar intolerances. A possible advantage is that symptom intolerance may be inferred from the cluster populations which in turn would enable earlier symptom intolerance detection at a individuals level. Moreover, such a structure may be represented by a hierarchical model. However, to perform reliable clustering, more data is probably needed.

Except for estimating priors for the hierarchical model from different clusters that users are assumed to belong to, additional information obtained from medical research on the disease could be interwoven. Examples of such information could be to increase the probability of an outcome for food consisting of high FODMAP components. An investigation regarding the optimal thresholding could probably improve the performance of the hierarchical model. Moreover, another possible area of improvement is the selection of threshold used for feature extraction.

The evaluation could have been extended to examine how many data points that are necessary in order to achieve 100% correct classification of the correct ingredient intolerances, given the ground truth. Moreover, the usage of a proportional hazard

model to evaluate how different factors (ingredients) influence the degree of symptom probability in time could also be investigated. This could possibly also be used for estimating the time of the predicted symptom outcome is expected to occur [32].

Further research could be undertaken to investigate if the point combination could be performed more dynamically using machine learning. Other actions considering using data more advantageously could also be evaluated in future research. For instance, ingredient data may be clustered into larger groups based on similarity. An example would be to cluster the dairy products such as milk and cream. Such procedures would lower the number of features while simultaneously increasing the number of data points within each dimension. Another possible preprocessing area of improvement would be, instead of clustering ingredients, divide them into their nutrient components or subcomponents.

5
Conclusion

The main goal was to identify available techniques to algorithmically identify causal relationships between food intake and stomach issues from patient's self-recorded journals, utilizing machine learning and statistical analytics. The second aim of was to identify which of the investigated algorithms that are the most optimal for the datasets available. Evaluation of more beneficial usage of data was also included in the investigation.

This study has identified that a correlation between meal predictive power and ingredient classification is difficult to confirm for synthetic data. However, the result indicated that linear regression might have similarities in meal predictive power and ingredient classification. The linear regression outperformed the other algorithms within the family of regression analysis for most evaluation metrics examined.

The family-wise comparison on the model person showed that linear regression, random forest regression, support vector machines regression and hierarchical modeling had the highest performance. Moreover, the hierarchical model achieves the highest average score for all metric considering all algorithms. The sweep over intolerance occurrences further supported the high performance of the hierarchical model. The result for the random forest regression from the family-wise comparison was enhanced by the sweep over intolerance occurrences.

The comparison of the result obtained from the model person with the result obtained from the 90th percentile of the real user data based on meal predictive performance gave the impression that the model person did not mimic the general behavior of the users. Moreover, by combining data points as a preprocessing stage, improvement in performance was obtained. However, the increase in performance was not substantial.

The hierarchical model acquired the highest average performance for all metrics considered. However, the model does not support detection of linear or nonlinear relationships between combinations of consumed ingredients and outcomes. In addition, the hierarchical model does not take the consumed amount into consideration. Consequently, intolerances thresholded based on the amount will not be captured accurately. Due to the unknown symptom generating behavior of users, the limitations of the model may affect its performance significantly when applied to real user data.

Due to sources of uncertainties, the presented findings must be interpreted with caution. The main source of uncertainty is the lack of knowledge regarding the

difference between the model person and real users. However, the model person is believed that to some extent model real user behaviours and therefore gives indications of the algorithmic performance.

To conclude, we believe that utilizing the hierarchical model in combination with another algorithm may be useful for the analysis of the available dataset. This is motivated by the hierarchical model's ability to detect intolerances given few data points. As the data size increases, more complex structures may be detectable and captured by other algorithms.

Further research should be undertaken to investigate the possibility of utilizing more complex algorithms to detect more complicated structures. Moreover, efforts to cluster individuals based on similar ingredient intolerances may improve the knowledge in the field of IBS. Such knowledge can further be used to detect ingredient intolerances at an earlier stage by inferring the cluster specific characteristics for individuals.

Bibliography

[1] R. Akbani and T. Korkmaz (2010). *Applications of Support Vector Machines in Bioinformatics and Network Security*, Application of Machine Learning, Yagang Zhang (Ed.), InTech.

[2] E. Alpaydin (2010) *Introduction to machine learning.* MIT Press, Cambridge, pp. 61-65

[3] S. Aminikhanghahi and D. J. Cook (2017). A survey of methods for time series change point detection. *Knowledge and Information Systems* 51.2: 339-367.

[4] P. C. Austin, L.J. Brunner and J.E. Hux. Bayeswatch: an overview of Bayesian statistics. *Journal of Evaluation in Clinical Practice*, vol. 8, 277–286, 2002.

[5] L. Böhn. *Food-related gastrointestinal symptoms, nutrient intake and dietary interventions in patients with irritable bowel syndrome*, Sahlgrenska Academy at University of Gothenburg, 2015.

[6] S. Bouix et al. (2007). On Evaluating Brain Tissue Classifiers without a Ground Truth. *NeuroImage*, vol. 36, pp. 1207–1224.

[7] N. V. Chawla et al. "SMOTE: Synthetic Minority Over-sampling Technique". *Journal of Artificial Intelligence Research*, vol. 16., pp. 321–357, Jun. 2002

[8] M. Camilleri (2012). Peripheral mechanisms in irritable bowel syndrome. *New England Journal of Medicine*, vol. 367, pp. 1626–1635

[9] C. Campbell and Y. Ying (2011). *Learning with Support Vector Machines*, Morgan & Claypool Publishers.

[10] C. Canavan, J. West, and T. Card. (2014). The epidemiology of irritable bowel syndrome. *Clinical Epidemiology*, vol. 6, pp. 71–80.

[11] C. Chen, A. Liaw, and L. Breiman. *Using random forest to learn imbalanced data. University of California*, Berkeley, pages 1–12, 2004.

[12] J. Chen and A. K. Gupta (2010). *Parametric Statistical Change Point Analysis, With applications to Genetics, Medicine, and Finance*, 2nd ed., Birkhäuser.

[13] A. Ng, 'Support Vector Machines - Week 7', coursera.org, 2017.

[14] F. Creed et al. (2001). Health-related quality of life and health care costs in severe, refractory irritable bowel syndrome. *Annals of Internal Medicine*, vol. 134, pp. 860-906.

[15] C. Cremon et al. (2009). Mucosal immune activation in irritable bowel syndrome: gender-dependence and association with digestive symptoms. *The American Journal of Gastroenterology*, vol. 104, pp. 392-400.

[16] B. de Ville (2013). *Decision trees.* WIREs Comput Stat, vol. 5: pp. 448–455.

[17] T. Fawcett, An Introduction to ROC Analysis. *Pattern Recognition Letters*, vol. 27., pp. 861-874, 2006.

[18] A. Gelman et al (2014). *Bayesian Data Analysis*. 3rd ed., CRC Press, Chapman & Hall, Boca Raton, FL.

[19] P. Geurts et al. (2005). Extremely randomized trees [Online]. Available: https://pdfs.semanticscholar.org/336a/165c17c9c56160d332b9f4a2b403fccbdbfb.pdf [Accessed: 2017-05-26]

[20] M. E. Glickman and D. A. van Dyk (2007). *Topics in Biostatistics: Basic Bayesian Methods*. 1st ed., Humana Press, pp. 319-338.

[21] C. Goutte and E. Gaussier (2005). A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. *In European Conference on Information Retrieval*, pp. 345–359, Springer

[22] H. Haibo, and M. Yunqian. "Foundations of Imbalanced Learning," in *Imbalanced Learning: Foundations, Algorithms, and Applications*, 1, Wiley-IEEE Press, 2013, pp.216-

[23] T. Hastie, R. Tibshirani, and J. Friedman (2009). *The Elements of Statistical Learning*, 2nd ed., ser. Springer Series in Statistics, Springer.

[24] D. G. Kleinbaum and M. Klein (2010). *Logistic Regression: A Self-Learning Text. Statistic for Biology and Health*, Springer Science+Business Media, pp. 1-39.

[25] I. Kononenko (2001). Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in Medicine*, vol. 23, pp. 89 – 109

[26] J. K. Kruschke (2015). *Doing Bayesian Data Analysis*, 2nd edition, Academic Press.

[27] S. Lessmann et al. An Evaluation of Discrete Support Vector Machines for Cost-Sensitive Learning, *International Joint Conference on Neural Networks*, pp. 347-354, 2006.

[28] R. M. Lovell, F. C. Ford (2012). Effect of Gender on Prevalence of Irritable Bowel Syndrome in the Community: Systematic Review and MetaAnalysis. *The American Journal of Gastroenterology*, vol. 107, pp. 991-1000.

[29] M. L. McHugh. Interrater reliability: the kappa statistic. *Biochemia Medica*, vol. 22., pp. 276–282, 2012.

[30] NIST/SEMATECH e-Handbook of Statistical Methods [Online]. Available: http://www.itl.nist.gov/div898/handbook/ (Accessed 2017-05-18).

[31] I. Rish. *An empirical study of the naive Bayes classifier*, T.J. Watson Research Center. IBM.

[32] G. Rodríguez (2007). Lecture Notes on Generalized Linear Models [Online]. Available: http://data.princeton.edu/wws509/notes/c7.pdf [Accessed: 2017-05-23]

[33] R.S. Sandler et al. (2002). The burden of selected digestive diseases in the United States. *Gastroentology*, vol. 122, pp. 1500-1511

[34] J. Shi et al. (2013). Perceptual Decision Making “Through the Eyes” of a Large-Scale Neural Model of V1, *Frontiers in Psychology*, vol. 4.

[35] M. Simrén et al. (2004). Quality of life and illness costs in irritable bowel syndrome. *Digestion*, vol 69, pp. 254-61.

[36] M. Simrén et al. (2001). Food-Related Gastrointestinal Symptoms in the Irritable Bowel Syndrome. *Digestion*, vol. 63, pp. 108-115

[37] Socialstyrelsen (2017). Statistik om hälso- och sjukvårdspersonal [Online]. Available: http://www.socialstyrelsen.se/statistik/statistikefteramne/halso-ochsjukvardspersonal (Accessed: 2017-04-12)

[38] S. J. Taylor and B. Letham (2017). Forecasting at Scale [Online]. Available: https://facebookincubator.github.io/prophet/static/prophet_paper_20170113.pdf [Accessed 2017-05-22]

[39] R. Tibshirani. Regression Shrinkage and Selection via the lasso. *Journal of the Royal Statistical Society.* Series B (methodological) vol. 58. Wiley: 267–88, 1996.

[40] C. J. Tuck et al. (2014). Fermentable oligosaccharides, disaccharides, monosaccharides and polyols: role in irritable bowel syndrome, *Expert Review of Gastroenterology & Hepatology*, vol. 8, pp. 819-834

[41] F. J. Valverde-Albacete, and C. Peláez-Moreno. *100% Classification Accuracy Considered Harmful: The Normalized Information Transfer Factor Explains the Accuracy Paradox*, 2014.

[42] D. von Seggern (2006). *CRC Standard Curves and Surfaces with Mathematica.* Chapman & Hall, 2nd ed, pp. 148.

[43] R. Wetzels and E. J. Wagenmakers. A Default Bayesian Hypothesis Test for Correlations and Partial Correlations. *Psychonomic Bulletin & Review* 19.6, pp. 1057–1064, 2007.

[44] Z. R. Yang. Biological applications of support vector machines, *Brief Bioinform*, vol. 5, pp. 328-338, 2004.