Florian Fuchs, BSc.

High Dimensional Feature Selection Methods and Application to Styrian Wine Grape Data

MASTER'S THESIS

to achieve the university degree of

Diplom-Ingenieur

Master's degree programme: Mathematics

submitted to

Graz University of Technology

Supervisor

Ao.Univ.-Prof.Dipl.-Ing.Dr.techn. Friedl Herwig

Institute of Statistics

Graz, February 2021

Abstract

Nowadays classification problems occur in almost all research areas such as in natural science, economics or engineering. Embedded in the concept of statistical modeling, a wide range of methodology and theory is developed for this type of problem and the selection of appropriate features is always crucial. Especially, a statistical challenge is the situation of large p and small n ($p >> n$) where the number of available features p is much larger than the sample size n. Here, most methods fail or are simply not applicable. In this thesis, the general concept of feature selection for classification problems considering flat features is introduced and the application for the large p and small n case is discussed in detail. Furthermore, the multinomial logistic classifier derived from the multinomial logistic model, which is a generalization of the logistic regression model, is described and the theoretical foundation, i.e. classical linear regression and generalized linear models, is provided. Afterwards, using the Styrian wine grape data, these methods are illustrated and the possibility of classifying the variety and geographical origin of wine grapes is investigated based on the results of a high-performance liquid chromatography in combination with a time-of-flight mass spectrometer.

Zusammenfassung

Heutzutage treten Klassifikationsprobleme in fast allen Bereichen wie Naturwissenschaften, Wirtschaftswissenschaften oder Ingenieurswissenschaften auf. Eingebettet in das Konzept der statistischen Modellierung wurde eine Vielfalt an Methodik und Theorie für diese Art von Problem entwickelt. Entscheidend ist dabei die Auswahl passender Prädiktoren (Features). Statistisch besonders herausfordernd ist die Situation, wenn p groß und n klein ($p >> n$) ist, hierbei ist die Zahl an verfügbaren Features p deutlich größer als der Stichprobenumfang n. In diesem Fall schlagen die meisten Methoden fehl oder sind nicht anwendbar. In dieser Arbeit wird das generelle Konzept der Auswahl von Features, unter Verwendung flacher Features, eingeführt und die Anwendung auf den Fall $p >> n$ im Detail diskutiert. Weiters wird der 'multinomial logistic classifier' aus dem multinomialen logistischen Modell hergeleitet. Letzteres ist eine Verallgemeinerung der logistischen Regression. Die theoretischen Grundlagen, d.h. die klassische lineare Regression und die generalisierten linearen Modelle, werden hierzu bereitgestellt. Danach werden diese Methoden verwendet, um mithilfe des steirischen Weintrauben-Datensatzes zu untersuchen, ob es möglich ist die Traubensorte sowie die geographische Herkunft mithilfe der Ergebnisse aus der High-Performance Chromatographie in Kombination mit einem Flugzeit-Massenspektrometer zu bestimmen.

Contents

Introduction

These days, the awareness for local food increases and so do the possibilities given by analytical devices. There is almost no food to which the geographical location is as important as it is for the production of wine. According to concerned winemakers, even less than a kilometer makes a difference regarding the taste.

Therefore, the Institute Dr. Wagner started a project to investigate whether or not it would be possible not only to differentiate between the variety of a wine grape but also between its geographical origin. For this task the analytical methodology of choice was the high-performance liquid chromatography in combination with a Time-of-Flight mass spectrometer. This combination generates data in very high resolution and is normally used to identify specified substances in a chemical sample.

Dealing with this amount of data and classification is not common for colleagues at the Institute Dr. Wagner, which is why the Institute of Statistics at the Graz University of Technology, represented by Prof. Friedl and Prof. Hörmann, was invited to join the project to provide their statistical expertise.

This work starts with a general chapter of linear regression models and their expansion on generalized linear models. This introduces the general theory of this work which is crucial to the following chapters. Chapter 2 then discusses the basic multinomial model which will be used in different variations later on.

Referring the high dimensionality of the data faced in the application Chapter 3 provides a brief introduction and heuristic overview for the concept of feature selection, especially focusing on classification problems. The restriction on classification problems allows to categorize the different methods used in later parts.

After summarizing the major part of the theory, the practical part of this thesis starts with a short explorative analysis of the data used in Chapter 4. After the first overview of the data, a shift of values over the measurements sequence was observable and therefore, Chapter 5 provides a basic method for the correction of the available features to make them more comparable.

The application part of this work is arranged in the last three chapters, where each chapter is based on one type of feature selection, which are derived from Chapter 3.

Chapter 6 is based on filter models, where by an appropriate measure the number of features for the classifier is reduced by 'filtering' all features which are available in the first place. Since this method is used again in Chapter 7 more explorative results are provided at this point.

The methods used in Chapter 7 and 8 are both using the same underlying classification algorithm, the multinomial logistic classifier. Whereas the preselection method (Chapter 7) is an iterative method which sequentially solves multiple optimization problems, the approach with penalization is more a one step method and therefore this step is more complex.

1 Linear Regression and Generalized Linear Model

This chapter provides a brief introduction to the field of statistical modeling, nowadays also often referred to as statistical learning. The later wording points out the usage of statistical modeling in the upcoming field of machine learning. The main idea of the subject and some consistent notation for this master thesis are also derived in the following sections.

Starting with a short overview of the classical linear regression model, the extension required for modeling of binary, counting or categorical data, as in McCullagh and Nelder (1989), is discussed in detail.

Due to the fact that multinomial logistic models with some variations, is the theoretical foundation of this master thesis and used in many different situations as primary model type, the explicit discussion on this topic, along with the implementation in the free statistic software R (c.p. R Development Core Team, 2008), is done at Chapter 2.

At the beginning of every modeling process there is a quantity of interest also called response, target variable or dependent variable. Most of the time this quantity is measured under different circumstances, therefore additional information is available. This additional information is called covariates, predictor variables or specially in classification problems referred to as features.

Assuming that the observed response is a realization of an underlying random variable the claim of statistical modeling is the usage of the additional information to describe the distribution of the response.

Remark 1.1. *(Notation)*
The notation in this master thesis should follow in general the well-known concepts in statistical literature. The difference between a random variable or the realization should be clear from the context but in almost all cases $\boldsymbol{x}$ *describes a vector,* $\boldsymbol{X}$ *a matrix, where* x *represents in general a scalar value.*

Assuming the realizations of a random variable y form the vector $\mathbf{y} = (y_1, \ldots, y_n)^t$, in the following also called response vector if y is describing our quantity of interest. For each observation $y_i,\ i = 1, \ldots, n$, the values of the covariates are provided as row vectors $\mathbf{x}_i := (x_{i1}, \ldots, x_{ip})^t$ and by stacking them row wise together the so called design matrix $\mathbf{X}$ with the form $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^t$ is defined. For the modeling process let $\mathbf{x} = (x_1, \ldots, x_p)^t$ be the general representative of the covariates.

Using an additive error ϵ the resulting general regression model equation is

$$y = f(\mathbf{x}) + \epsilon, \tag{1.1}$$

where $f : \mathbb{R}^p \to \mathbb{R}$ is a function connecting the additional information (covariates) to the response. Due to the generality of the function f there is no practical usage possible without further assumptions on the structure. A special assumption will be discussed in Section 1.1.

In regression analysis there are two major concepts when the assumption on the connecting function f is discussed.

The first and also the traditional one uses a quite strong structural assumption like linearity and introduces only a few parameters for the shape of the function. This concept makes resulting models easier to comprehend and they require less data than other approaches. One disadvantage coming with the structural restrictions is related to the lack of capturing very complex problems which cannot be reflected by this type of model.

The other concept is known as non-parametric regression and tries to approximate the function f in a more technical way, i.e. using B-splines or other polynomial approaches. The advantage is that for more complex problems only the number of parameters increases. The drawback when using this non-parametric approach comes from the fitting procedure. In general, there is more data required to estimate the larger number of parameters in a more or less robust way.

Another point of view in statistical modeling is to use conditions directly connected to the distribution of y. They can be stated by fixing a family of distribution functions and an additional assumption on the dependence between the covariates and the parameters of the distribution. For this type of modeling the according model can be written as

$$y \sim F_y(g(\mathbf{x})). \tag{1.2}$$

Here again the function $g : \mathbb{R}^p \to \mathbb{R}^q$ is not specified further and therefore not of practical use without additional structural assumptions.

One assumption for all models discussed in the following is the independence of the responses $y_i, i = 1, \ldots, n$.

1.1 Classical Linear Regression Model

The classical linear regression model occurs if Equation (1.1) is used with two additional assumptions:

- The deterministic function $f : \mathbb{R}^p \to \mathbb{R}$ is linear in the parameter vector.
- The random term is Gaussian distributed with mean zero, i.e. $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

With the first assumption Equation (1.1) can be written as

$$y = f(\mathbf{x}) + \epsilon = \beta_1 x_1 + \cdots + \beta_p x_p + \epsilon = \mathbf{x}^t \boldsymbol{\beta} + \epsilon, \tag{1.3}$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^t \in \mathbb{R}^p$ is the p-dimensional parameter vector. The second assumption adds an additional parameter σ^2 to the model, which usually needs to be estimated because it is unknown.

A quite common practice is the usage of an intercept as an additional parameter, which is a special case of a covariate. It occurs when the covariate vector is extended by 1 in the first entry and therefore the model equation results in

$$y = (1, x_1, \ldots, x_p)(\beta_0, \beta_1, \ldots, \beta_p)^t + \epsilon = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \epsilon. \tag{1.4}$$

The zero-indexing for the new parameter vector is used to clarify that there are p different quantities used as additional, or linked information but there are $p + 1$ parameters to be estimated in this model and also the parameter σ^2.

The usage of an intercept is important in many practical situations but also has the advantage of allowing to model the response without additional information. In this case the intercept β_0 is equal to the mean $\bar{y}$ of the response vector $\mathbf{y}$.

This methodology allows to compare models with or without additional information and therefore the usage of an intercept will be obligatory in the following if not explicitly stated otherwise.

By using model (1.2) the classical linear model can also be expressed as

$$y \sim \mathcal{N}(\boldsymbol{\theta}) \text{ with } \boldsymbol{\theta} = (\mu, \sigma^2) = (g_1(\mathbf{x}), g_2(\mathbf{x})), \tag{1.5}$$

where in this case $g_1(\mathbf{x}) := \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$ and $g_2(\mathbf{x}) := \sigma^2$.

Remark 1.2.
Notice that the two different definitions of the regression model are not contradicting each other. This can be seen in quite a lot of theoretical concepts mentioning statistical modeling or also machine learning.

For statistical inference a random sample of n observations for the response is assumed in the following. Then under the assumption of the classical linear model above the sample can be written as follows:

$$y_i \overset{ind.}{\sim} \mathcal{N}(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}, \sigma^2) \quad i = 1, \ldots, n.$$

Two aspects of the linear regression analysis linked to the parameter vector $\boldsymbol{\beta}$ are of further interest and therefore discussed in the following sections.

- Estimating the general values of the parameter vector $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^t$.
- Testing the significance for a group of covariates can equivalently be formulated by testing if the according parameter is zero or not.

1.1.1 Parameter Estimation

For the estimation of the parameter vector $\boldsymbol{\beta}$ there are two different optimization problems commonly used, but both result in the same estimator.

- The least squares estimator is motivated by minimizing the distance of the observed response to its mean. This idea can be formulated as minimizing a specified loss function, i.e. the mean-squared-error loss functions.
- The maximum likelihood estimator (MLE) maximizes the likelihood function for given observations with respect to the model assumptions.

Both methods start with a sample of the response $\mathbf{y} = (y_1, \ldots, y_n)^t$ and the according covariates provided in form of the design matrix $\mathbf{X}$.

Given an estimator $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \ldots, \hat{\beta}_p)^t$ for the parameter vector $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_p)^t \in \mathbb{R}^{p+1}$ the estimated mean $\hat{\boldsymbol{\mu}} = (\hat{\mu}_1, \ldots, \hat{\mu}_n)^t$ of the response arise by $\hat{\boldsymbol{\mu}} = \mathbf{X}\hat{\boldsymbol{\beta}}$.

The question answered in the following sections is how to estimate the parameter vector $\boldsymbol{\beta}$ in a reasonable way.

The Least Squares Estimator

The least square estimator minimizes the Euclidean distance between $\mathbf{y}$ and $\boldsymbol{\mu}$, which is described by using the mean-squared-error loss function $V : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ defined

as

$$V(\mathbf{y}, \boldsymbol{\mu}) := ||\mathbf{y} - \boldsymbol{\mu}||^2 = \sum_{i=1}^{n} (y_i - \mu_i)^2.$$

Plugging $\boldsymbol{\mu} = \mathbf{X}\boldsymbol{\beta}$ in the loss function results in

$$V(\mathbf{y}, \boldsymbol{\mu}) = ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2 = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^t(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) = \mathbf{y}^t\mathbf{y} - 2\boldsymbol{\beta}^t\mathbf{X}^t\mathbf{y} + \boldsymbol{\beta}^t\mathbf{X}^t\mathbf{X}\boldsymbol{\beta},$$

where with some basic calculus the first derivative (gradient) has the form

$$\nabla V(\mathbf{y}, \boldsymbol{\mu}) = -2\mathbf{X}^t\mathbf{y} + 2\mathbf{X}^t\mathbf{X}\boldsymbol{\beta}.$$

Equating it to zero and assuming that $\mathbf{X}^t\mathbf{X}$ is invertible the least squares estimator for the parameter vector $\boldsymbol{\beta}$ is given by

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^t\mathbf{X})^{-1}\mathbf{X}\mathbf{y}.$$

For completeness the Hessian matrix is given by $2\mathbf{X}^t\mathbf{X}$, which is by definition (as quadratic form) positive semidefinite and since it is already assumed that $\mathbf{X}^t\mathbf{X}$ is invertible, it is indeed positive definite and $\hat{\boldsymbol{\beta}}$ is minimizing the mean-squared-error loss function.

The Maximum Likelihood Estimator

The maximum likelihood estimator (MLE) is based on the likelihood function, or equivalently the log-likelihood function, and maximizes it w.r.t. the parameter vector $(\boldsymbol{\beta}, \sigma^2)$. The independence assumption, as stated in the beginning of the chapter, allows to set the variance-covariance matrix of the multivariate normal distribution of $\mathbf{y}$ to $\sigma^2\mathbf{I}_n$ where $\mathbf{I}_n$ is the n-dimensional identity matrix. Therefore, the likelihood function of $\mathbf{y}$ can be written as

$$\begin{aligned} L(\boldsymbol{\beta}, \sigma^2|\mathbf{y}) &= \frac{1}{\sqrt{(2\pi\sigma^2)^n}} \cdot \exp\Big(-\frac{1}{2\sigma^2}(\mathbf{y} - \boldsymbol{\mu})^t(\mathbf{y} - \boldsymbol{\mu})\Big) \\ &= \frac{1}{\sqrt{(2\pi\sigma^2)^n}} \cdot \exp\Big(-\frac{1}{2\sigma^2}V(\mathbf{y}, \boldsymbol{\mu})\Big). \end{aligned} \tag{1.6}$$

By applying the logarithm to Equation (1.6) it can be observed that the log-likelihood function can be optimized in the parameter $\boldsymbol{\beta}$ and σ^2 independently, which is called optimizing the profil-log-likelihood.

$$l(\boldsymbol{\beta}, \sigma^2|\mathbf{y}) = \log(L(\boldsymbol{\beta}, \sigma^2|\mathbf{y})) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}V(\mathbf{y}, \boldsymbol{\mu}). \tag{1.7}$$

Therefore the least squares estimator and the maximum likelihood estimator coincide for the classical linear regression model.

For completeness the MLE for the variance σ^2 is derived as

$$\frac{\partial l(\hat{\boldsymbol{\beta}}, \sigma^2|\mathbf{y})}{\partial \sigma^2} = -\frac{n2\pi}{4\pi\sigma^2} + \frac{1}{2\sigma^4} V(\mathbf{y}, \hat{\boldsymbol{\mu}}) = \frac{1}{2\sigma^2}\left(\frac{V(\mathbf{y}, \hat{\boldsymbol{\mu}})}{\sigma^2} - n\right) \stackrel{!}{=} 0$$
$$\Rightarrow \hat{\sigma}^2 = \frac{1}{n} V(\mathbf{y}, \hat{\boldsymbol{\mu}}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{\mu}_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^t\hat{\boldsymbol{\beta}})^2.$$

1.1.2 The Analysis of Variance (ANOVA)

One major advantage of the normality assumption and the structure of a linear regression model is the possibility of explicitly receiving distributions for the test statistics, testing the hypothesis that parameters are equal to zero.

The most general test which can be formulated in this context is known as the F-test and is embedded in the methodology of the analysis of variance. Since this concept is well known only a short summary of the theory required for the F-test is provided in the following.

Assume a linear regression model with $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} = (\mathbf{X}_1, \mathbf{X}_2)(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2)^t + \boldsymbol{\epsilon}$, where $\boldsymbol{\beta}_1 = (\beta_0, \ldots, \beta_q)^t$ and $\boldsymbol{\beta}_2 = (\beta_{q+1}, \ldots, \beta_p)^t$. Then $\mathbf{X}_1 \in \mathbb{R}^{n\times(q+1)}$ and $\mathbf{X}_2^{n\times(p-q)}$ are the corresponding design matrices. Notice that the intercept for this model is implicitly given as a column in a design matrix which only contains ones.

The hypothesis to test has the form

$$\mathcal{H}_0 : \boldsymbol{\beta}_2 = \mathbf{0} \text{ vs } \mathcal{H}_A : \boldsymbol{\beta}_2 \neq \mathbf{0}. \tag{1.8}$$

The choice which covariates are contained in $\mathbf{X}_1$ and which in $\mathbf{X}_2$ depends on the hypothesis to test but can be chosen in generally without restrictions since p and q itself are not further restricted. Therefore a group of covariates or only a specific one can be tested with this formulation.

Also notice that the formulation $\beta_j = 0$ for a parameter j is equivalent to the statement that the covariate j has no influence on the mean of the response.

Theorem 1.1. *(F-Test)*
With the notation from above the following holds:

$$F = \frac{(||\mathbf{y} - \mathbf{X}_1\boldsymbol{\beta}_1||^2 - ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2)/(p-q)}{||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2/(n-p)} \sim \mathcal{F}_{p-q,n-p},$$

where $\mathcal{F}$ is the F-distribution. The null hypothesis (Equation (1.8)) is rejected on level α if $F > \mathcal{F}_{p-q,n-p;1-\alpha}$.

For a detailed discussion on the theory of the F-test or the analysis of variance in general consult Casella and Berger (2002).

1.2 Generalized Linear Regression Model

As shown in Section 1.1 and according to McCullagh and Nelder (1989) the classical linear regression model for a sample can be expressed by the following parts:

1. The random component: $\mathbf{y} = (y_1, \ldots, y_n)^t$ has an independent Normal distribution with constant variance σ^2 and $\mathbb{E}[\mathbf{y}] = \boldsymbol{\mu}$.
2. The systematic component: p covariates $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ with $\mathbf{x}_i = (1, x_{i1}, x_{i2}, \ldots, x_{ip})$, $i = 1, \ldots, n$, produce a linear predictor η_i given by $\eta_i = \sum_{j=0}^{p} \beta_j x_{ij}$.
3. The link function between the random and the systematic component: $g(\mu_i) = \eta_i,\ i = 1, \ldots, n$, where $g : \mathbb{R} \to \mathbb{R}$ is the identity function.

Now generalized linear models allow two extensions:

- The distribution of the random component is allowed to be a member of the linear exponential family.
- The link function can be chosen as any monotonic twice differentiable function.

For this extension and the definition of the generalized linear model the following two sections discuss the different aspects separately. Afterwards the parameter estimation is shortly mentioned, and a similar concept like the F-test is also provided.

1.2.1 Likelihood Function

At the first look the definition of the linear exponential family seems like a very specific and rather technical description of a probability density function (pdf) or a probability mass function (pmf). But some of the most important members of this class are the Normal, Poisson, Binomial, Gamma and Inverse Gaussion distributions, which cover a huge range of distributions widely used in the statistical context.

Definition 1.1. *(Linear Exponential Family)*
Assume a random variable y which pdf (or pmf) can be written as

$$f(y|\theta, \phi) = \exp\left(\frac{y\theta - b(\theta)}{a\phi} + c(y, \phi)\right),$$

where b and c are known functions and a is a known weight for the dispersion parameter ϕ. Then y has a distribution being a member of the linear exponential family (LEF).

According to McCullagh and Nelder (1989) the mean and variance of y can be derived by using the well-known properties

$$\mathbb{E}[\frac{\partial l}{\partial \theta}] = 0 \tag{1.9}$$

$$\mathbb{E}[\frac{\partial^2 l}{\partial \theta^2}] + \mathbb{E}[\frac{\partial l}{\partial \theta}]^2 = 0, \tag{1.10}$$

where $l = \log f(y|\theta, \phi)$ is the log-likelihood function. Therefore, the mean and variance of a random variable with distribution from the LEF is given as

$$\mathbb{E}[y] = b'(\theta) \tag{1.11}$$

$$Var[y] = a\phi b''(\theta). \tag{1.12}$$

Equation (1.11) provides also a good candidate for the link function. Hence the specified link for which it is assumed that $\theta = \mathbf{x}^t\boldsymbol{\beta}$ is called canonical link.

1.2.2 Link Functions

The link function connects the systematic component with the random part. Therefore, this function allows to connect the linear combination of the covariates with the parameter for the response. This makes it possible to link real valued covariates with the success probability of a binomial distribution. To illustrate this the following example of the standardized binomial distribution is provided.

Example 1.1.
For the following calculation it is assumed that m is known and $my \sim Binom(m, \pi)$. Thus the random variable y takes values in $0, \frac{1}{m}, \frac{2}{m}, \ldots, 1$ and we model a relative frequency and not the absolute frequency $(0, 1, 2, \ldots, m)$. Then the logarithm of the pmf for the standardized binomial distribution has the representation

$$\begin{aligned}
\log(f(y|m,\pi)) &= \log\binom{m}{my} + my\log(\pi) + (m - my)\log(1-\pi) \\
&= \frac{y(\log(\pi) - \log(1-\pi)) - \log(\frac{1}{1-\pi})}{m^{-1}} + c(y,\phi) \\
&= \frac{y\log\left(\frac{\pi}{1-\pi}\right) - \log(\frac{1}{1-\pi})}{m^{-1}} + c(y,\phi).
\end{aligned}$$

By defining

$$\theta = \log\left(\frac{\pi}{1-\pi}\right) \Leftrightarrow \pi = \frac{e^\theta}{1+e^\theta}$$

and recognizing that the dispersion parameter ϕ is equal one, the final form of the logarithm of the pmf is given by

$$\log(f(y|m,\theta)) = \frac{y\theta - \log(1+e^{\theta})}{m^{-1}} + \log\binom{m}{my}. \tag{1.13}$$

Therefore $a := m^{-1}, b(\theta) := \log(1+e^{\theta})$ and $c(y) := \log\binom{m}{my}$, hence the standardized binomial distribution indeed belongs to the linear exponential family.

For the canonical link the first derivative of $b(\theta)$ connects the mean of y and the systematic component θ in the following way:

$$\begin{aligned}\mu = \mathbb{E}[y] = b'(\theta) &= \frac{e^{\theta}}{1+e^{\theta}} \\ g(\mu) = b'^{-1}(\mu) &= \log\left(\frac{\mu}{1-\mu}\right) = \theta = \eta \\ \Rightarrow \mu &= \frac{e^{\eta}}{1+e^{\eta}}.\end{aligned}$$

This link function is also called logit link and has a nice interpretation in term of log odds and log odds ratio.

Example 1.1 shows that the link function projects from the real axis to the interval $[0,1]$ which yield the connection between the covariates and the response. Because the technical assumptions on the link function are quite flexible, many possible functions could serve for a given distribution.

But in practice only a few functions with nice statistical properties like the canonical link or very intuitive explanation like the probit model for the binomial case are used more frequently.

More details on this topic as well as the properties of the canonical link can be found in McCullagh and Nelder (1989).

1.2.3 Parameter Estimation

After defining a model in terms of the distribution and the link function, the next step in statistical modeling is usually the estimation of the parameters. For the class of generalized linear models, the optimization according to the least squares loss function is not possible in general and hence no analytic solution can be provided for all members of this family of models.

Due to the fact that the definition of generalized linear models relies on the assumption

of the pdf (or pmf if the scale is discrete) the maximum likelihood estimation is always a valid procedure. In this case the likelihood can be maximized at least numerically.

Because this work does not provide new insights for the well documented numerical optimization on the likelihood it should be only mentioned that the iterative weighted least squares algorithm, which is based on the Newton method for numerical optimization, is used in most cases.

1.2.4 The Analysis of Deviance

For the classical linear regression model there is a way to test individual covariates via the F-test statistic (c.p. Theorem 1.1). These results derive from the assumption of a normally distributed error term and do not longer hold for the generalized linear model framework.

Therefore, and due to a lack of exact theory only approximative results can be considered. In the following some results should provide an intuitive introduction to the deviance, but for a detailed discussion on the theory Shao (1998) should be consulted.

Definition 1.2. *(Likelihood Ratio Test, see Casella and Berger, 2002)*
Assume a sample $\mathbf{y}$ *with size* n*, then the likelihood ratio test statistic for the hypotheses*

$$\mathcal{H}_0 : \boldsymbol{\theta} \in \boldsymbol{\Theta}_0 \qquad vs \qquad \mathcal{H}_A : \boldsymbol{\theta} \in \boldsymbol{\Theta}_0^c$$

is defined as

$$\lambda_n(\mathbf{y}) = \frac{\underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}_0}{sup}\ L(\boldsymbol{\theta}|\mathbf{y})}{\underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{sup}\ L(\boldsymbol{\theta}|\mathbf{y})}.$$

A likelihood ratio test (LRT) is any test that has a rejection region of the form $\{\mathbf{y} : \lambda_n(\mathbf{y}) \leq c\}$*, where* c *is any number satisfying* $0 \leq c \leq 1$*.*

Notice that the likelihood ratio test only requires that the likelihood function exists and additionally the specification of c. For all members of the LEF the existence of a likelihood function is ensured by definition.

To provide a reasonable choice of c the asymptotic distribution of $-2\log(\lambda_n(\mathbf{y}))$ is studied in the following theorem.

Theorem 1.2. *(Asymptotic distribution of the LRT statistic, see Shao, 1998) Assume that $\boldsymbol{\Theta}_0$ is determined by*

$$\mathcal{H}_0 : \mu = g^{-1}(\mathbf{x}^t\boldsymbol{\beta}) \qquad vs \qquad \mathcal{H}_A : \text{ not } \mathcal{H}_0,$$

where $\boldsymbol{\beta}$ is a p-vector of unknown parameters and g^{-1} is the continuously twice differentiable inverse link function mapping from $\mathbb{R}^1$ to $\mathbb{R}^1$. Under the null hypothesis and some regularity conditions $-2\log(\lambda_n) \xrightarrow{d} \mathcal{X}^2_{n-p}$.

Remark 1.3.
The LRT with rejection region $\lambda_n \leq \exp\left(-\frac{\mathcal{X}^2_{n-p;1-\alpha}}{2}\right)$ has asymptotic significance level α, where $\mathcal{X}^2_{n-p;1-\alpha}$ is the $(1-\alpha)$th quantile of the chi-square distribution with $n-p$ degrees of freedom.

One important aspect of using this approximation is that the Theorem 1.2 says nothing about the convergence speed. For example, in some simulation studies it is shown that for a binomial setting with success probability close to zero or one the convergence to the chi-square distribution is indeed very pure.

The Deviance

For introducing the deviance, the following hypothesis is stated

$$\mathcal{H}_0 : \mu = g^{-1}(\mathbf{x}^t\boldsymbol{\beta}) \qquad vs \qquad \mathcal{H}_A : \mu \textit{ without restriction.} \tag{1.14}$$

Notice that this hypothesis is checking whether the specification or parameterization of the mean under the defined model is right or not. Here the correctness of the link as the set of covariates are tested simultaneously.

For the alternative hypothesis no structure is assumed and therefore the means are estimated by the observed response values. This is called the saturated model and can also occur when the number of parameters defined in the model equals the number of observations.

The deviance is now the resulting likelihood ratio test statistic for the hypothesis above and defined in the following way.

Definition 1.3. *(Deviance, see McCullagh and Nelder, 1989)*
The likelihood ratio test statistic for hypothesis (1.14) *can be written as*

$$\lambda_n(\mathbf{y}) = \frac{\sup\limits_{\boldsymbol{\mu}\in\mathcal{H}_0} L(\boldsymbol{\mu}|\mathbf{y})}{\sup\limits_{\boldsymbol{\mu}\in\mathbb{R}^n} L(\boldsymbol{\mu}|\mathbf{y})} = \frac{L(\hat{\boldsymbol{\mu}}|\mathbf{y})}{L(\mathbf{y}|\mathbf{y})} = \frac{L(g^{-1}(\mathbf{x}^t\hat{\boldsymbol{\beta}})|\mathbf{y})}{L(\mathbf{y}|\mathbf{y})}.$$

Here $\hat{\boldsymbol{\beta}}$ is the maximum likelihood estimator for $\boldsymbol{\beta}$ under the defined model. The choice $\hat{\mu}_i = y_i,\ i = 1,\dots,n$, is the saturated model and maximizes the likelihood under no constraints.

Then the scaled deviance is defined as

$$-2\log(\lambda_n(\mathbf{y})) = -2\log\left(\frac{L(\hat{\boldsymbol{\mu}}|\mathbf{y})}{L(\mathbf{y}|\mathbf{y})}\right) = -2(l(\hat{\boldsymbol{\mu}}|\mathbf{y}) - l(\mathbf{y}|\mathbf{y})) =: \frac{1}{\phi}D(\mathbf{y},\hat{\boldsymbol{\mu}}).$$

Since generalized linear models are defined by an explicit form of the pdf or pmf the deviance can be expressed in a more precise way as the following corollary shows.

Corollary 1.3. *(Deviance of the LEF)*
The scaled deviance defined in Definition 1.3 can be written in the following way:

$$\frac{1}{\phi}D(\mathbf{y},\hat{\boldsymbol{\mu}}) = -\frac{2}{\phi}\sum_{i=1}^{n}\frac{y_i(\theta(\hat{\mu}_i) - \theta(y_i)) - (b(\theta(\hat{\mu}_i)) - b(\theta(y_i)))}{a_i} = \frac{1}{\phi}\sum_{i=1}^{n} d_i.$$

Proof.
Since for all generalized linear models the log-likelihood can be written as

$$l(\theta_i,\phi|y_i) = \frac{y_i\theta_i - b(\theta_i)}{a_i\phi} + c(y_i,\phi)$$

and the assumption of independence along with the calculation below the statement of the corollary holds.

$$\begin{aligned}
\frac{1}{\phi}D(\mathbf{y},\hat{\boldsymbol{\mu}}) &= -2(l(\hat{\boldsymbol{\mu}}|\mathbf{y}) - l(\mathbf{y}|\mathbf{y})) \\
&= -2\sum_{i=1}^{n}\left(\frac{y_i\theta(\hat{\mu}_i)) - b(\theta(\hat{\mu}_i))}{a_i\phi} + c(y_i,\phi) - \frac{y_i\theta(y_i) - b(\theta(y_i))}{a_i\phi} - c(y_i,\phi)\right) \\
&= -\frac{2}{\phi}\sum_{i=1}^{n}\frac{y_i(\theta(\hat{\mu}_i) - \theta(y_i)) - (b(\theta(\hat{\mu}_i)) - b(\theta(y_i)))}{a_i}.
\end{aligned}$$

□

The Analysis of Deviance for the Generalized Linear Model

According to McCullagh and Nelder (1989) there are some problems in the generalization of the analysis of variance. The major one results from the fact that the singular sum of squares is no longer an appropriate measure of the contribution of a term to the total discrepancy, but they offer the following solution for this problem:

Given a sequence of nested models we can use the deviance as our generalized measure of discrepancy and form an analysis-of-deviance table by taking the first differences.

As McCullagh and Nelder (1989) stated the analysis-of-deviance table should be regarded as a screening device for picking out obviously important terms, and no attempt should be made to assign significance levels to the raw deviances.

Residuals for the Generalized Linear Model

For the linear regression we define raw residuals by $r_i := (y_i - \hat{\mu}_i), i = 1, \ldots, n$. Therefore we can express the dependent variable in the form

$$y_i = \hat{\mu}_i + (y_i - \hat{\mu}_i) = \hat{\mu}_i + r_i.$$

Residuals can be used to explore the adequacy of the fit of a model with respect to the goodness of fit or model assumptions and they may also indicate the presence of anomalous values, which would require further investigation.

For generalized linear models an extended definition of residuals is required, which should be applicable to all members of the LEF. McCullagh and Nelder (1989) provide several definitions of residuals which can be used. At this point only one choice which is based on the deviance will be defined now.

Definition 1.4. *(Deviance residual, see McCullagh and Nelder, 1989)*
By Corollary 1.3 we already showed that the deviance can be written as

$$D(\mathbf{y}, \hat{\boldsymbol{\mu}}) = \sum_{i=1}^{n} d_i = 2 \sum_{i=1}^{n} \frac{(b(\theta(\hat{\mu}_i)) - b(\theta(y_i))) - y_i(\theta(\hat{\mu}_i) - \theta(y_i))}{a_i}.$$

Then the deviance residuals are defined by

$$r_i := sign(y_i - \hat{\mu}_i)\sqrt{d_i} \qquad i = 1, \ldots, n.$$

This definition ensures that the sum of squares adds up to the deviance.

2 Multinomial Logistic Model

The data analyzed in this work has quiet a special shape. On one hand there are response variables like the geographical origin or the variety of the grapes, which have different and non-comparable categories. On the other hand, the data from the chemical analysis can be considered to have positive values.

Therefore, a modeling approach is needed, which allows to connect continuous covariates with categorical responses and a natural choice is given by the multinomial logistic model. In the next sections an overview of the multinomial distribution itself and its relationship to the generalized linear model framework will be presented. Afterwards the implementation in R is shortly discussed.

2.1 Multinomial Distribution

Assume that the categorical response allows for K different categories, and the aim is to model the probabilities $p_1, \ldots, p_K$ which belong to the individual categories or classes.

Definition 2.1. *(Multinomial Distribution)*
Assume a population with K different classes. By sampling m elements (with replacement) from the population the pmf of the random vector $\mathbf{y} = (y_1, \ldots, y_K)^t$, where y_k counts the number of elements from class k, $k = 1, \ldots, K$, is

$$f(\mathbf{y}|\mathbf{p}) = \binom{m}{y_1, \ldots, y_K} \prod_{k=1}^{K} p_k^{y_k} = \frac{m!}{y_1! \cdot \ldots \cdot y_K!} p_1^{y_1} \cdot \ldots \cdot p_K^{y_K}$$

for $y_k \in \mathbb{N}_0$ and $\sum_{k=1}^{K} y_k = m$, where $\mathbf{p} = (p_1, \ldots, p_K)^t$ is the parameter vector consisting of the individual class probabilities with $\mathbf{p} \in [0,1]^K$ and $\sum_{k=1}^{K} p_k = 1$.

One assumption which has to be made is to regard m known. This is required for technical reasons and does not seem to be a very restrictive assumption since it only says the number of observations is known.

Remark 2.1.
Notice that the multinomial distribution with $K = 2$ is just the binomial distribution and hence the multinomial distribution can also be motivated as generalization of the binomial distribution for more than two possible classes.

In the following the log-likelihood of a sample, with independent observations from a multinomial distribution with K different classes will be discussed. We begin with the situation where each observation $i,\ i = 1, \ldots, n$, is allowed to have its own probability vector $\mathbf{p}_i = (p_{i1}, \ldots, p_{iK})^t$.

For clearness the following notation is used:

- y_{ik} The number of elements in class k of the observation i.
- n Total number of multinomial vectors.
- n_i Number of elements in observation i.
- $\mathbf{P}$ The matrix combining all probability vectors, i.e. $\mathbf{P} = (\mathbf{p}_1, \ldots, \mathbf{p}_n)^t$.

Since the multinomial coefficients for each observation do not depend on the parameters, the relevant part of the log-likelihood function is

$$l(\mathbf{P}|\mathbf{Y}) \overset{ind.}{=} \sum_{i=1}^{n} l(\mathbf{p}_i|\mathbf{y}_i) = \sum_{i=1}^{n}\sum_{k=1}^{K} y_{ik}\log(p_{ik}) \qquad \text{with } \sum_{k=1}^{K} p_{ik} = 1. \tag{2.1}$$

If the log-likelihood should be optimized w.r.t. the parameters, the Lagrange method is used resulting in solving the following equation system:

$$\frac{\partial}{\partial p_{ik}}\left(l(\mathbf{P}|\mathbf{Y}) - \sum_{i=1}^{n}\lambda_i\left(\sum_{k=1}^{K} p_{ik} - 1\right)\right) = \frac{y_{ik}}{p_{ik}} - \lambda_i \overset{!}{=} 0, \tag{2.2}$$

$$\frac{\partial}{\partial \lambda_i}\left(l(\mathbf{P}|\mathbf{Y}) - \sum_{i=1}^{n}\lambda_i\left(\sum_{k=1}^{K} p_{ik} - 1\right)\right) = \sum_{k=1}^{K} p_{ik} - 1 \overset{!}{=} 0. \tag{2.3}$$

The first system of equations provides $\hat{p}_{ik} = y_{ik}/\hat{\lambda}_i$ and the quantity $\hat{\lambda}_i$ can be calculated by using Equation (2.3):

$$1 = \sum_{k=1}^{K}\hat{p}_{ik} = \sum_{k=1}^{K}\frac{y_{ik}}{\hat{\lambda}_i} \qquad \Rightarrow \qquad \hat{\lambda}_i = \sum_{k=1}^{K} y_{ik}.$$

Summarizing everything the maximum likelihood estimator for the individual class probability is given by

$$\hat{p}_{ik} = \frac{y_{ik}}{\sum_{k=1}^{K} y_{ik}} \qquad i = 1, \ldots, n.$$

The remaining task for the modeling is the determination of the link function, this is shown in the following section.

2.2 Multinomial Response Models

Seeing the multinomial response model again as a generalization of the binomial model a natural extension of the well-known logit link can be made and is discussed in the following.

Therefore, the binomial setting (c.p. Example 1.1) is assumed and one of the available classes is chosen as reference class with according class probability q_i. This implicates that the effect of several covariates on the class probability for the other class p_i is of interest since q_i can, in the binomial case, be expressed as $1 - p_i$.

Now the logit function is modeling the probability for the non-reference class p_i in relation to the probability of the reference class q_i for each observation by

$$\text{logit}(p_i) = \log \frac{p_i}{1 - p_i} = \frac{p_i}{q_i}.$$

Hence for the multinomial case, after specifying the reference class as the first one, each class probability can be modeled as

$$\log \frac{p_{ik}}{p_{i1}} \qquad k = 2, \ldots, K.$$

With this link function the multinomial response model can be determined by

$$\log \frac{p_{ik}}{p_{i1}} = \mathbf{x}_i^t \boldsymbol{\beta}_k = \eta_{ik}.$$

This choice allows also the comparison between two classes in one observation by the equation

$$\log \frac{p_{ik}}{p_{ij}} = \log \frac{p_{ik} p_{i1}}{p_{ij} p_{i1}} = \log \frac{p_{ik}}{p_{i1}} - \log \frac{p_{ij}}{p_{i1}} = \eta_{ik} - \eta_{ij} = \mathbf{x}_i^t \boldsymbol{\beta}_k - \mathbf{x}_i^t \boldsymbol{\beta}_j = \mathbf{x}_i^t (\boldsymbol{\beta}_k - \boldsymbol{\beta}_j). \tag{2.4}$$

As Equation (2.4) shows the parameter vector $\boldsymbol{\beta}_k,\ k = 2, \ldots, K$, is class dependent. This means that for each class in the multinomial response model, except for the reference class, there needs to be a separate parameter vector $\boldsymbol{\beta}_k$ describing the influence of an explanatory variable on the odds p_{ik}/p_{i1}.

For a more detailed discussion on this topic and also a detailed explanation of the so called Poisson trick, where the parameter of a multinomial response model is estimated by loglinear models, consult McCullagh and Nelder (1989).

Another link which is mostly used in the context of neural networks uses the softmax function for connecting the systematic and random part of the multinomial model. In this case the probabilities are directly modeled via the exponential function and standardized to ensure that $\sum_{k=1}^{K} p_{ik} = 1$ holds for $i = 1, \ldots, n$.

This means that the softmax link can be written as

$$\text{softmax}(\mathbf{p}_i) = \left(\frac{\exp(\mathbf{x}_i^t \boldsymbol{\beta}_1)}{\sum_{k=1}^{K} \exp(\mathbf{x}_i^t \boldsymbol{\beta}_k)}, \ldots, \frac{\exp(\mathbf{x}_i^t \boldsymbol{\beta}_K)}{\sum_{k=1}^{K} \exp(\mathbf{x}_i^t \boldsymbol{\beta}_k)} \right). \tag{2.5}$$

As it can be seen from (2.5) the softmax link requires one more parameter vector since the first class also gets an own parameter vector. But by using the softmax function the restriction of

$$\sum_{k=1}^{K} p_{ik} = 1, \ i = 1, \ \ldots, \ n,$$

is fulfilled by construction so it does not really allow more flexibility for the model at all.

Due to the fact that for the multinomial logistic model the logit model is implemented and far more often used, the parametrization with the softmax link is only mentioned to show that there are further options for specifying a multinomial model.

2.3 Multinomial Logistic Model in R

For the application of the multinomial logistic model there are several packages available in R. In this work the package `nnet` with the function `multinom` is used since, the parameters for the model are directly estimated via numerical optimization of the maximum likelihood function and not by the Poisson trick.

The package `nnet` was implemented by Venables and Ripley (2002) and is well integrated into the statistical modeling framework of R. For example, a fitted model can predict the response class of a new covariate by using the common `predict` function in R.

For completeness the main parts and features of the function are described in the following.

The function call is

```
multinom(formula, data, weights, subset, na.action, contrasts = NULL,
            Hess = FALSE, summ = 0, censored = FALSE, ...)
```

The detailed documentation for the function along with some examples to show the functionality can be found by typing `help("multinom")` into the R-console.

A short overview of the input parameter is provided in Table 1.

Variable	Explanation
`formula`	A formula expression as for regression models, of the form `response` $\sim$ `predictors`. The response should be a factor or a matrix with K columns, which will be interpreted as counts for each of K classes.
`data`	Optional data frame in which to find the variables occurring in formula.
`weights`	Optional observation weights in fitting.
`subset`	Expression saying which subset of the rows of the data should be used in the fit. All observations are included by default.
`na.action`	A function to filter missing data.
`contrasts`	A list of contrasts to be used for some or all of the factors appearing as variables in the model formula.
`Hess`	Logical for whether the Hessian (the observed/expected information matrix) should be returned.
`summ`	Integer; if non-zero summarize by deleting duplicated rows and adjust weights. Methods 1 and 2 differ in speed (2 uses `C`); method 3 also combines rows with the same X and different Y, which changes the baseline for the deviance.
`censored`	If Y is a matrix with K columns, interpret the entries as one for possible classes, zero for impossible classes, rather than as counts.
...	additional arguments for nnet

Table 1: Input arguments for the `multinom` function.

2.4 Multinomial Logistic Model for Classification

The main idea of this master thesis is to use the multinomial logistic model as classification algorithm. Therefore, the general concept of classification along with the definition of a classifier is provided in the following.

Remark 2.2.
*In common literature for classification problems the space of the covariates, i.e. $\mathbb{R}^p$, is often called feature space and single covariates itself named **features**. Therefore, some mixture of the wording will be used in the following chapters. Also, the response is often called **label** since it takes values in a discrete set $\mathcal{K}$ which could contain characters like names or numerical values.*

Definition 2.2. *(Classifier, see Hastie, Tibshirani, and Friedman, 2009)*
Let $\mathcal{X} \subseteq \mathbb{R}^p$ be a feature space and $\mathcal{K}$ a set of possible classes. Then every well-defined function $c : \mathcal{X} \to \mathcal{K}$ is called classifier. That is why each classifier has a corresponding partition of the feature space according to the predicted classes.

After defining the general concept of a classifier and the general theory of the multinomial logistic model (c.p. Section 2.2), the multinomial logistic classifier can be formally defined in the following way.

Definition 2.3. *(ML-Classifier)*
Assume a sample with pairs $(y_i, \mathbf{x}_i) \in \mathcal{K} \times \mathbb{R}^p$, $i = 1, \ldots, n$, and furthermore let $p_1(\mathbf{x}_i), \ldots, p_K(\mathbf{x}_i)$ be the individual class probabilities model by a multinomial logistic model with covariates $\mathbf{x}_i \in \mathbb{R}^p$. Then the multinomial logistic classifier $c : \mathbb{R}^p \to \mathcal{K}$ is defined by

$$c(\mathbf{x}_i) := \underset{k=1,\ldots,K}{\operatorname{argmax}}\ p_k(\mathbf{x}_i).$$

If the class probabilities are estimated by $\hat{p}_1(\mathbf{x}_i), \ldots, \hat{p}_K(\mathbf{x}_i)$, then the estimated (trained) classifier is given as

$$\hat{c}(\mathbf{x}_i) = \underset{k=1,\ldots,K}{\operatorname{argmax}}\ \hat{p}_k(\mathbf{x}_i).$$

As Definition 2.2 shows, the focus lies on the partitioning of the feature space into subregions (not necessary subspaces). Therefore, a characterization of the different classifiers according to this partition seems reasonable.

2.4.1 Linear Methods for Classification

Since every partition can be described by the boundaries between the individual subregions Hastie, Tibshirani, and Friedman (2009) mentioned the categorization of the classification algorithms, and the resulting classifier, according to the shape of the decision boundaries.

Definition 2.4. *(Linear Classifier, see Hastie, Tibshirani, and Friedman, 2009) Suppose there are K classes in a classification problem and a classifier $c : \mathbb{R}^p \to \mathcal{K}$. Then the decision boundary between class i and j is defined by*

$$B_{ij} := \{\mathbf{x} \in \mathbb{R}^p : i = c(\mathbf{x}) = j\} = B_{ji}.$$

The set of decision boundaries is then defined as $B := \bigcup_{i \neq j} B_{ij}$. A classifier is called linear classifier or linear method for classification, if and only if the decision boundaries are a finite set of hyperplanes or affine sets.

Some linear classifiers are provided by using linear discriminant analysis (LDA), performing threshold regression for classification or as Theorem 2.1 will show the ML - classifier containing the simple case $K = 2$, the logistic regression, as well.

2.4.2 Multinomial Logistic Model as Linear Classifier

Theorem 2.1.
The multinomial logistic classifier as defined in Definition 2.3 is a linear classifier in terms of Definition 2.4.

Proof.
Assume a general $\mathbf{x} \in \mathbb{R}^p$ which defines the bound between two classes $k_1, k_2 \in \mathcal{K}$. Since $\mathbf{x}$ is on the decision bound it holds that

$$k_1 = \underset{k=1,\dots,K}{\operatorname{argmax}}\ \hat{p}_k(\mathbf{x}) = c(\mathbf{x}) = \underset{k=1,\dots,K}{\operatorname{argmax}}\ \hat{p}_k(\mathbf{x}) = k_2.$$

Therefore $\hat{p}_{k_1}(\mathbf{x}) = \hat{p}_{k_2}(\mathbf{x})$ must hold for the assumed $\mathbf{x}$. By using Equation (2.4) without loss of generality the following holds:

$$\mathbf{x}^t(\hat{\boldsymbol{\beta}}_{k_1} - \hat{\boldsymbol{\beta}}_{k_2}) = \log\left(\frac{\hat{p}_{k_1}(\mathbf{x})}{\hat{p}_{k_2}(\mathbf{x})}\right) = \log(1) = 0.$$

Using the last equation, the set of decision boundaries can be characterized in the following way:

$$\{\mathbf{x} \in \mathbb{R}^p : \mathbf{x}^t(\hat{\boldsymbol{\beta}}_i - \hat{\boldsymbol{\beta}}_j) = 0,\ i \neq j\}.$$

With this the decision region can be described through hyperplanes and therefore the multinomial logistic classifier is a linear classifier in the sense of Definition 2.4.

□

3 Feature Selection for Classification Problems

Due to the use of high dimensional data in this work ($\approx$ 2.000 features) and also a very small sample size ($\approx$ 100 measurements) this chapter explains the problems which accompany this setting. Afterwards some ideas of feature selection with a possible categorization according to the different approaches is provided.

The topics covered in this chapter are based on the paper by Tang, Alelyani, and Liu (2014) which provides an overview of the different methods of feature selection also mentioning the different types of data.

3.1 Feature Selection and Feature Evaluation

In statistical modeling there are usually two questions considered:

1. Which covariates or features have significant influence on the response variable?
2. In which way do the significant covariates influence the distribution of the response variable?

The first question is often called the problem of feature selection, whereas the second one refers to parameter estimation and model interpretation. Due to the fact that parameter estimation is essential for all kind of modeling topics, for most of the new developed models this problem is usually solved first. Therefore at least a numerical solution can be provided for more or less all types of models.

The topic of feature selection is a more complex one because a lot of different factors like the model itself or the type of data affect it. For example, in big data problems the elimination of redundant information is quite important but it needs to be handled with care to differentiate between similar and redundant information.

Since a lot of optimization problems can only be solved numerically, a lot of estimators do not have a closed form. Thus the development of adequate theory is very difficult and hence restricted to the usage of approximation and heuristic results.

Especially in the high dimensional case, which will be discussed in the following, the

topic of feature selection also faces the limitation of computational power to answer questions in reasonable time.

Apart from the two questions considered above there are also two different types of available data if the number of observations n and the number of available features is concerned:

- Classical problem: p is much smaller than n,
- High dimensional case: $p >> n$.

In the classical setting ($n >> p$) a lot of theory is available since the setting allows good estimation and testing for multivariate samples.

In the high dimensional case ($p >> n$) several problems occur. One of them refers to the fact that most multivariate techniques for estimation and testing require much more observations than available, because the underlying models need to fit more parameters than the number of observations allows. This results in a lack of exact theory and again more heuristic approaches, which often include restrictions and cause bias in application.

Another problem faces the required computational resources which usually increase with the number of possible parameters and do not allow to check all possible combinations of features. For example, if 5 covariates are available and all models with two explanatory variables are considered, estimated and compared only 10 different models need to be calculated. Now if 10 covariates are available 45 models need to be estimated and compared which shows that the computational time does not increase linearly in the number of features. Thus, only restricted searches deliver the semi-optimal model, which is the optimal model under restrictions.

The last problem to mention is given by the concept of overfitting. This means that a model is maybe very complex in terms of the used parameters but also very good w.r.t. the explanation of the observed data. But the intention of statistical learning is to extract valid information from the observed data, and afterwards use this knowledge to predict scenarios that are not observed like different experimental settings. By using a very complex model the data gets reflected more or less by itself which makes the idea of statistical learning obsolete. This problem, often described by the out-of-sample prediction accuracy, is very difficult to detect because cross-validation or out-of-sample studies are usually used. But both techniques reduce the number of available observations in the model fitting step and therefore the ratio gets even worse.

In the following sections some strategies for feature selection, especially in the high dimensional case, are discussed in a more theoretical way and categorization is provided.

3.2 Feature Selection Algorithms

Before starting with the different types of feature selection algorithms some basic considerations of the data are provided. Afterwards the three major feature selection methods for the data used in the practical part are explained in more detail.

3.2.1 Data Types for Feature Selection

Since this work is dealing with classification problems in the practical part, this chapter is restricted to feature selection for these kind of problems. Therefore it is assumed that the response variable is taking values in a set $\mathcal{K} = \{k_1, \ldots, k_K\}$ if not stated otherwise. According to Tang, Alelyani, and Liu (2014) features can be divided into the following three different categories.

For features of the first category, also called flat features, it is assumed that the features are independent or at least the dependency is negligible. This will also be the case for the features which are considered in the practical part of this work.

The second category consists of features which have a certain structure, e.g., spatial or temporal smoothness or disjoint/overlapping groups, sometimes the structure can also be described by certain trees or graphs. They are called structured features and incorporating knowledge about the structures of features may significantly improve the classification performance and help to identify the important features for the classification problem. An example of tree structured features is given by the image pixels of the face image which can be represented as a tree, where each parent node contains a series of child nodes that enjoy spatial locality.

For the last one we relax the implicit assumption that all features are known in advance. Therefore, we obtain a scenario where candidate features are sequentially presented to the classifier for potential inclusion in the model. In this scenario, the candidate features are generated dynamically, and the size of features is unknown. We call this kind of features streaming features, and a famous example is the microblogging website Twitter which produce more than 250 million tweets per day and many new words (features) are generated such as abbreviations. When performing feature selection for tweets, it is not practical to wait until all features have been generated, thus it could be more preferable to consider streaming feature selection.

The examples above show that the right categorization of the available features is important for the usage of appropriate feature selection algorithms. As mentioned before the data from the experimental setting we are dealing with in the practical part are considered to be flat features, therefore only feature selection algorithms dealing with flat features are discussed further.

3.2.2 General Framework of Feature Selection

Before getting into detail on the different feature selection methods, the general concept of feature selection for classification problems will be described in a more abstract way. This allows to divide the problem of feature selection into different subgroups and develops further understanding of the underlying mechanisms.

Starting with a training set containing a sample of size n with q 'raw' features the first step is to generate the final features. This means that the raw features are transformed, i.e. via standardization or transformation (PCA), into a set of p different features which are specifically used in the feature selection and classification problem in general.

An example for this generation of new features is the so-called kernel trick for support vector machines. Here the idea is, at least theoretically, to map non-linear separable data into a higher dimensional space where it is possible to find a hyperplane that separates the data points. Since this is not a good idea for the small sample size setting, we focus more on feature standardization or just avoid this step and work with the original data instead.

After preparing the features they are combined with the label information, i.e. the response vector, and the feature selection is performed. Since we discuss the feature selection later in more detail at this point it should be noted that for the feature selection it is also possible to interact with the learning algorithm (or classification algorithm).

At the end of the feature selection a final collection of features is the result and via the learning algorithm (i.e. fitting the parameters) we obtain a final model or more precise classifier from the described algorithm.

Notice also that the described framework of feature selections is applicable to every classifier (here called learning algorithm) and to every type of feature discussed before.

Tang, Alelyani, and Liu (2014) provide a general framework of feature selection for classification problem which is visualized in Figure 1.

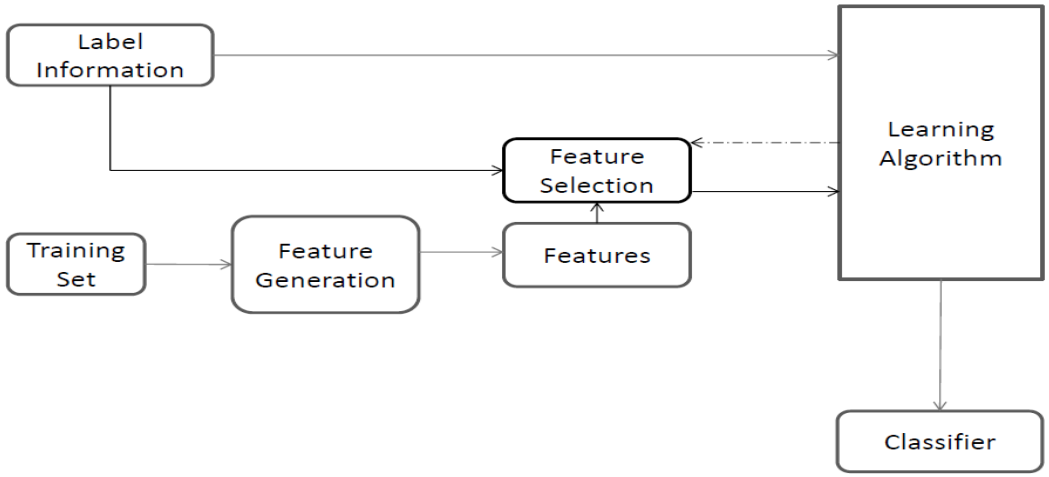

Figure 1: A general framework of feature selection from Tang, Alelyani, and Liu (2014).

If we only consider algorithms for flat features now, they then can be categorized as

- Filter Models,
- Wrapper Models and
- Embedded Models.

The methodology of each of them will be discussed in the following and the application along with the required theory will be provided in an individual chapter.

3.3 Filter Models

A quite simple method for filter selection is provided by filter models. Here each feature is evaluated by their relationship to the response and then a certain number of the best, according to the relation with the response, features are used in the final classifier.

Therefore a filter model requires the following three ingredients to be specified:

- A measure $\rho : \mathcal{K}^n \times \mathbb{R}^n \to [0, \infty)$ for the relationship between a feature and the response.
- An additional hyperparameter $m \in \mathbb{N}$, which determines the number of features in the final classifier.
- A classifier $c : \mathbb{R}^m \to \mathcal{K}$, which only uses the best m features for training, i.e. fitting the corresponding parameters.

If the relationship measure evaluates the feature only by using itself and no other features, then the filter model is called univariate and otherwise multivariate. Since the multivariate scheme evaluates a bunch of features at a time it is capable to handle redundant features.

One big advantage of the (univariate) filter models is provided by the simplicity and the corresponding run time. This results from the fact that each feature needs to be evaluated only once.

The drawback of filter models in general is the additional tuning parameter m which needs to be chosen or estimated in a proper way. For example, methods like cross validation or other methods which estimate the out-of-sample accuracy could be used to optimize the tuning parameter, but this would require additional computational resources.

On the other hand, if the in-sample classification accuracy is used to evaluate different values for m this could lead to overfitting, since with an increasing number of parameters the in-sample classification accuracy will increase as well.

Because the choice of the tuning parameter is not that simple, Chapter 6 will only focus on the behavior of different filters and visualize the results by using the principal component analysis. A combination of filter and wrapper model will be introduced in Chapter 7 and deals with the choice of the hyperparameter m.

3.4 Wrapper Models

If we use the interaction of the feature selection and the learning algorithm it is possible to repeatedly choose a set of features, train or fit the learning algorithm and evaluate the resulting classifier until some kind of termination condition is fulfilled. This method is called wrapper models and because this concept is quite general a formal definition would be not adequate at this point.

Nevertheless, each wrapper model consists of the following parts:

- A feature search mechanism, which allows to choose a set of features in a reasonable way.
- A feature evaluation criterion, which uses a classifier and evaluates the performance of the same when the chosen features are used.
- A classifier, since this concept is not restricted to the usage of the final classifier algorithm in the evaluation step. But it is very unusual to use a different classifier for the feature selection and the learning algorithm.

To provide some kind of definition the following Pseudo-Code reflects the mechanism of a wrapper model, where the training set is a data matrix (design matrix) $\mathbf{X} \in \mathbb{R}^{n \times p}$ and a classifier $c : \mathbb{R}^j \to \mathcal{K}, j \in \{1, \ldots, p\}$.

Algorithm 1: Feature Selection - Wrapper Model

while *termination criteria is not fulfilled* **do**
  select features $(i_1, \ldots, i_j)$ where $j \in \{0, 1, \ldots, p\}$
  fit classifier $\hat{c} \leftarrow c(x_{i_1}, \ldots, x_{i_j})$
  evaluate classifier $\hat{c}$

Before taking a closer look at a special type of feature search method, there are two aspects of wrapper models which should be mentioned here.

The first one is given by the required computational power, because for every set the feature search method chooses, the parameter for the corresponding classifier needs to be fitted and afterwards the results need to be evaluated. Due to the fact that fitting the parameter for the classifier is in most cases done by numerical optimization for complex classifier and a higher number of features this type of feature selection has a long run time.

The second aspect is the problem of overfitting, because when the same classifier is used for feature selection and for the final model this could induce overfitting if the performance measure is not taking care of the model complexity or overfitting in general.

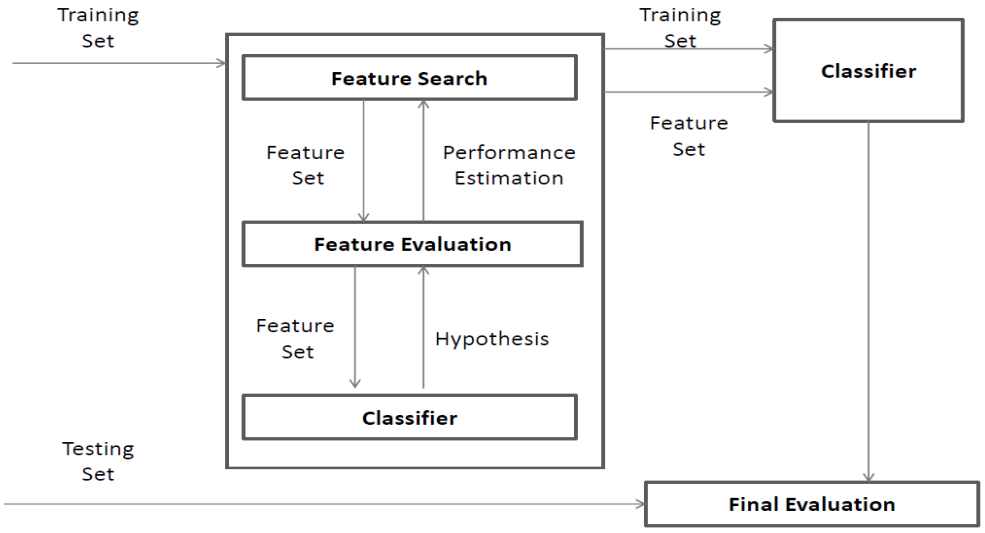

Figure 2: A general framework of wrapper models from Tang, Alelyani, and Liu (2014).

Before methods for the searching of features are discussed we take a closer look at the concepts of feature evaluation, because the feature search methods considered in this work are based on feature evaluation methods.

3.4.1 Information Based Feature Evaluation

A reasonable feature evaluation has to compare the goodness of fit of a model and the model complexity. While it is obvious that the goodness of fit should be as high as possible, the model complexity should be as low as reasonable. This comes from the fact that we assume for a very complex model, i.e. representing the data by itself, that the generalization error is much larger than for a simpler model. The generalization error is defined as the chance that a new sample $\mathbf{x} \in \mathbb{R}^m$ is classified wrong.

An established way to handle this interplay between model complexity and goodness of fit is provided by information criteria. Here there is a term measuring the goodness of fit and a term penalizing the model complexity. This setup allows to compare different models, i.e. with different complexity, when the classifier is in general of the same type, i.e. a multinomial logistic classifier. If we now identify the models with the contained features we can compare a set of features according to the performance of the classifier.

For the following information criteria the goodness of fit is always measured by the value of the log likelihood function at the maximum likelihood estimator, i.e. $l(\hat{\boldsymbol{\beta}}|\mathbf{y}, \mathbf{X})$. This allows comparing the multinomial logistic classifier based on the underlying multinomial distribution. Therefore, we are not comparing the predicted classes of the classifier but the underlying probability that a sample is in the corresponding class. For samples close to the decision boundary this allows us to compare them with more nuances, i.e. we have rather a 'continuous' comparison than an 'discrete' one.

In the following it is assumed that there is a likelihood function $L(\boldsymbol{\beta}|\mathbf{y}, \mathbf{X})$, depending on the parameter vector for the covariates $\boldsymbol{\beta} \in \mathbb{R}^p$.

Definition 3.1. *(Akaike Information Criteria)*
The information criteria introduced by Akaike is defined as

$$AIC(\hat{\boldsymbol{\beta}}) = -2 \log\ L(\hat{\boldsymbol{\beta}}|\mathbf{y}, \mathbf{X})\ +\ 2p,$$

where $\hat{\boldsymbol{\beta}}$ is the maximum likelihood estimator. Notice that p is the total number of parameters which are estimated so if $l-1$ covariates are available and a intercept is used along with a parameter for the volatility (σ^2 in the classical regression model) then p is replaced by $l+1$. This information criteria is based on estimating the difference between the 'true' probability and the probability estimated with the model using the parameter $\hat{\boldsymbol{\beta}}$ using the Kullback-Leibler-divergence.

Definition 3.2. *(Akaike Information Criteria Corrected)*
In practice it can be observed that for small sample size problems the AIC tend to overfit, therefore a corrected version of it is introduced by

$$AIC_c(\hat{\boldsymbol{\beta}}) = -2\log\ L(\hat{\boldsymbol{\beta}}|\mathbf{y},\mathbf{X})\ +\ 2p\ +\ 2\frac{(p+1)(p+2)}{n-p},$$

where again $\hat{\boldsymbol{\beta}}$ is the maximum likelihood estimator.

Remark 3.1.
Notice that by

$$AIC_c(\hat{\boldsymbol{\beta}}) = AIC(\hat{\boldsymbol{\beta}})\ +\ 2\frac{(p+1)(p+2)}{n-p},$$

for $n \to 0$ the additional penalization term vanishes and both criteria asymptotically coincide.

Another approach which takes care of the sample size is the Bayesian information criteria.

Definition 3.3. *(Bayesian Information Criteria)*
The Bayesian information criteria is introduced as follows:

$$BIC(\hat{\boldsymbol{\beta}}) = -2\log\ L(\hat{\boldsymbol{\beta}}|\mathbf{y},\mathbf{X}) + \log(n)p,$$

where $\hat{\boldsymbol{\beta}}$ is the maximum likelihood estimator.

Remark 3.2.
Notice that for $n = e^2 \approx 7.389$ both criteria coincide and for sample sizes larger or equal 8 the penalization term in the Bayesian information criteria is greater. Therefore, in the case of $n \geq 8$ the Bayesian information criteria favors sparser models compared to the Akaike information criteria.

After defining some criteria for the feature evaluation, the next subsection deals with the feature search.

3.4.2 Feature Search

Choosing and comparing all combinations of available features is not a practicable method for most data situations because the number of model growths exponentially with the number of features and the size of the model. Therefore, two approaches for the feature search, the forward and the backward selection, are discussed in the following. Both are restricted to a certain search path which is the reason why only semi optimal models are possible.

Forward selection starts by including one feature into the model, which can be done by educated guess or by taking the feature which minimizes the information criteria when all features are used individually.

Afterwards all remaining features are included in the model, again individually, and therefore the feature which minimizes the information criteria over this set of models is included in the model.

This procedure is repeated until an inclusion does not reduce the information criteria, then the algorithm stops, and the current model is selected.

Backward elimination goes the other way round. Instead of starting with the smallest model possible, the backward elimination uses as much information as possible.

Each feature is dropped out and the according information criteria is calculated. If there are one or more features which decrease the information criteria by leaving them out of the model, the feature which decrease the information criteria most by leaving it out is finally dropped.

If the information criteria does not decrease at all, the algorithm stops and the remaining features build up the final model.

Comparison of Different Strategies

While the forward selection method may miss good models because at some point during the search no improvement can be observed, the backward elimination favors too complex models.

This fact should be kept in mind when considering these strategies. When the number of features is in a practicable range both methods can be applied, and the resulting models can be compared based on the information criteria used in the search steps.

Remark 3.3.
In the case of $p > n$ a backward selection is not possible since the model with the maximal amount of information would include more parameter then available observations. Therefore, it is not possible to estimate the parameter and evaluate the information criteria so in the case of $p > n$ and as a consequence in the high dimensional case ($p >> n$) only a forward selection is practicable.

3.5 Embedded Models

While the wrapper models have an iterative structure consisting of a searching step (select features) and an optimization step (fit the classifier), the embedded models combines both steps to optimize and select appropriate features simulations in one step.

The main idea of these models is to modify the optimization problem to embed the feature selection. Due to the fact that this is in general a one-step method, i.e. we only need to solve one optimization problem. The embedded models have the advantage that their required computational power is in general lower than for wrapper models.

For completeness the method described above is called regularization method and is only one of several approaches which are named embedded models but since this work focuses on the regularization methods we will not distinguish between the embedded models and this special type of embedded models.

Before we can introduce a formal definition of regularization methods there is one aspect of the multinomial logistic classifier worth to mention at this point. Up to now we always assumed that p is the number of parameter and the number of features, we implicitly assumed a 1-1 relation between these two quantities. But for the multinomial logistic classifier this relation does no longer hold.

For a design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ the parameter vector for the multinomial logistic model with K classes is given by $\boldsymbol{\beta} = (\boldsymbol{\beta}_2^t, \ldots, \boldsymbol{\beta}_K^t)^t$, where $\boldsymbol{\beta}_k = (\beta_{0k}, \ldots, \beta_{pk})^t$ (c.p. Chapter 2). Therefore the complexity for the multinomial logistic classifier fitted on data $\mathbf{X} \in \mathbb{R}^{n \times p}$ is $q := (K-1) \cdot p$, and we will use this notation in the following section.

Definition 3.4. *(Regularization Methods, see Tang, Alelyani, and Liu, 2014)*
Consider a linear classifier only depending on the parameter $\boldsymbol{\beta} \in \mathbb{R}^q$, and a design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ containing all available features. Furthermore let $V : \mathbb{R}^n \times \mathbb{R}^q \times \mathbb{R}^{n \times p} \to \mathbb{R}$ be a loss function depending on the response vector $\mathbf{y}$, the parameter vector $\boldsymbol{\beta}$ and the design matrix $\mathbf{X}$. Then the regularization method is provided by the optimization problem

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^q}{\operatorname{argmin}} \; V(\mathbf{y}, \boldsymbol{\beta}, \mathbf{X}) + \alpha \times \mathit{penalty}(\boldsymbol{\beta}), \tag{3.1}$$

where $\alpha \in [0, \infty)$ is called the regularization parameter for the penalization function denoted by $\mathit{penalty} : \mathbb{R}^q \to [0, \infty]$.

Remark 3.4.
Note that the usage of a loss function at Definition 3.4 coincides with the definition

used for the description of the least squares estimator in Chapter 1. If we fix the design matrix the loss function only depends on the response vector and the parameter vector. At this point this definition should only point out the dependence on the input data.

By Equation (3.4) we can observe the interplay between the goodness of fit and the penalization for model complexity, as we have already discussed for the information criteria. Therefore the negative log-likelihood function, without a constant, will be used as loss function in the following, i.e. $V(\mathbf{y}, \boldsymbol{\beta}, \mathbf{X}) := -\log L(\boldsymbol{\beta}|\mathbf{y}, \mathbf{X})$. With this choice for the loss function we get by $\alpha = 0$ the maximum likelihood estimator $\hat{\boldsymbol{\beta}}$ for the parameter vector $\boldsymbol{\beta}$ one could use well established theory with at least asymptotic properties.

For $\alpha \neq 0$ we introduce a bias for the corresponding estimator, which shows the major drawback of regularization methods. By modifying the optimization problem, we take into account that the resulting estimators are biased. Therefore the choice of the regularization parameter and the type of penalization function is very important because these two elements allow us to 'control' the amount of bias we are willing to pay for the benefits which come along with these methods.

Up to this point we have no assumptions or restrictions on the penalization functions which is why there could be a wide range of possible choices for them. But there are some commonly used penalization functions, which have the advantage that corresponding optimization methods are already developed and implemented in R. Some examples of these penalization functions are

- lasso penalty $penalty(\boldsymbol{\beta}) = \sum_{i=1}^{q} |\beta_i| = ||\boldsymbol{\beta}||_1$,
- ridge penalty $penalty(\boldsymbol{\beta}) = \frac{1}{2} \sum_{i=1}^{q} \beta_i^2 = \frac{1}{2}||\boldsymbol{\beta}||_2^2$,
- elastic net penalty $penalty(\boldsymbol{\beta}) = \lambda \sum_{i=1}^{q} |\beta_i| + \frac{1-\lambda}{2}\big(\sum_{i=1}^{q} \beta_i^2\big)$ with $\lambda \in [0, 1]$.

Notice that all examples of penalization functions above are based on a norm or are a convex combination of norm-based functions. By the following lemma we will see that this setting ensures that the penalization functions are convex, which is a requirement for a lot of optimization methods to find a global minimum.

The statistical impact of these functions is discussed and visualized by an example in the following subsection.

Lemma 3.1.
Assume $f_i : U \to \mathbb{R}$ for $i = 1, \ldots, n$ are convex functions mapping from a convex set $U \subseteq \mathbb{R}^q$ to the real numbers. Then the function $f : U \to \mathbb{R}$ defined by the convex combination

$$f(\mathbf{x}) := \sum_{i=1}^{n} \lambda_i f_i(\mathbf{x}), \qquad \lambda_i \geq 0 \text{ and } \sum_{i=1}^{n} \lambda_i = 1,$$

is convex. Furthermore if there exists an $\lambda_i > 0$ and the f_i are strictly convex then so is f.

Proof.
Let $\mathbf{x}_1, \mathbf{x}_2 \in U$ and $t \in [0,1]$ then by the convexity of f_i, i.e.

$$f_i(t\mathbf{x}_1 + (1-t)\mathbf{x}_2) \leq tf_i(\mathbf{x}_1) + (1-t)f_i(\mathbf{x}_2),$$

it holds that for $\lambda_i \geq 0$

$$\sum_{i=1}^{n} \lambda_i f_i(t\mathbf{x}_1 + (1-t)\mathbf{x}_2) \leq t \sum_{i=1}^{n} \lambda_i f_i(\mathbf{x}_1) + (1-t) \sum_{i=1}^{n} \lambda_i f_i(\mathbf{x}_2),$$

which implies

$$f(t\mathbf{x}_1 + (1-t)\mathbf{x}_2) \leq tf(\mathbf{x}_1) + (1-t)f(\mathbf{x}_2).$$

Under the additional assumptions the same argumentation leads to the additional statement. □

Lemma 3.2.
Every norm $||\cdot|| : \mathbb{R}^q \to \mathbb{R}$ is a convex function.

Proof.
By the triangle inequality and the homogeneity of the norm if follows for $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^q$ and $t \in [0,1]$

$$||t\mathbf{x}_1 + (1-t)\mathbf{x}_2|| \leq ||t\mathbf{x}_1|| + ||(1-t)\mathbf{x}_2|| = t||\mathbf{x}_1|| + (1-t)||\mathbf{x}_2|||.$$

□

Along with the well-known fact that the composition of convex functions is again a convex function, which can be proven by just using two times the definition of convexity, and the fact that the function $f(x) = x^2$ is convex we have shown that all provided penalty functions are convex.

According to Tang, Alelyani, and Liu (2014) embedded models and embedding feature selection with classifier construction, have the advantages

(1) of wrapper models - they include the interaction with the classifier model and

(2) of filter models - they are far less computationally intensive than wrapper methods.

3.5.1 Different Types of Penalization for Embedded Models

For embedded models the choice of the right penalization is crucial, therefore we will take a closer look at the underlying norms and also regard an example to illustrate the influence on the optimization problem. To visualize the differences between the L2-norm (ridge penalization) and the L1-norm (lasso penalization) and to see how the elastic net is related to both of them Figure 3 shows the unit circle, i.e. $\{\mathbf{x} \in \mathbb{R}^2 : ||\mathbf{x}|| = 1\}$, with respect to the different norms $||\cdot||$.

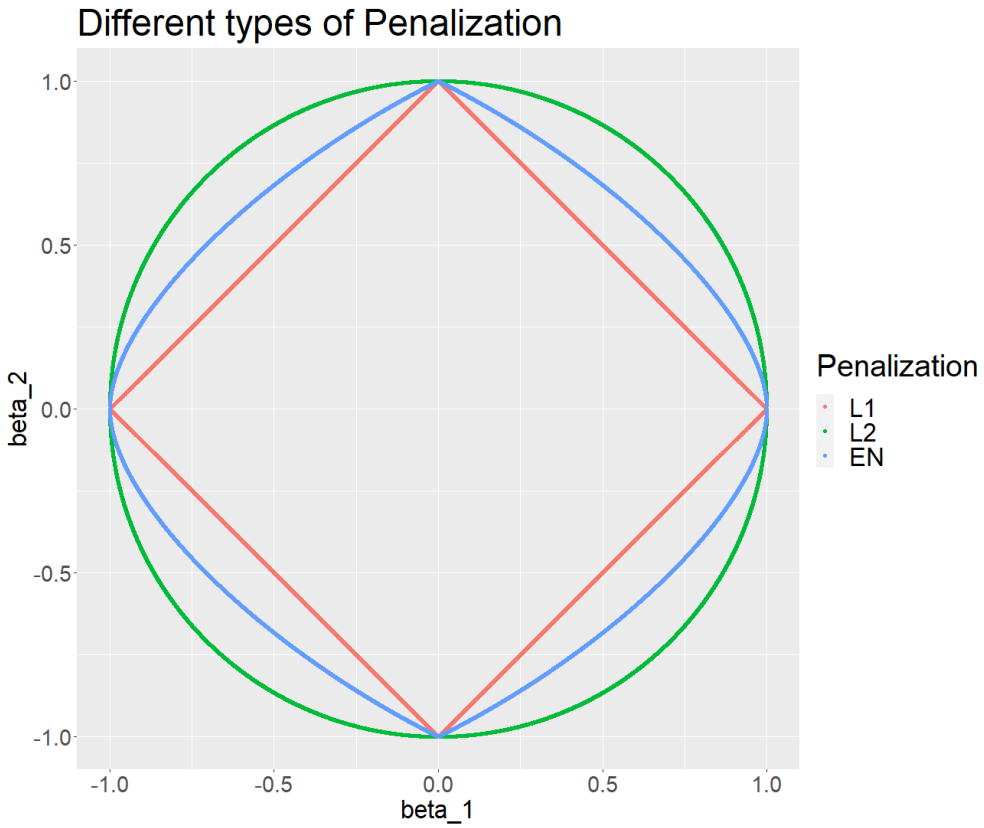

Figure 3: Unit circle w.r.t different metrics and EN with $\lambda = \frac{1}{2}$.

Assuming that we have a two dimensional parameter vector, i.e. $\boldsymbol{\beta} := (\beta_1, \beta_2) \in \mathbb{R}^2$ and w.l.o.g. we can assume that $||\boldsymbol{\beta}|| = 1$ for each norm $||\cdot||$. If we identify the horizontal axis with β_1 and the vertical axis with β_2 every point on the individual circles represent a possible choice for the parameter vector we are considering.

As an example to see that the L1-norm favors sparse models consider the two parameter vectors $\boldsymbol{\beta}_1 = (1, 0)$ and $\boldsymbol{\beta}_2 = (2^{-\frac{1}{2}}, 2^{-\frac{1}{2}})$. Both are on the unit circle w.r.t the L2-norm, but while $\boldsymbol{\beta}_1$ is also on the unit circle w.r.t. the L1-norm, it holds that $||\boldsymbol{\beta}_2||_2 \approx 1.41$ is clearly not on the unit circle w.r.t. the L1-norm. This makes it clear

that when we optimize with a lasso penalization the corresponding estimator is more likely to be sparse then the estimator which corresponds to the L2-norm.

The elastic net penalization now is a hybrid, here the additional tuning parameter λ allows us to control the amount of sparsity we want for the estimator.

After providing some first insights for the usage of different penalization functions, the following example shows that the regularization parameter α is capable to control the bias we are introducing with the penalization. Therefore, we consider simulated data and only use an intercept as model parameter. To keep this example simple the data were simulated from a normal distribution.

Example 3.1.
Before going into the details of penalization, first verify that the 1,000 simulated data points from a normal distribution (i.e. $\mathcal{N}(5,1)$) truly have the right distribution and structure. For completeness the seed to simulate this data is equal to 7. Figure 4 compares the empirical density of the data points, visualized by the normalized histogram, to the theoretical density of the normal distribution (the red curve) with mean 5 and variance 1.

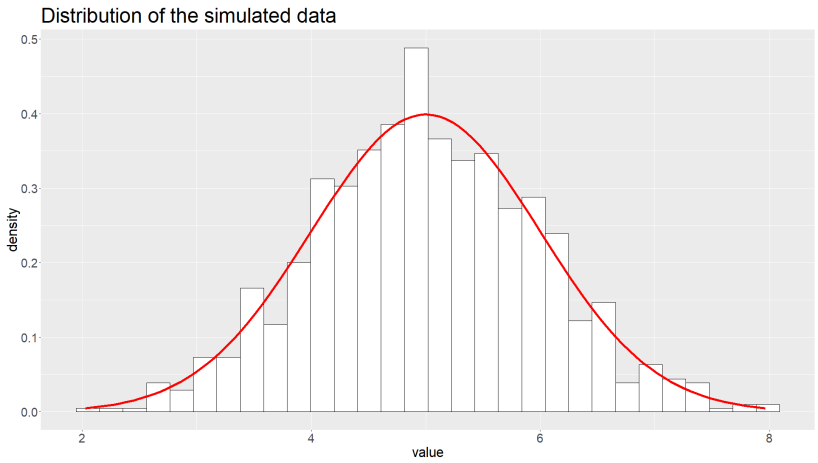

Figure 4: Histogram of the simulated data with theoretical density ($\mu = 5,\ \sigma^2 = 1$).

For simplicity it is assumed that we know the true variance ($\sigma^2 = 1$) in the following, so only the parameter μ needs to be estimated. To get the problem in the shape of Definition 3.4 we define the design 'matrix' $\mathbf{X} = (1, \ldots, 1)^t$ as column vector only consisting of ones and get for the model parameter $\beta = \mu$ when using the linear model.

Therefore, the loss function can be written as

$$\begin{aligned} V(\mathbf{y}, \beta, \mathbf{X}) &= -l(\beta, \sigma^2 = 1|\mathbf{y}) + \alpha \times penalty(\beta) \\ &= -\frac{n}{2}\log(2\pi) - \frac{1}{2}\sum_{i=1}^{1000}(y_i - \beta)^2 + \alpha \times penalty(\beta). \end{aligned}$$

For developing further understanding of the penalization terms, in the next step we optimize not via an analytic solution but with the brute force method. That way we calculate the loss function for a fine grid of values and take the value which minimizes the loss function.

With this setting two different aspects are considered in the following figures.

- *The first one (Figure 5) shows the comparison of the loss function against the different values of β for different penalization terms. Here the regularization parameter alpha was set to $\alpha = 100$ to see some differences in the figure. The argument which minimized the corresponding optimization problem is marked by a point.*
- *The second one (Figure 6, 7 and 8) shows how each penalization function is affected by a change of the regularization parameter.*

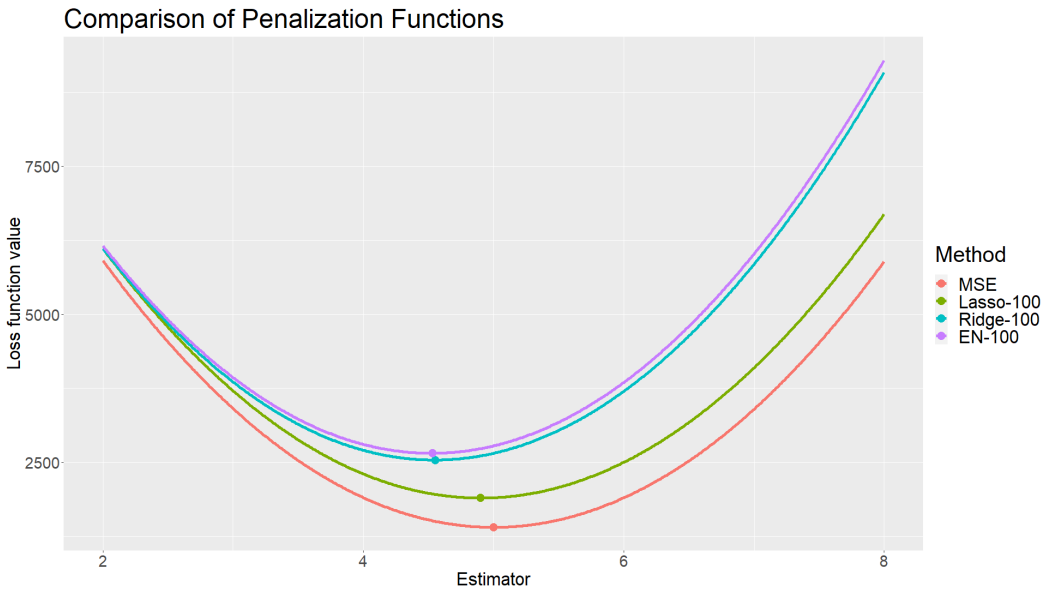

Figure 5: Comparison of different penalization functions.

At Figure 5 we can observe that by using a penalty on the parameter vector a bias is induced. At a closer look we can see that the lasso penalization has the smallest bias

in this example, this is caused by the sample since true value for the parameter μ and also the values drawn in the figure are greater than one.

For the elastic net penalization, the value of the mixture parameter was set to $\lambda = \frac{1}{2}$ to see the in between character of this penalization function.

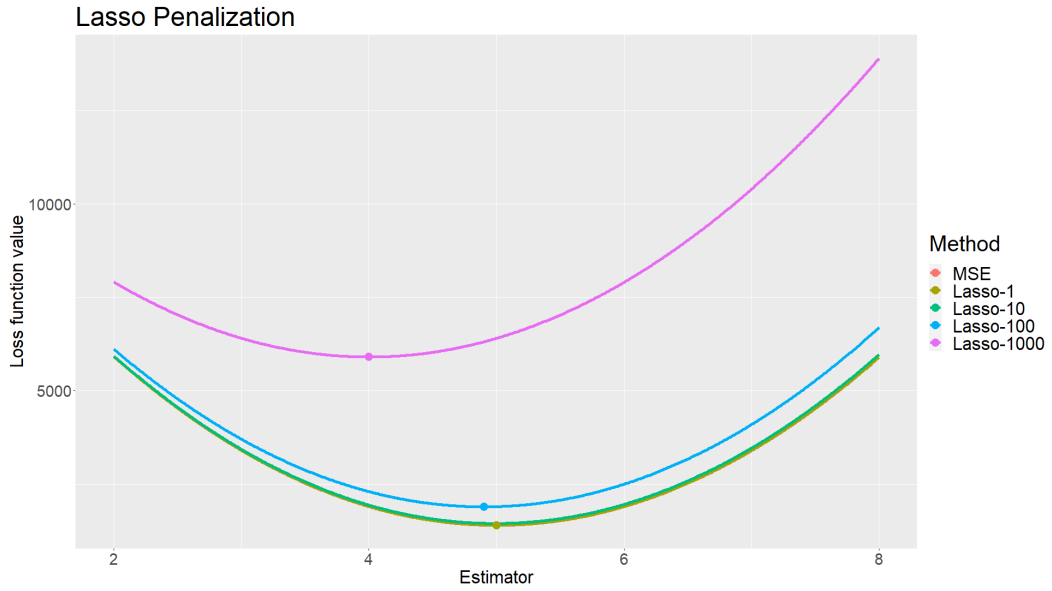

Figure 6: Regularization parameter α on the lasso penalization.

Figure 6 shows the impact of the regularization parameter on the estimation when the lasso penalization is used. Note that up to a value of 100 the introduced bias is very low compared to the other penalization functions. Here we cannot observe the property of the lasso to favor a sparse parameter vector because we are in the one dimensional case.

Table 2 provided the arguments $\hat{\beta}$ which minimizes the corresponding loss function, and by the construction of this example the estimated parameter equals the estimated value for the response $\hat{\beta} = \hat{\mu}$.

Regularization parameter α	**Estimated value $\hat{\beta}$**	**Value loss function at $\hat{\beta}$**
1	5.00	1,405.94
10	4.99	1,450.92
100	4.90	1,896.24
1,000	4.00	5,903.99

Table 2: Estimated values of μ for the lasso penalization.

For the lasso penalization the bias goes from numerically zero ($\alpha = 1$) up to 20% ($\alpha = 1,000$) which seems quite high but compared to the other penalization functions it is indeed rather low. The value of the log-likelihood at $\beta = 5$ is given by 1,400.94 and can serve as a reference value when the values of the loss functions are considered in the following.

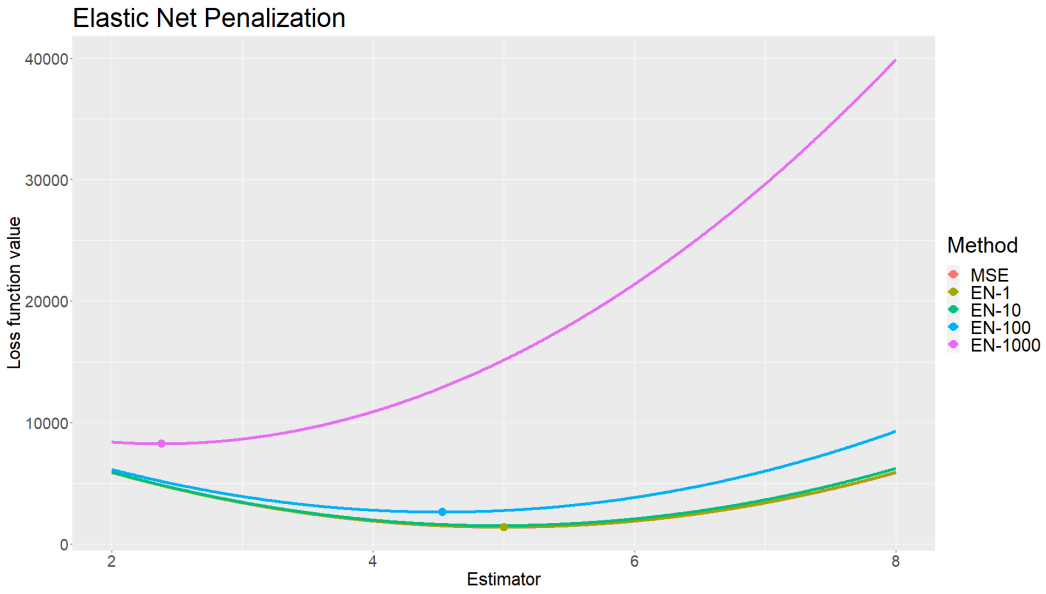

Figure 7: Regularization parameter α on the elastic net penalization ($\lambda = 0.5$).

Referring to Figure 7 we can observe that the elastic net, in the one dimensional case, causes a higher bias then the lasso penalization. Even by a value for the regularization parameter α of 100 we see a proportional lager bias then we have observed before.

Regularization parameter α	**Estimated value $\hat{\beta}$**	**Value loss function at $\hat{\beta}$**
1	5.00	1,414.69
10	4.95	1,537.23
100	4.53	2,652.12
1,000	2.38	8,268.33

Table 3: Estimated values of μ for the elastic net penalization.

If we consider the estimated values for $\hat{\beta}$ under the elastic net we can see that the bias for $\alpha = 1000$ is over 50% which is very high given that this is just a small example.

At this point we can conclude that a regularization parameter which is this high is only reasonable if we are very confident to assume that the parameter is close to zero,

or in the multidimensional case most entries of the parameter vector are close or equal to zero.

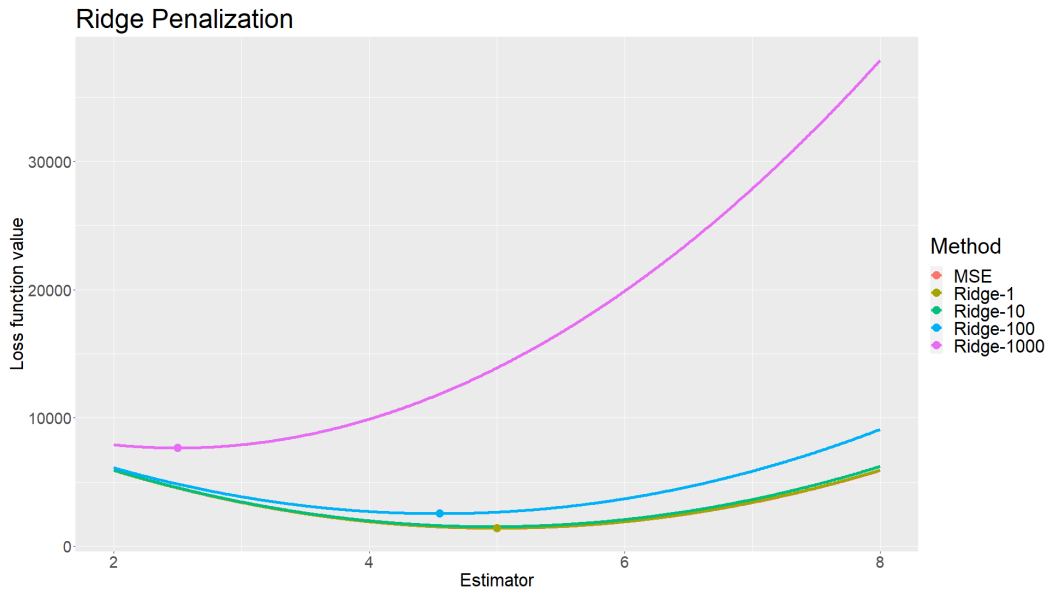

Figure 8: Regularization parameter α on the ridge penalization

Considering the results for the ridge regression it is crucial to keep in mind that here the L2-norm is scaled by the factor 0.5 *and since the results are quite similar to the one from the elastic net penalization Table 4 provides the estimated values* $\hat{\beta}$ *for completeness.*

Regularization parameter α	**Estimated value** $\hat{\beta}$	**Value loss function at** $\hat{\beta}$
1	5.00	1,413.44
10	4.95	1,524.85
100	4.55	2,538.69
1,000	2.50	7,658.56

Table 4: Estimated values of μ for the elastic net penalization.

This small example shows the drawbacks of the embedded models, which are given by introducing bias and also an additional parameter, or for the elastic net penalization even two additional parameters.

Nevertheless, this method is well-established for high dimensional data and the real usability for the Styrian wine grape data will be analyzed in Chapter 8.

4 Explorative Data Analysis

Since the last chapters were more of theoretical nature the following one will present an overview of the data used in the practical part of this master thesis. Two measurement sequences were run by the Institute Dr. Wagner for capturing the two major research questions in this project. The data from both sequences are combined and, in the following, named Styrian wine grape data. A short explorative analysis of this data is provided in the following by using the same method as described in the project report by Fuchs and Friedl (2020).

Due to the fact that organic chemistry is time sensitive and the measurement procedure takes its time, some quality control (QC) measurements were measured additionally to the internal standards of the high-performance liquid chromatography instrument. These QC measurements analyze the same substance, a specific mixture of different grapes, several times during the measurement sequence and allow to extract reference values for the measurements dedicated to the research.

The positioning of these QC measurements in the measurement sequence was chosen in a way that allows to evaluate the variability of the extracted features. If there proofs to be a trend over time, this setting will be able to capture and quantify it. This aspect is of special interest because one measurement took approximately 30 minutes, so the time for the whole sequence (approximately 100 measurements) sums up to 3.000 minutes or 50 hours. This could be enough time causing changes in the experimental environment, like changes to the stationary phase in the hplc, or even lead to some chemical reactions which change the composition of the later measured samples. Both ways tend to bias the results and make them less comparable.

For completeness there was also a proof-of-concept measurement sequence run by the Institute Dr. Wagner in the first place. This sequence was analyzed in order to make sure that classification would be possible at all. Since the results were satisfactory, the sequences presented in this work were rather chosen to present the methodology and theory.

An additional problem is given by the fact that the QC measurements used in the sequences are not comparable and therefore only the later measurement sequences are discussed in this work.

4.1 The Styrian Wine Grape Data

The Styrian wine grape data consist of the extracted data from two different measurement sequences. Both sequences were measured successively, starting with the measurement sequence regarding the analysis of the geographical origin. After the real measurement procedure was done the data was extracted independently from each other with the Profinder software from Agilent.

For a more detailed discussion on the background of the chemical analysis and the data extraction procedure compare the explorative data analysis of the first proof of concept measurement by Fuchs and Friedl (2020).

The second measurement sequence was designed to analyze different varieties. To eliminate the regional factor for this approach all samples are from a narrow geographical region, called "Vulkanland" which is located in the south-east of the Austrian district Styria. The according dataset for this measurement sequence is called **Variety-dataset**.

The first measurement sequence was designed to determine the geographical origin of the wine grape, when the sample is restricted to one variety. Because the sampling process was restricted to voluntary participation there were not enough samples from one variety to fulfill this task, therefore two varieties were chosen to form the foundation of this measurement sequence. Analogously the dataset belonging to this measurement sequence is called **Geography-dataset**.

Since the data extraction algorithm, the Profinder software provided by Agilent, extracts the area and the height of relevant peaks. Each of the datasets contains two data frames. They are called **Variety-Area-dataset** for the data frame containing the area of the peaks from the Variety measurement sequence and analogously the other three data frames.

For a better understanding of the four available data frames some information on the size and the number of features of each data is provided in the following.

	Variety-Area	**Variety-Height**	**Geography-Area**	**Geography-Height**
Sample	95	95	89	89
QC	22	22	22	22
Total	117	117	111	111
Features	2661	2661	2335	2335

Table 5: Meta-information for the available data.

As Table 5 shows the number for the data frames containing the area or the height of the peaks coincide; this is obvious because both describe the same underlying peak in the raw data only using different aspects of the same.

As mentioned before the same methodology of visualization for the range of the high dimensional data as described in the project report by Fuchs and Friedl (2020) was applied. For completeness a short overview of the general idea of this method will be described in the following.

In a first step each feature is handled individually; notice that for each feature there are approximately 100 values through the samples available. A summarizing statistic like the minimum, mean or maximum is applied to each feature projecting approximately 100 values to one single number per feature. These numbers (approximately 2,500) are then treated as a random sample which can be visualized and interpreted.

This allows us to get a first impression of the range and the shape of the available data. Since we are using the data in a more systematic way in the application part of this work only an impression and nothing else is intended at this point.

For example, when the minimum, as function from $\mathbb{R}^n$ to $\mathbb{R}$, is chosen to project the values of a feature then the distribution of these minimum values can be observed as well as some characteristics like the minimum value of the minimum values, which is the overall observed minimum.

Another interesting aspect would be the maximum value of the minimum values which allows to identify features in the dataset which only contain large values.

4.2 Discrimination of the Variety Sequence

Before starting a detailed discussion concerning the values of the features in the Variety-dataset, the following contingency table shows the distribution of the available varieties through the first measurement sequence.

BW	**CH/MO**	**GB**	**GM**	**SAM**	**SB**	**TR**	**WB**	**WR**	**ZW**
3	14	8	14	3	14	6	14	14	5

Table 6: Number of samples on varieties for the first measurement sequence (Variety-dataset).

From Table 6 we observe a very unbalanced experimental design. This is caused by the fact that only 366 samples from 8 different geographical regions and 25 different

varieties were available for designing all measurement sequences.

A complete overview of all available varieties along with their abbreviations and German expressions and the according Wikipedia article is attached in the appendix in Table 34.

4.2.1 Value Range of the Variety-dataset

As mentioned before only a few aspects of the data are concerned in the following, this includes the distribution of the minimum values their means and the maximum values throughout the features.

Minimum Values of the Variety-dataset

The first quantity provided is the minimum as shown in Figure 9 which illustrates the histogram of the minimum values concerning the area of the peaks.

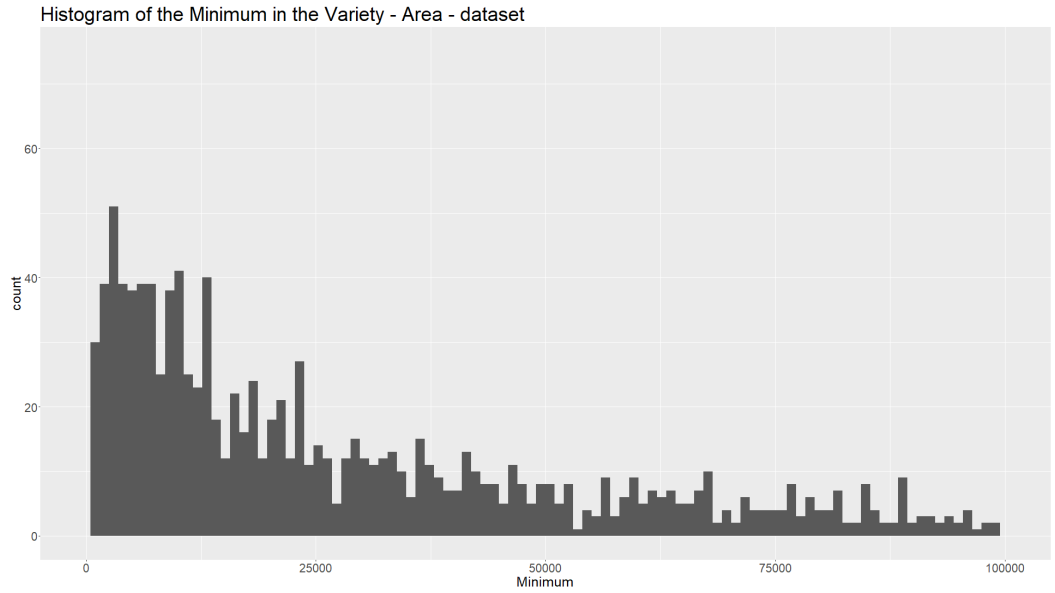

Figure 9: Relevant part of the distribution of the minimum values for the Variety-Area-dataset.

Since the range of the minimum values is quite high, Figure 9 only shows the interesting part of the distribution. As we can observe most features have values quite close to zero. This is not surprising since we know that not all features are necessarily observable in all sample regarding the chemistry.

Leaving the area of the peaks aside, its height in Figure 10 shows the histogram of the corresponding minimum values.

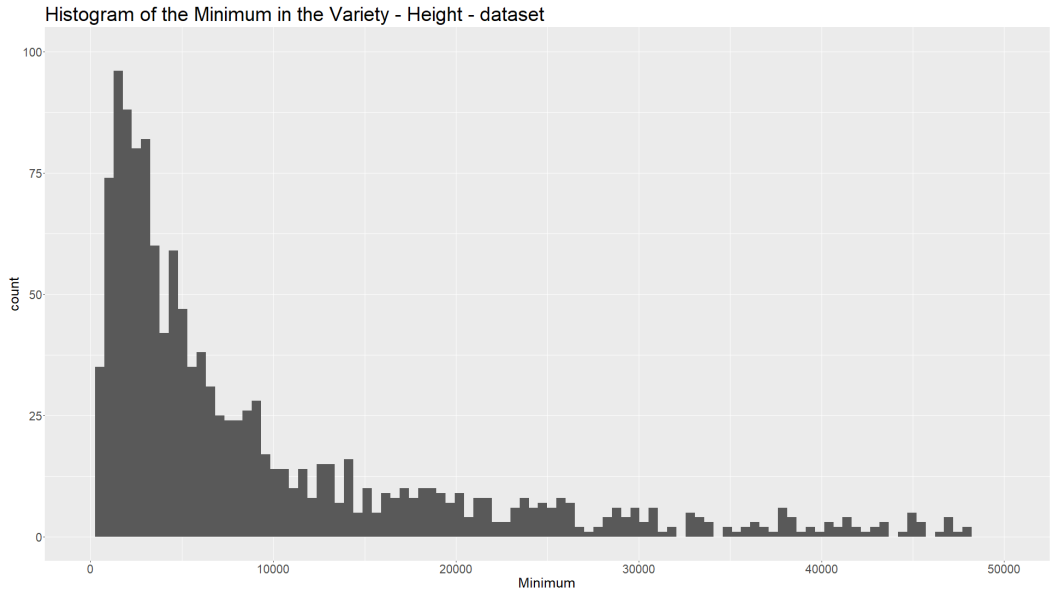

Figure 10: Relevant part of the distribution of the minimum values for the Variety-Height-dataset.

In Figure 10 we can observe that again most values are concentrated around zero. By taking a closer look, we can also see that the right tale flattens out at around 10,000. This value is clearly lower than its counterpart regarding the area data, here it is at around 25,000. By a first inspection, we can verify that the values of the area are in general higher than the values for the height.

Since we have features with a minimum value clearly non-zero, it seems that there are roughly two groups of features.

- The first one captures specific characteristics and is therefore at least for one sample zero, or close to zero.
- The second one is observed in every sample, and therefore varies only by its concentration.

For a more numerical analysis Table 7 shows the minimum, maximum, 1st and 3rd quantile and also the mean and median for the minimum values of the features.

Dataset	Minimum	0.25-Quantile	Median	Mean	0.75-Quantile	Maximum
Area	-127,887	0	3,961	80,084	37,598	21,388,759
Height	0	0	1,459	17,403	7,900	3,114,371

Table 7: Summary statistic for the minimum values of the features in the Variety-dataset.

Remarkable is the fact that the minimum of the minimum values, so the overall minimum, is negative for the area of the peaks. By definition the area is always non-negative which indicates that some parts of the measurement or the data extraction are not entirely accurate.

After consulting Agilent, the company that provides the Profinder software used for the feature extraction, this is a known problem and object of current work. However, due to meetings with the development team this should only occur related to features with small values. Hence, in the following it is treated as artificial noise because we cannot influence the data extraction.

Another interesting aspect provided by Table 7 is the total range of the minimum values. They vary from zero, which is the minimum value of more than a quarter of the features, up to over 21 million. This emphasizes that even if only the minimum values of the features are considered, a lot of variability can be observed.

Mean Values of the Variety-dataset

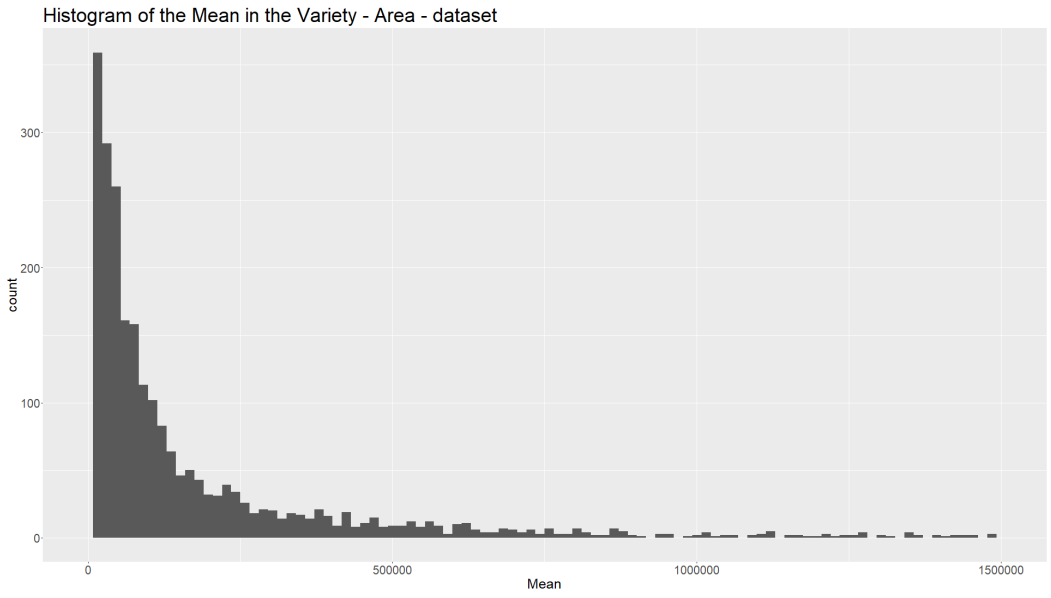

Figure 11: Relevant part of the distribution of the mean values of the Variety-Area-dataset.

If we take a look at the mean values of the features it is possible to get an idea about the different average values of the features and the meaning behind a high or low feature value. Figure 11 provides the histogram for the mean values of the feature when the area of the peaks is considered.

Here we can observe that the values range from close to zero up to over 1,500,000. However, be aware that Figure 11 only shows a restricted part of the whole histogram because the total range is up to over 53,000,000 (c.p. Table 8).

Figure 12 illustrates the histogram of the mean values of the features when the height of the peaks is considered. As for the minimum values, the mean values from the features representing the height of the peaks tend to be lower than the mean values from the feature representing the area of the peaks.

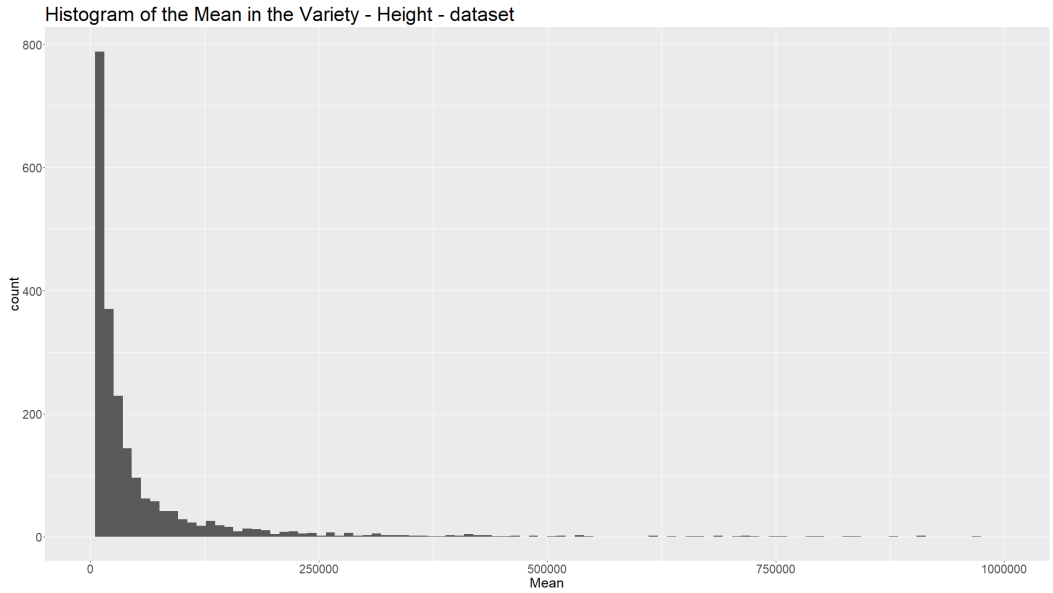

Figure 12: Relevant part of the distribution of the mean values of the Variety-Height-dataset.

The general structure of the distribution for the mean values is the same for both, the area and the height data. In both situations, we observe a concentration around the lower levels and a flatten right tale with a large maximum value. So more technically speaking both distributions have a positive skewness.

This can also be verified by comparing the mean and median provided in Table 8.

Dataset	Minimum	0.25-Quantile	Median	Mean	0.75-Quantile	Maximum
Area	525	27,253	72,655	350,524	211,429	53,179,992
Height	201	6,711	15,578	59,733	41,421	4,918,204

Table 8: Summary statistic for the mean values of the features in the Variety-dataset.

As by the minimum values, for the mean values it can be observed that the range of values differ a lot within the features which could indicate that the classification might work well, or that some data correction or standardization might be necessary in order for the features to be comparable when dealing with numerical methods.

Maximum Values of the Variety-dataset

Providing the empirical distribution of the maximum values for the features and considering the area of the peaks from the underlying measurement Figure 13 shows a similar picture as before.

We again observe a positive skewness and a value range from approximately 12,000 up to over 168,000,000. This means that the features not only quite differ in their minimum values but by their maximum values as well.

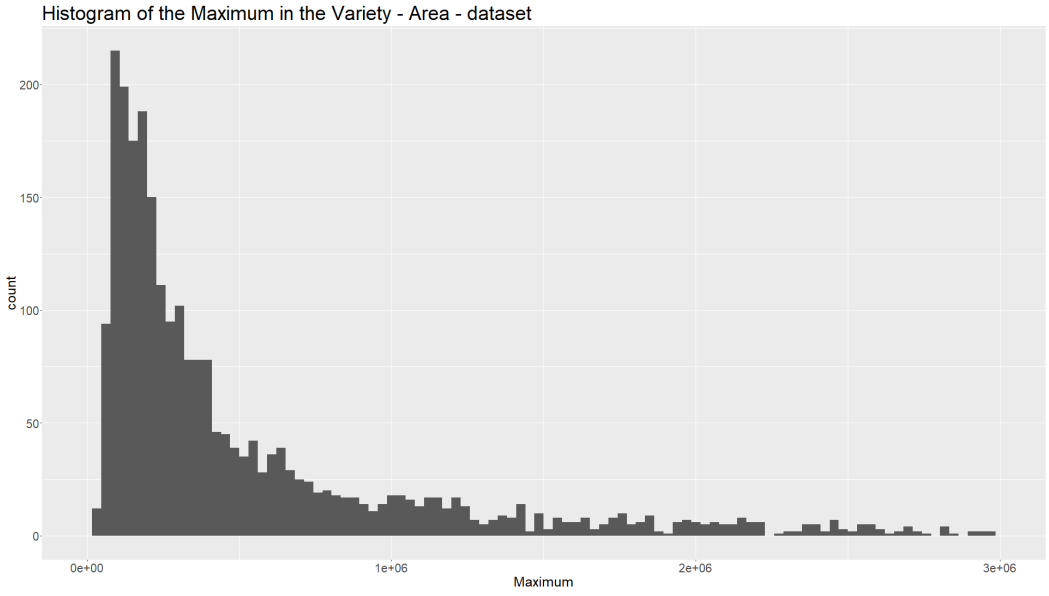

Figure 13: Relevant part of the distribution of the maximum values of Variety-Area-dataset.

The counterpart for the height data can be observed in Figure 14, and up to a general shift of the values the underlying structure is very similar. Here the values range from approximately 4,000 up to over 8,000,000 which is a lot less than for the area data but seems to fit in the context.

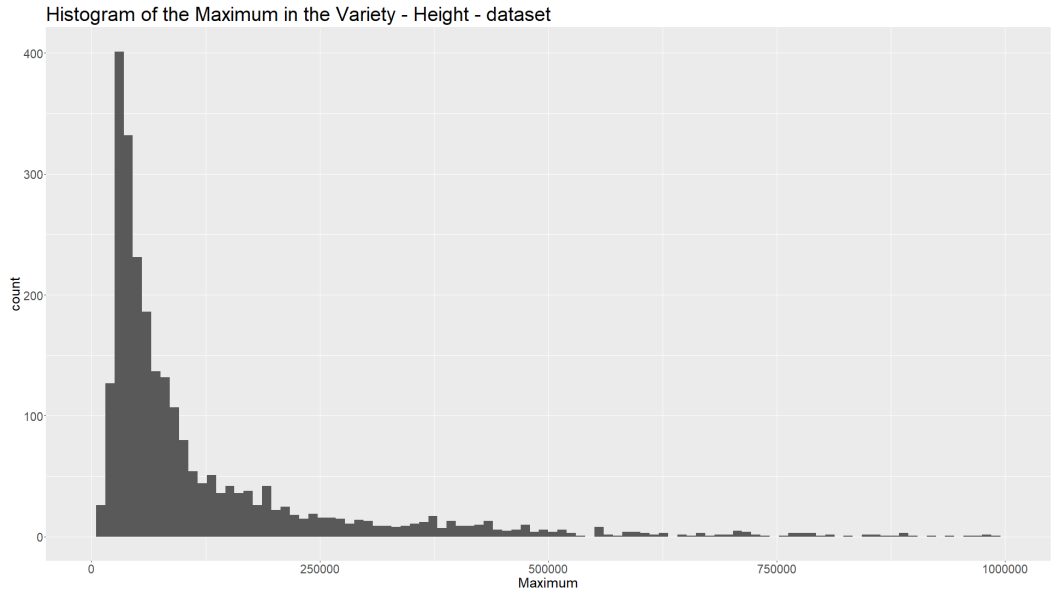

Figure 14: Relevant part of the distribution of the maximum values of Variety-Height-dataset.

Combining the results from Table 9 with the results from Table 7 it can be stated that the range of values in the Area-dataset can be described by the minimum of minimum values which is -127,887 up to the maximum of maximum values given by 168,998,591.

This range seems very large, and clearly is, but it is important to keep in mind that there are 311,337 single values partitioned in 117 samples where each sample has 2,661 features. Hence, regarding the large amount of data, the range must be put in context.

For completeness the general statistics for the maximum values are provide in Table 9.

Dataset	**Minimum**	**0.25-Quantile**	**Median**	**Mean**	**0.75-Quantile**	**Maximum**
Area	12,326	162,133	313,644	1,243,147	791,879	168,998,591
Height	4,348	38,164	67,075	193,517	158,885	8,414,869

Table 9: Summary statistic for the maximum values of the features in the Variety-dataset.

4.3 Discrimination of Geographical Origin Sequence

The second measurement sequence intends to analyze the ability to classify the geographical origin and contains samples belonging to the variety Chardonnay or Morillon (CH/MO) and Sauvignon Blanc (SB). The number of samples on varieties from the geographical origin can be found in Table 10.

Variety	Leithaberg	Slovenia	South Styria	Vulcanic Land	West Styria
CH/MO	4	5	14	14	3
SB	0	9	15	14	7

Table 10: Geographical origins in the Geography-dataset.

Notice that every geographical origin besides Slovenia is a so-called Districtus Austriae Controllatus (DAC) region, which is a special category of location and directly linked to the wine production (c.p. Figure 15). For later analysis, it is important to keep in mind that Slovenia is separate from the other regions due to the different laws for the production of wine and therefore wine grapes.

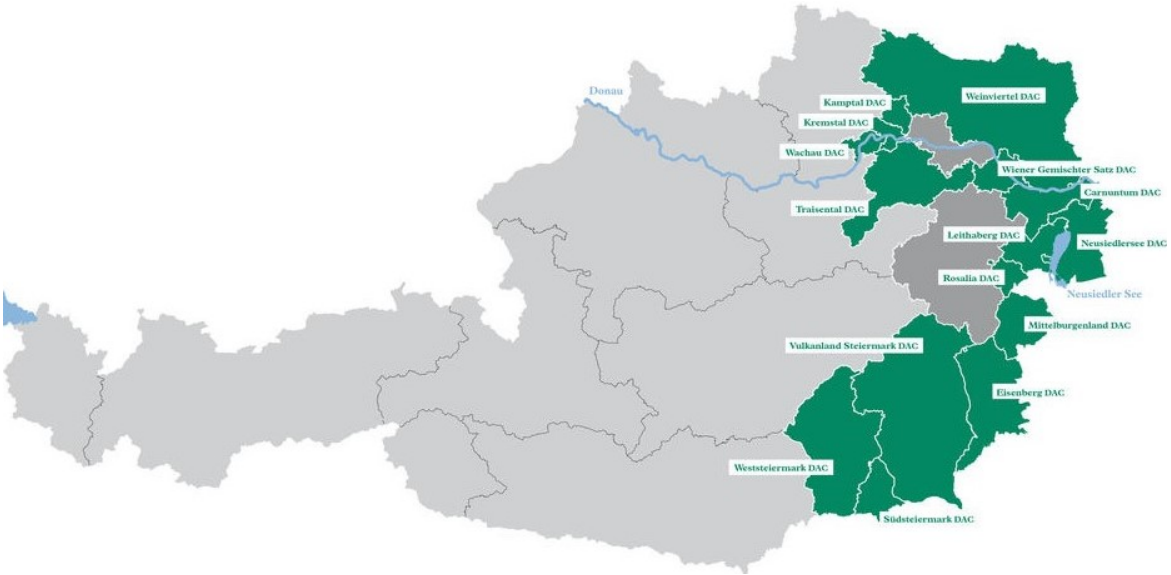

Figure 15: Districtus Austriae Controllatus (DAC) regions in Austria.[1]

[1]https://www.oesterreichwein.at/unser-wein/strategie-des-herkunftmarketings/dac-districtus-austriae-controllatus/dac-gebiete

4.3.1 Value Range of the Geography-dataset

As for the Variety-dataset, a short overview of the value range for the Geography-dataset is provided in the following. We use the same methodology and therefore only some short remarks and comparisons are made between those datasets.

Minimum Values of the Geography-dataset

Starting with the distribution of the minimum values through the features by considering the area of the underlying peaks in Figure 16 we can observe that the concentration around zero is slightly lower than for the variety data.

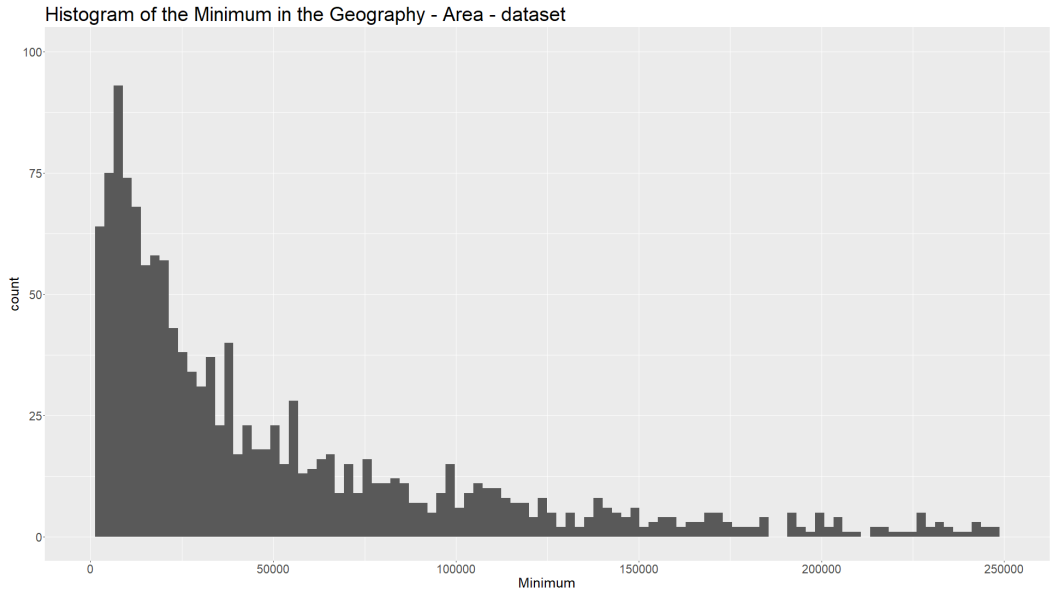

Figure 16: Relevant part of the distribution of the minimum values for the Geography-Area-dataset.

Looking at Figure 17, which shows the distribution of the minimum values for the height data, we can again verify that the values for the height are in general lower than those of the values for the area.

One major observation considering the presented results is that the overall minimum for the Geography-dataset is much lower than for the Variety-dataset. By an exact number of -899,583 it seems that there are again problems induced by the feature extraction algorithm.

But notice that the software and the methodology of high-performance liquid chromatography is not usually used to deal with over 2,000 features compared to only 111 samples at all.

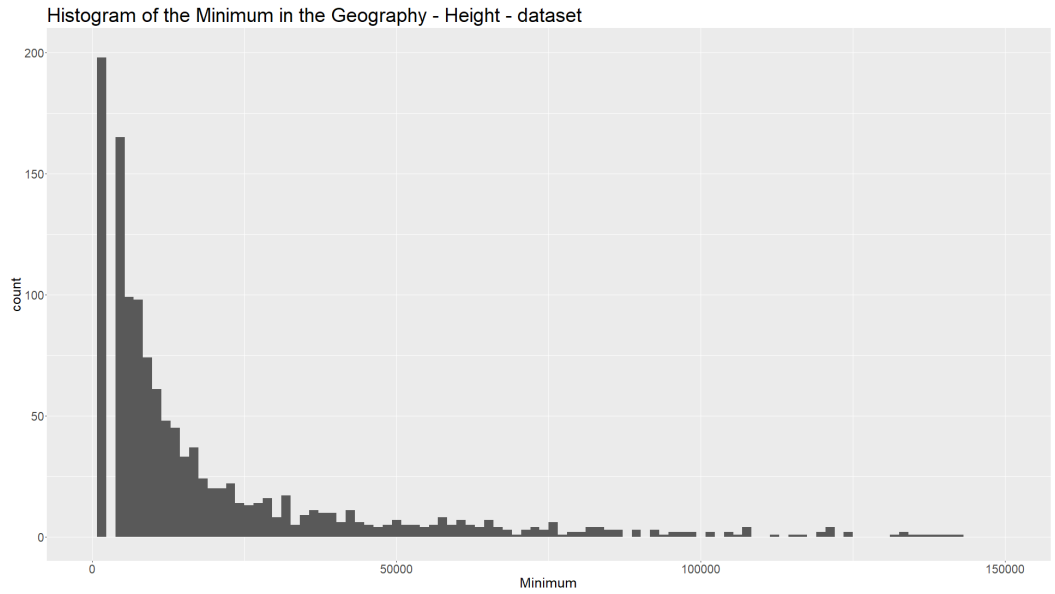

Figure 17: Relevant part of the distribution of the minimum values for the Geography-Height-dataset.

By comparing Table 7 and 11, we can observe that the minimum values in both the Geography- and the Variety-dataset are in the same magnitude when comparing the area data and the height data to each other.

Notice that the features extracted in the Variety-dataset, identified by their mass and retention time, and the features extracted in the Geography-dataset do not necessary overlap. In fact, under 0.5% (i.e. seven features) of the features occur in both datasets.

Dataset	**Minimum**	**0.25-Quantile**	**Median**	**Mean**	**0.75-Quantile**	**Maximum**
Area	-899,583	0	12,596	93,181	55,994	14,245,641
Height	0	0	3,384	20,797	12,303	3,269,746

Table 11: Summary statistic for the minimum values of the features in the Variety-dataset.

Mean Values of the Geography-dataset

For the consideration of the mean values for the area and the height of the peaks, the same systematic used for the Variety-dataset can be observed. This means that there is a positive skewness and lower height than area.

By comparing the results according to the Geography-dataset with the ones described for the Variety-dataset before, it again seems that the values in the Geography-dataset tend to be lower than in the Variety-dataset.

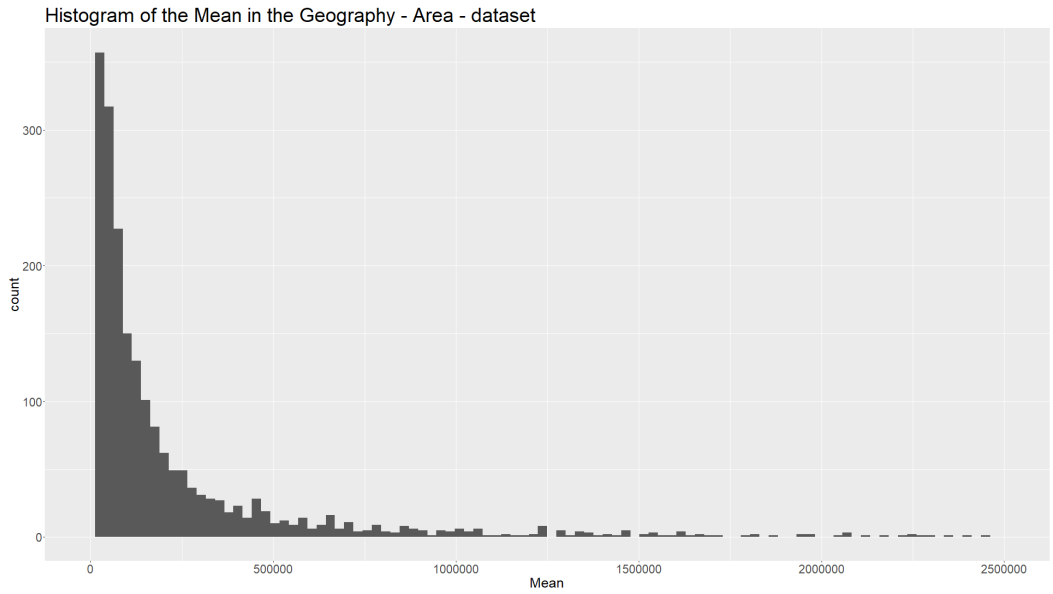

Figure 18: Relevant part of the distribution of the mean values for the Geography-Area-dataset.

For completeness, Figure 18 and 19 provide the relevant part of the distribution of the mean values for the area and the height of the features.

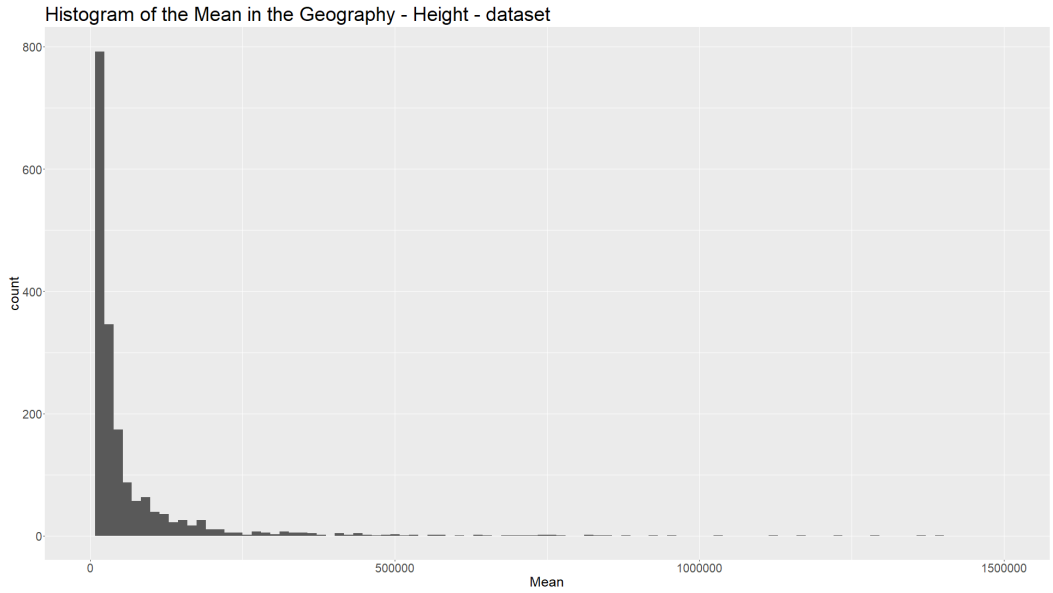

Figure 19: Relevant part of the distribution of the mean values for the Geography-Height-dataset.

For a numerical based comparison, Table 12 provides some statistical characteristics of the empirical distribution for the area and the height data.

Dataset	Minimum	0.25-Quantile	Median	Mean	0.75-Quantile	Maximum
Area	502	33,017	85,332	398,907	231,705	53,021,448
Height	180	8,823	18,532	66,365	45,793	4,940,879

Table 12: Summary statistic for the mean values of the features in the Variety-dataset.

Maximum Values of the Geography-dataset

As shown before the relevant part of the distribution of the maximum values through the features when considering the area of the peaks is provided in Figure 20.

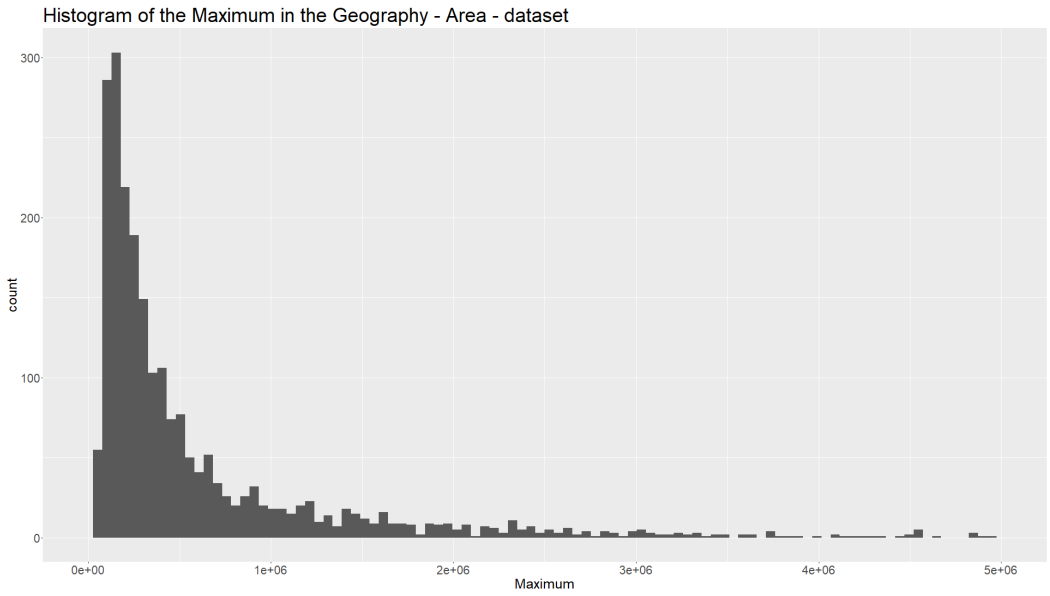

Figure 20: Relevant part of the distribution of the maximum values for the Geography-Area-dataset.

For the Geography-dataset the total range of values of the area of the peaks starts at -899,583 and goes up to 137,811,356. This range is therefore narrower than in the Variety-dataset. This is also valid when negative values are considered as zeros because they are caused by software problems and do not represent the chemical situation.

Considering the height of the peaks, the range of the Geography-dataset, starting at zero and going up to 894,965 is slightly higher compared to the total range in the Variety-dataset where the values go from zero to 8,414,869.

Figure 21 provides the corresponding histogram for a detailed view on the maximum values of the features when the height of the peaks is observed.

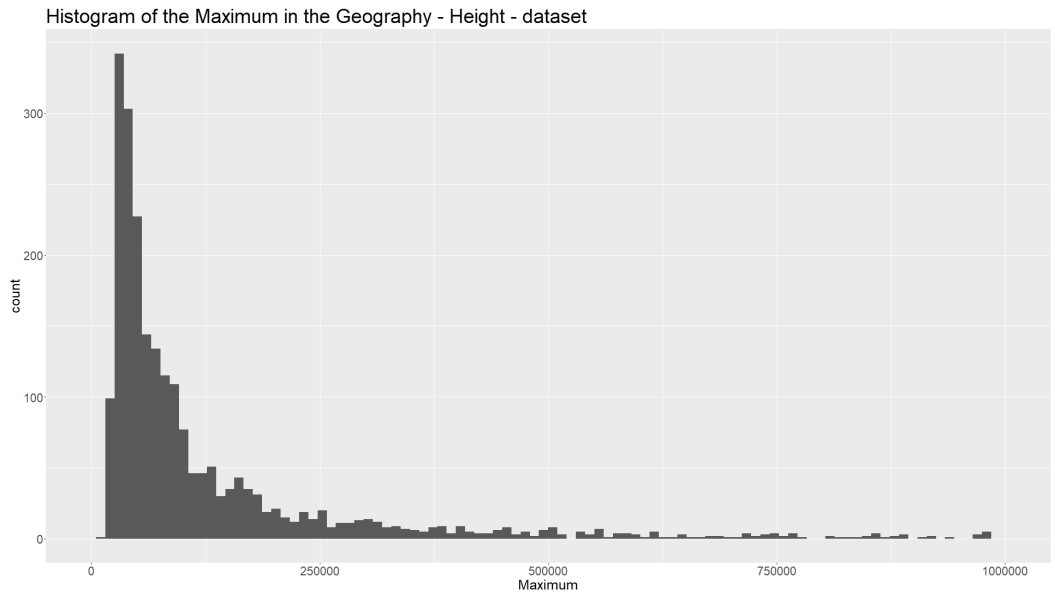

Figure 21: Relevant part of the distribution of the maximum values for the Geography-Height-dataset.

For completeness as for the other aspects of the explorative analysis the characteristics for the maximum values are provided in Table 13.

Dataset	Minimum	0.25-Quantile	Median	Mean	0.75-Quantile	Maximum
Area	37,501	164,237	312,576	1,185,887	753,119	137,811,356
Height	15,041	39,632	69,235	191,884	153,065	8,894,965

Table 13: Summary statistic for the maximum values of the features in the Variety-dataset.

Finally, it can be said that the ranges of the values differ a lot between the features and in general, the values in the height data are lower than the ones for the area data.

We can also conclude that the values corresponding to the area have a wider range in the Variety-dataset than their counterparts in the Geography-dataset. By considering the height of the peaks this behavior is the opposite.

Another point is that features differ between each other by factors of over 1,000. This could cause some problems in the modeling step so the option of standardizing the features by their means and standard deviations will also be considered later.

4.4 Quality Control Measurements

As mentioned before the long amount of time that the chemical analysis requires (over 50 hours) could cause some changes in the experimental setting. On one hand, the instability of certain organic compounds in the samples could lead to differences in the measurements and therefore make the data less comparable. On the other hand, changes in the experimental environment, i.e. by erosion of the stationary phase, could interfere with the analyzed substances and therefore change the results of the measurements over time.

This fact could induce variability in the final data, which does not represent the chemical situation. The internal standards which are usually used for stabilizing the measurements are applied but they do not capture these changes sufficiently. That is why some modified standards are required and used for the later modeling step.

In order to analyze this situation the same quality control sample, a mixture of different grapes, was measured several times. The general scheme behind these measurements is provided by the measurement sequence of the Geography-dataset (c.p Table 15).

To provide a better understanding of this problem some representative features were chosen to illustrate the different types of features and the according problems of the correction in Chapter 5.

For completeness the mixture of the quality control substance, i.e. a mixture of mixtures, which was used in the QC measurements, is drafted in Table 14. A complete overview of the samples used for producing the individual mixtures can be found in the appendix.

Quantity [μL]	**Substance**	**Remark**
300	QC-Ch/Mo_ges	Mixture of different samples with variety Chardonnay or Morillon from all available origins.
300	QC-GM_ges	Mixture of different samples with variety 'Gelber Muskateller' from all available origins.
300	QC-SB_ges	Mixture of different samples with variety Sauvignon Blanc from all available origins.
300	QC-WB_ges	Mixture of different samples with variety 'Weißburgunder' from all available origins.
300	QC-WR_ges	Mixture of different samples with variety 'Welschriesling' from all available origins.
1,500	Total Volume	

Table 14: Mixture for the substance used in the QC measurements.

Abbr.	Type	Sample name	Variety	Region
QC - 1	QC	QC-WHITE-GES 1		
QC - 2	QC	QC-WHITE-GES 2		
S - 1	Sample	320_GEO_CH_VL	CH/MO	VL
⋮	⋮	⋮	⋮	⋮
S - 9	Sample	72_GEO_CH_SUED	CH/MO	SUED
QC - 3	QC	QC-WHITE-GES 3		
QC - 4	QC	QC-WHITE-GES 4		
S - 10	Sample	190_GEO_CH_LEITHABERG	CH/MO	LEITHABERG
⋮	⋮	⋮	⋮	⋮
S - 18	Sample	318_GEO_SB_SUED	SB	SUED
QC - 5	QC	QC-WHITE-GES 5		
QC - 6	QC	QC-WHITE-GES 6		
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
QC - 19	QC	QC-WHITE-GES 19		
QC - 20	QC	QC-WHITE-GES 20		
S - 81	Sample	196_GEO_CH_VL	CH/MO	VL
⋮	⋮	⋮	⋮	⋮
S - 89	Sample	297_GEO_CHSUED	CH/MO	SUED
QC - 21	QC	QC-WHITE-GES 21		
QC - 22	QC	QC-WHITE-GES 22		

Table 15: Measurement sequence for the Geography-dataset.

Notice that the different numbers of the quality measurements in Table 15 indicate that the sample substance was measured at different points in time, but it was always the same substance. Therefore, the effect of time on this substance can be analyzed.

The measurement was chosen to get measured in batches which always measure the substance two times in a row. This makes it possible to analyze the variability of the measurement procedure itself and determine a trend over time.

For further analysis the non-QC measurements are removed from the datasets (Geography and Variety). With this reduced data containing 22 sample elements and approximately 2,300 to 2,600 features the problem of high dimensionality occurs again. Therefore, usual methods are not eligible to be applied but Chapter 5 will show how this problem can be solved by using basic linear regression methods along with some strong assumption on the data.

5 Feature Correction

As mentioned in Chapter 4, there were some special samples, which should allow us to quantify the change of the individual features over time. The resulting data can be used to extract the experimentally induced bias as a preprocessing step to increase the reliability of the data in the later modeling process.

Due to the high dimensionality of the data, one fundamental assumption, which is made in the following, is that each feature is independent of the others. This is quite a strong assumption but is needed in order to apply well-known statistical concepts.

Under this independence assumption, each feature is treated individually as an univariate statistical problem. Details to this treatment will be described later, but scalable and specifically automatic methodology is required because this treatment must be applied up to almost 5,000 (2,300 plus 2,600) times.

One natural approach in order to model a deterministic trend in time in our setting is the usage of classical linear regression models where the time of measurement is the additional information. The usage of this model class allows to describe the connection of the response, in this case real values, to the time of measurement which is described by natural numbers.

We first start with the theoretical background of the so-called polynomial regression which is used later in this chapter. Since this concept is just a special case of the classical linear regression discussed in Chapter 1 no further theory is developed at this point.

After describing the theoretical concept of the feature correction three features from the Variety-Area-dataset are presented to illustrate the feature correction problem.

The last part of the chapter formally describes the feature correction algorithm along with a detailed discussion of the application on the three features mentioned before, and finally the application on all available datasets.

5.1 Polynomial Regression for Feature Correction

The main idea of polynomial regression is the usage of one underlying covariate in order to generate some transformed versions (by monomials) for the usage in the final regression model. In our case, this concept allows us to model the trend over time not only by a constant or linear function, but also to model quadratic or cubic structures.

For a formal definition, assume that x is one covariate. Then the model equation for the polynomial regression model of order p is given by

$$y = f(x, \boldsymbol{\beta}) + \epsilon, \tag{5.1}$$

where $f : \mathbb{R} \times \mathbb{R}^{p+1}$ is a polynomial with degree p and coefficient vector $\boldsymbol{\beta} \in \mathbb{R}^{p+1}$, i.e.

$$f(x, \boldsymbol{\beta}) := \beta_0 + \beta_1 x + \beta_2 x^2 + \cdots + \beta_p x^p.$$

Notice that the difference between Equation (5.1) and Equation (1.4) is the dimensionality of the covariates. But in both models the number of parameters is equal. This derives from the fact that each polynomial regression model can be written as classical linear regression model, as defined in Equation (1.4), by using a variable transformation of the kind

$$z_i = x^i, \qquad i = 1, \ldots, p.$$

With this transformation Equation (5.1) can be written as

$$y = f(x, \boldsymbol{\beta}) + \epsilon = \beta_0 + \beta_1 z_1 + \cdots + \beta_p z_p + \epsilon.$$

As mentioned before, polynomial regression is just a special case of the classical linear regression defined in Chapter 1 and all results, especially the F-test, can be used for this model approach.

Formally, the usage of polynomials does not cause problems in terms of the independence for the design matrix because the new covariates x^i are linearly independent. However, in practice the usage of high order polynomials indeed cause problems because they some kind of collinear behaviour in the new transformed covariates vector is observable. Therefore, the usage of high dimensional polynomials, beside the explainability of the model, should be handled carefully.

To convince ourself that some kind of feature correction is required at all the following section provides some representative examples which should illustrate different aspects of the problem and the algorithm developed to overcome the same.

5.2 Representative Examples

To visualize the methodology of the feature correction used in this work the following features from the Variety-Area-dataset were chosen:

Mass	Retention Time	Structure
381.1526	13.09	linear trend
100.0648	27.54	quadratic trend
102.9479	27.65	cubic trend

Table 16: Examples of features for different trends over time.

Feature 'm=381.1526 rt=13.09'

The first feature, which is used to visualize a linear trend over time, is identified by a mass of 381.1526 and a retention time of 13.09.

Figure 22 shows the area of the peaks over the measurement sequence. Here the QC measurements and the sample measurements are colored differently to illustrate the difference between the same substance measured several times (QC) and the samples required for the modeling process later (Sample).

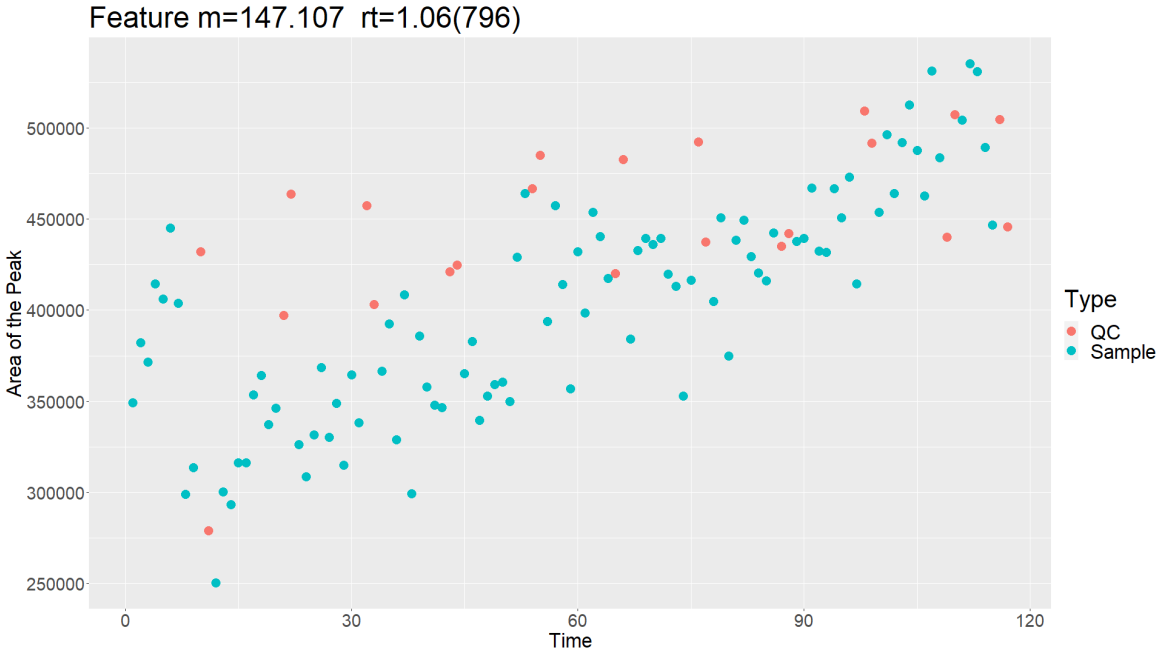

Figure 22: Example of a feature with linear trend over time.

As Figure 22 shows, there exists a clear increasing trend during the measurement sequence. If the QC measurements and the sample measurements are seen as independent groups it can be argued that both groups follow a linear increasing trend in time.

By further inspection, one can observe that the linear trend of the QC measurements is slightly more flat than the linear trend in the sample measurements. Therefore, this example shows that the usage of the QC measurements for estimating the deterministic trend could sometimes fail because the QC data does not capture the trend perfectly.

Feature 'm=100.0648 rt=27.54'

Figure 23 provides the area of the peaks over the measurement sequence for the feature with mass 100.0648 and retention time 27.54. Furthermore, a clearly decreasing behavior over time is observable.

In this case, the QC measurements seem to reflect the general trend of all measurements quite well. Therefore and compared to the previous example, it seems not totally out of mind to use the observable behavior of the QC measurements as a reference for the general trend.

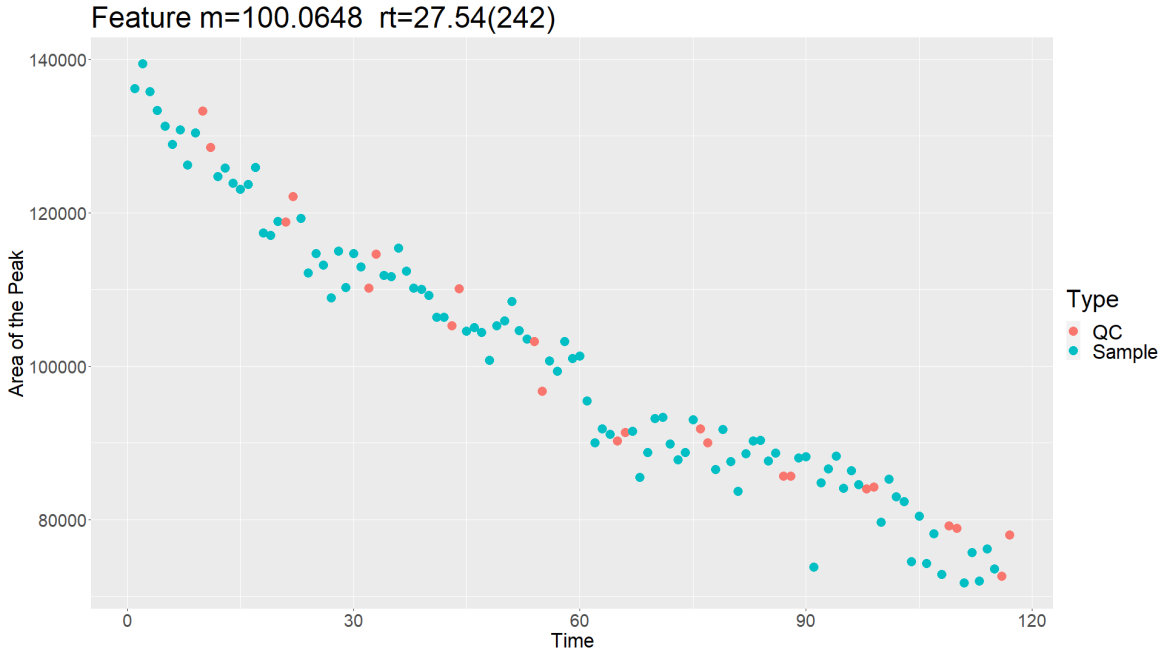

Figure 23: Example of a feature with quadratic trend over time.

These two examples already show totally different trends over time. Therefore, it can be verified that handling each feature individually is a reasonable strategy.

A chemical explanation of these two different trends can be provided by the following fact. The stationary phase interacts with the mobile phase, i.e. erosion, which changes the adhesion between these two phases. This causes that the concentration of some chemical substances at a given retention time could vary in both directions. Therefore, we can observe increasing trends for features as well as decreasing ones.

One could argue that for this example again a linear trend would model the behavior of the QC measurements well. Nevertheless, a closer look reveals that there are two patterns, which are both linear but differ in the gradient. The change point of the provided data could be fixed at around 60. Therefore, it seems reasonable to use a quadratic approach to approximate this trend when only one function for the whole time horizon is used.

Feature 'm=102.9479 rt=27.65'

For the last feature provided in the following, the choice for the feature with mass 102.9479 and retention time 27.65 was made because concerning the colleagues of the Institute Dr. Wagner the observed behavior is quite strange.

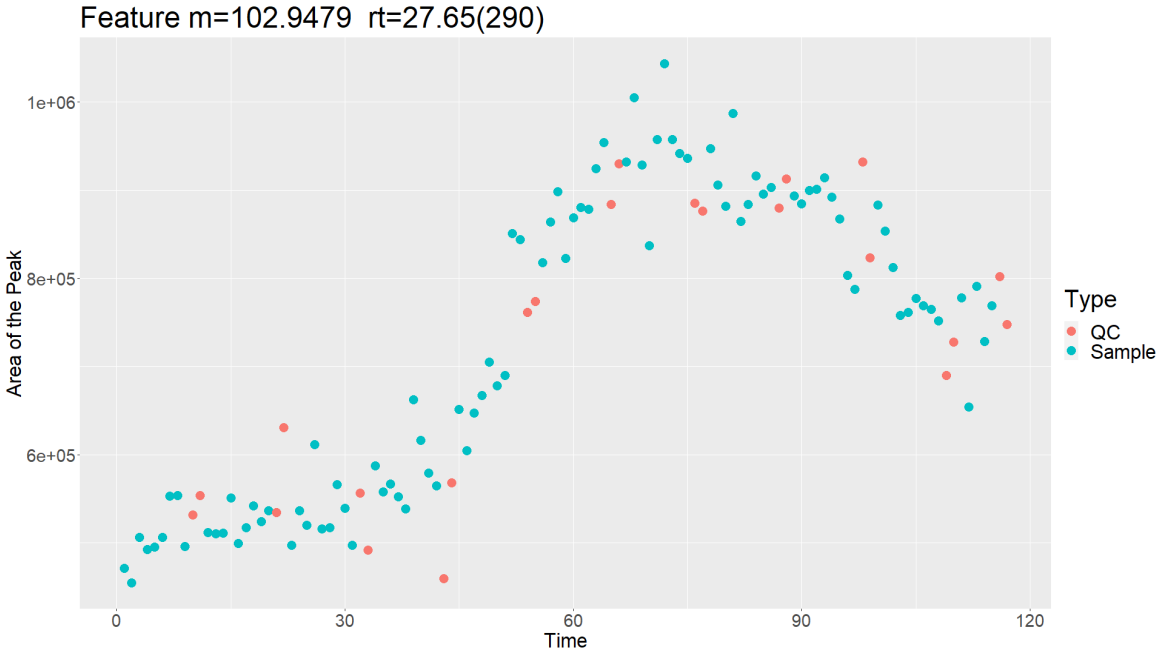

Figure 24: Example of a feature with cubic trend trend over time.

As Figure 24 shows, for approximately the first 60 measurements there is an increasing trend. However, for the later half of the measurements a decreasing trend is clearly detectable. A mathematical description by a polynomial approach with a cubic term would fit this data in a proper way.

From the mathematical or statistical point of view, the data clearly induces structure and the same can be accurately modeled by a cubic polynomial function. However, from a chemical point of view this feature should not be valid at all because these structural changes should not be observable and are likely induced by some problems linked to the measurement procedure.

The practical part of this work is more a proof of concept for classification and the main focus lies on different applications of the multinomial logistic model as classifier. Hence only a intuitive algorithm for data correction is provided in the following.

5.3 Feature Correction Algorithm

The algorithm for the feature correction, applied in this work, follows two steps. Each of them relies on different data and is applied on each feature individually.

1. Determine the deterministic structure of the trend by only using the QC measurements. This is done by fitting a polynomial regression and reducing the degree of the polynomial by the ANOVA up to a point where this step would cause a significance loss of model accuracy.
2. Use the determined regression model to calculate the correction factor for each observation; this includes the non-QC measurements, of the feature.

By defining the area or height of the feature as $y_i,\ i \in \mathcal{I}$, and $\mathcal{I} = \{1, \ldots, n\}$ the set of all indexes or times of measurements for all available samples, the index for the QC measurements can be written as

$$\mathcal{I}_{qc} := \{i \in \mathcal{I} : \text{sample } i \text{ is a QC measurement}\}.$$

The algorithm to determine the trend of the features over the time by using the QC measurements ($\mathcal{I}_{qc}$) can therefore be defined as follows.

Algorithm 2: Determination of the trend over time

for $p \leftarrow p_{max}$ **to** 0 **by** -1 **do**
    fit model corresponding to equation: $y_i := \beta_0 + \beta_1 i^1 + \cdots + \beta_p i^p \quad i \in \mathcal{I}_{qc}$
    if $\mathcal{H}_0 : \beta_p = 0$ *can be rejected* **then**
        break
return $model = \beta_0 + \beta_1 i + \cdots + \beta_p i^p$

For applying Algorithm 2 two aspects need to be specified:

- The setting of the highest degree for the polynomial (p_{max}).
- The specification of a test statistic and significance level for the hypothesis declared in the if condition.

The first problem was solved by consulting the cooperation partner. Therefore, only linear or quadratic trends should be valid. Because the examples from before and a visual inspection of the QC measurements in the Variety-Area-dataset convinced us that some features clearly have a cubic trend (c.p. Figure 24), p_{max} was set to three and features with significant cubic structure got removed from the dataset.

The second problem was solved by using a version of the F-Test (c.p. Theorem 1.1) with the significance level $\alpha = 0.05$ for the linear and quadratic model and the significance level of $\alpha = 0.01$ for the cubic term because these features will be removed afterwards.

For calculating the correction factor it is assumed that there is an adequate model $f(x, \boldsymbol{\beta})$ with estimated parameter $\hat{\boldsymbol{\beta}}$ such that for each measurement $i \in \mathcal{I}$ the mean, μ_i of the response can be estimated as

$$\hat{\mu}_i = f(i, \hat{\boldsymbol{\beta}}) \qquad i \in \mathcal{I}.$$

Furthermore if it is assumed that the model specifies the trend over time. Then the correction factor is defined to project the value of each observation to the value it would have, under the model, when it would be measured in the middle of the sequence. Therefore, the correction factor K_i for observation i is defined as

$$K_i := \frac{\hat{\mu}_{\bar{i}}}{\hat{\mu}_i} \qquad i \in \mathcal{I},$$

where $\hat{\mu}_{\bar{i}}$ is the estimated value for the feature in the middle of the measurement sequence. This reference point was chosen since in the middle of a measurement sequence the results should be as reliable as possible.

The complete R-code used for the correction of the features can be found in the appendix.

In the case of a model that only contains the intercept, the correction factor would be a constant factor for all observations. Since this multiplication should not improve the data quality, the original data are returned in this situation.

5.4 Application to Specified Features

Algorithm 2 combined with the calculation of the correction factor (c.p. R-code) was used to correct all available features. To visualize the effect of the method for the features provided in Section 5.2 the original data along with the automatically estimated deterministic trend and finally the corrected data are plotted.

Feature 'm=381.1526 rt=13.09'

Figure 25 shows the linear trend, but we can observe that the trend was determined only by the QC measurements and not by all samples which are plotted in the figure. Therefore, after the correction a slight linear trend is still present. Nevertheless, compared to the original data the gradient of this linear trend of the corrected data is much closer to zero than before.

It is important to keep in mind that only the QC measurements are measurements with the same underlying chemical substance, all other data points come from different chemical samples. Therefore, variability or certain patterns observable in all data points can be caused by the chemical situation itself.

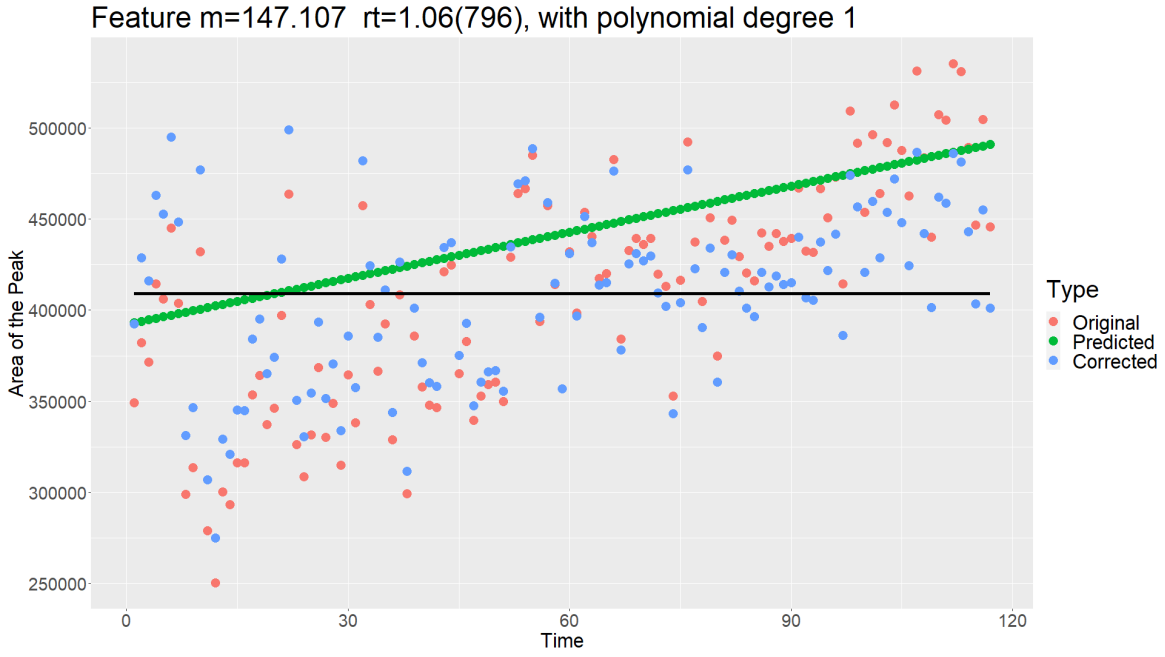

Figure 25: Example of a feature with linear trend over time.

It seems that the method applied to this feature at least does not make things less

comparable, which is also a good property for a preprocessing step. After all, this is just a first approach to take the change of the experimental environment into account.

Feature 'm=100.0648 rt=27.54'

In Figure 26 we can observe the best case in some sense. First of all there is a clear trend over the measurement sequence, and because it is observable for all QC measurement it is a trend independent of the underlying chemical substances.

Therefore, we see that the correction algorithm applied on this feature indeed increases the comparability of the results. Here the variability of the corrected data can be assumed to come from the difference related to the chemical situation.

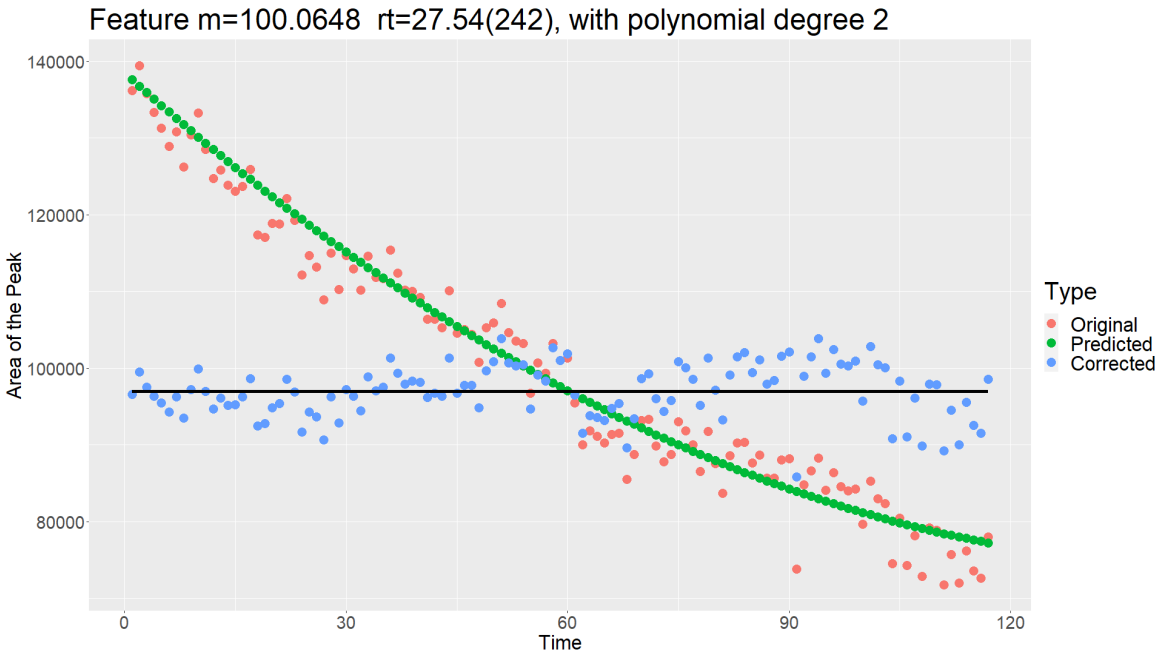

Figure 26: Example of a feature with quadratic trend over time.

Feature 'm=102.9479 rt=27.65'

Analog to the features discussed before Figure 27 shows the area of the feature with mass 102.9479 and retention time 27.65. The green line visualize the clear detectable cubic structure of the QC measurements.

Here, the last presented feature shows again that the method seems to work quite well when it comes to identify the trend over time, but also shows that there are

trends found in the data, which cannot be explained from the chemical point of view.

Due to this problem all features, which have a clear cubic structure, got removed in the preprocessed and corrected dataset. Remember that for the cubic polynomial regression the significance level α was set to 0.01 to ensure that only features with a clear cubic trend got removed.

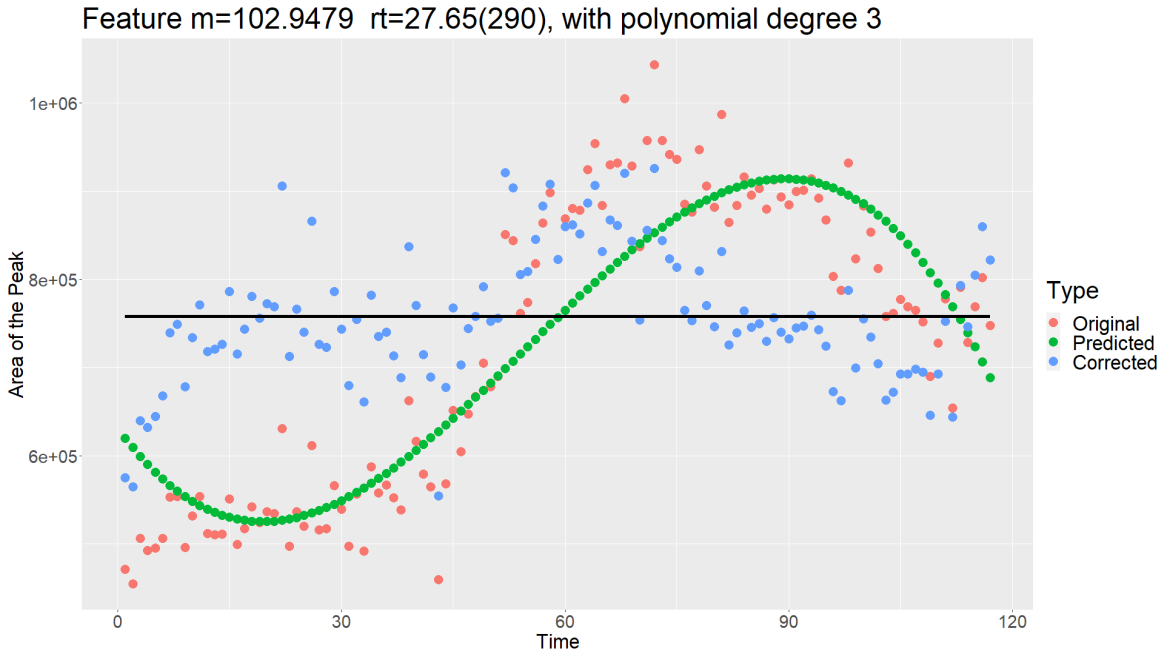

Figure 27: Example of a feature with cubic trend over time.

5.5 Application to All Available Datasets

After discussing the methodology, the necessity and the effect of the feature correction, a short overview for the available data is provided. Notice that for each measurement sequence there are two quantities, i.e. the area and the height of the peaks, available which leads to the datasets provides in the following.

As Table 17 shows, the available samples are of the same sizes for each sequence since each sequence describes the same underlying peaks. Nevertheless, the number of available features varies according to the number of features with significance cubic trend over time of the QC measurements.

Name	Sequence	Quantity	Samples	Features
Variety-Area-Original	Variety	Area	95	2,661
Variety-Area-Corrected	Variety	Area	95	2,591
Variety-Height-Original	Variety	Height	95	2,661
Variety-Height-Corrected	Variety	Height	95	2,590
Geography-Area-Original	Geography	Area	89	2,335
Geography-Area-Corrected	Geography	Area	89	2,171
Geography-Height-Original	Geography	Height	89	2,335
Geography-Height-Corrected	Geography	Height	89	2,030

Table 17: All available datasets.

Also notice that features where the modeling process failed, because of computational issues, are traded like they would have had a constant trend structure and therefore are not changed at all. This method should ensure that the datasets are not artificially biased and so the data quality at least does not decrease.

If we would use all eight datsets for the modeling process and provide the results in this work a lot of redundant information would be presented and discussed. Since this correction process is more a technical requirement to make the features comparable and not the main focus of this work only the corrected data are further analyzed.

This means that in the following chapters the dataset **Variety-Area** is linked to the data representing the area of the peaks in the variety measurement sequence, after the correction process described in this chapter. Analogously are the datasets **Variety-Height**, **Geography-Area** and **Geography-Height** representing the data after the correction.

After discussing different aspects of the data and the sampling procedure in detail and preprocess the raw data to obtain datasets where the features can be considered to be comparable over the measurement time, the following chapters are dealing with the modeling process.

Here we start with the simplest one, i.e. the filter model, then we take a closer look to the most resource intensive model, represented by the stepwise forward selection, and finally discussing the application of penalization terms in the context of embedded models.

6 Filter Models for Feature Selection

After discussing the data for the practical part, this chapter will explore the general concept of filter models and especially some filter in detail. The first subsection will provide three different filters which are used in this work. Every presented filter is univariate, which means that each feature is ranked individually (c.p. Chapter 3). Therefore, no groups of features got evaluated simultaneously. This restriction was chosen in order to keep the required computational time at a reasonable pace.

The first filter is based on a measure of the correlation between the response (categorical variable) and the feature (nominal variable). The following chapter will discuss and explain possible difficulties emerging in this setting and provides an overview of some strategies that may be used to overcome the aforementioned.

The second filter is based on effect measures, more precisely the effect measure used in the context of regression analysis. By changing the question from "How important is this feature?" to the equivalent question of "How strong do the classes stratify the values of the feature?" some classical results and measures from the analysis of variance can be used.

The last one is not a filter in the sense of Section 3.3, but all these filters are used for the wrapper models in Chapter 7 which is why it is also included in the following. The main idea is given by using the classification accuracy as a measure, with the restriction that each feature is used individually.

6.1 Different Filter for Features Extraction

In the following the point-biserial correlation coefficient, the coefficient of determination and the single predictor classification accuracy measure are explored in more details.

Each filter is based on a different approach concerning the data and since this work covers a lot of topics only a heuristic overview and no deeper theory can be provided here.

6.1.1 Point-Biserial Correlation Coefficient

The point-biserial correlation coefficient is a special case of the empirical correlation coefficient, also known as Pearson's correlation coefficient, and both are based on the concept of correlation. Therefore, we will first introduce the correlation along with some properties for the later use. Because these are well known results no proofs are provided in the following.

Definition 6.1. *(Correlation, see Casella and Berger, 2002)*
The correlation of a random variable x and a random variable y is the number defined by

$$\rho_{xy} = \frac{Cov(x,y)}{\sigma_x \sigma_y} = \frac{\mathbb{E}[(x-\mu_x)(y-\mu_y)]}{\sqrt{\mathbb{E}[(x-\mu_x)^2]\mathbb{E}[(y-\mu_y)^2]}}.$$

The value ρ_{xy} is also called the correlation coefficient.

Some properties of the correlation coefficient are shown in the following theorems.

Theorem 6.1. *(Correlation and Independence, see Casella and Berger, 2002)*
If x and y are independent random variables, then $Cov(x,y) = 0$ and therefore $\rho_{xy} = 0$.

Theorem 6.2. *(Range of the Correlation, see Casella and Berger, 2002)*
For any random variables x and y,

- $-1 \leq \rho_{xy} \leq 1$.
- $|\rho_{xy}| = 1$ *if and only if there exist numbers $a \neq 0$ and b such that $\mathbb{P}[y = ax+b] = 1$. If $\rho_{xy} = 1$, then $a > 0$, and if $\rho_{xy} = -1$, then $a < 0$.*

For a measure we want to ensure that only positive values will occur, but we can even restrict it to take values between zero and one. Therefore, Theorem 6.1 and 6.2 show that the absolute value of the correlation coefficient could be a good candidate for a measure. Since the correlation coefficient is in general not known, the empirical correlation coefficient can serve as an estimator for this quantity.

Definition 6.2. *(Empirical Correlation Coefficient)*
Let $(x_1, y_1), \ldots, (x_n, y_n)$ be a bivariate random sample. By plugging in the moment estimator for the expectation and variance used in the correlation coefficient (Definition 6.1) the empirical correlation coefficient is defined as

$$\hat{\rho}_{xy} := \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}.$$

This quantity is also called Pearson's correlation coefficient.

After formulating the theoretical foundation and the empirical version of the correlation coefficient, the derivation of the point-biserial correlation coefficient can be done by relabeling the data and calculation. This is provided in the following.

Lemma 6.3.
Let $(x_1, y_1), \dots, (x_n, y_n)$ be a bivariate random sample, where $x_i \in \mathbb{R}$ and $y_i \in \{0, 1\}$ (dichotomous variable). Then the empirical correlation coefficient is given by

$$\hat{\rho}_{xy} = \frac{(\bar{x}_1 - \bar{x}_0)}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}} \sqrt{\frac{n_1 n_0}{n}}.$$

Here $\bar{x}_1$ is the mean of all x_i, where $y_i = 1$ and analog for $\bar{x}_0$. This quantity is also called point-biserial correlation coefficient and can be extended to each y taking two different values k_0 and k_1.

Proof.
For simplicity let $\mathcal{I} := \{1, \dots, n\}$ be the index set of the entire sample, while $\mathcal{I}_0 := \{i \in \mathcal{I} : y_i = 0\}$ and $\mathcal{I}_1 := \{i \in \mathcal{I} : y_i = 1\}$ are the indices of the sample elements corresponding to class zero or one. Then the mean of $x_1, \dots, x_n$ and $y_1, \dots, y_n$ can be written as

$$\begin{aligned}
\bar{y} &= \frac{1}{n}\sum_{i \in \mathcal{I}} y_i = \frac{1}{n}\sum_{i \in \mathcal{I}_1} y_i + \frac{1}{n}\sum_{i \in \mathcal{I}_0} y_i = \frac{n_1}{n} \\
n\bar{x} &= \sum_{i \in \mathcal{I}} x_i = \sum_{i \in \mathcal{I}_1} x_i + \sum_{i \in \mathcal{I}_0} x_i = n_1\bar{x}_1 + n_0\bar{x}_0 \Rightarrow \bar{x} = \frac{n_1}{n}\bar{x}_1 + \frac{n_0}{n}\bar{x}_0,
\end{aligned}$$

where $n_i := |\mathcal{I}_i|,\ i = 0, 1$, is the sample size of each group. With this calculation the value of the numerator of the empirical correlation coefficient is given by

$$\begin{aligned}
\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) &= \sum_{i=1}^{n}(x_i - \bar{x})y_i - \underbrace{\sum_{i=1}^{n}(x_i - \bar{x})\bar{y}}_{=0} = \sum_{i=1}^{n}(x_i - \bar{x})y_i = \sum_{i \in \mathcal{I}_1}(x_i - \bar{x}) \\
&= n_1\bar{x}_1 - \frac{n_1}{n}(n_0\bar{x}_0 + n_1\bar{x}_1) = \bar{x}_1\left(n_1 - \frac{n_1^2}{n}\right) - \bar{x}_0\left(\frac{n_1 n_0}{n}\right) \\
&= \bar{x}_1\frac{n_1(n - n_1)}{n} - \bar{x}_0\frac{n_1 n_0}{n} = (\bar{x}_1 - \bar{x}_0)\frac{n_1 n_0}{n}.
\end{aligned}$$

For the denominator the following calculation will allow to get the final formulation for the correlation coefficient

$$\sum_{i=1}^{n}(y_i - \bar{y})^2 = \sum_{i=1}^{n} y_i^2 - n\bar{y}^2 = n_1 - \frac{n n_1^2}{n^2} = \frac{n_1}{n}(n - n_1) = \frac{n_1 n_0}{n}.$$

Putting everything together ends up in

$$\hat{\rho} := \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} = \frac{(\bar{x}_1 - \bar{x}_0)n_1 n_0/n}{\sqrt{n_1 n_0/n \sum_{i=1}^{n}(x_i - \bar{x})^2}}$$
$$= \frac{(\bar{x}_1 - \bar{x}_0)}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}} \sqrt{\frac{n_1 n_0}{n}}.$$

□

The restriction of the point-biserial correlation coefficient can by observed by its definition. Here only two classes are allowed for the categorical variable y_i. If we extend the number of the classes and use the same idea of writing them as numbers, we implicitly assume an order which is not possible for categorical variables. Therefore, another strategy is required to achieve a meaningful measure for the filter model when more than two classes are available.

For a given response $\mathbf{y} = (y_1, \ldots, y_n)^t$ with values in $\mathcal{K} = \{k_1, \ldots, k_K\}$ the following mapping will be useful:

$$f : \mathcal{K}^n \to \{0,1\}^{n \times K}$$
$$\mathbf{y} = (y_1, \ldots, y_n)^t \to (\mathbb{1}_{\mathbf{y}=k_1}, \ldots, \mathbb{1}_{\mathbf{y}=k_K}),$$

where $\mathbb{1}_{\mathbf{y}=k_i} = (\mathbb{1}_{y_1=k_i}, \ldots, \mathbb{1}_{y_n=k_i})^t$ and $\mathbb{1}$ is the indicator function.

With this setting the point-biserial correlation coefficient can be applied to each column of $f(\mathbf{y})$ and afterwards a summarizing method $g : \{0,1\}^{n \times K} \to \mathbb{R}$ is needed to get one final number as a measure for the filter model. With this method the extended point-biserial correlation coefficient can be defined in the following way.

Definition 6.3. *(Extended Point-Biserial Correlation Coefficient)*
Let $(x_1, y_1), \ldots, (x_n, y_n)$ be a bivariate random sample, where $\mathbf{x} = (x_1, \ldots, x_n)^t \in \mathbb{R}^n$ and $\mathbf{y} = (y_1, \ldots, y_n)^t \in \mathcal{K}^n$ with available classes $\mathcal{K} = \{k_1, \ldots, k_K\}$. Furthermore let $g : [-1,1]^K \to [0,1]$ be a predefined function, then the extended point-biserial correlation coefficient measure (epbccm) is defined as

$$\rho^{epbccm} := g((\rho^{pbcc}(\mathbf{x}, \mathbb{1}_{\mathbf{y}=k_1}), \ldots, \rho^{pbcc}(\mathbf{x}, \mathbb{1}_{\mathbf{y}=k_K})).$$

Possible choices for g could be

- $g(z_1, \ldots, z_K) := min(|z_1|, \ldots, |z_K|)$,
- $g(z_1, \ldots, z_K) := \sum_{j=1}^{K} |z_j|/K$,
- $g(z_1, \ldots, z_K) := max(|z_1|, \ldots, |z_K|)$.

Remark 6.1.
Notice that each example uses the absolute value of the point-biserial correlation coefficient since only the 'strength' of the correlation is of interest not the direction (positively or negatively correlated).

After some evaluation of different summarizing functions g it seems that for this project the choice $g(z_1, \ldots, z_K) := \sum_{j=1}^{K} |z_j|/K$ accomplishes the best results. This could be explained by the fact that for good results the correlation to each available class must be high where, for example the choice $g(z_1, \ldots, z_K) := \max(|z_1|, \ldots, |z_K|)$ favors features which have a high correlation to only one class. With an increasing class number K it is a very useful property.

Therefore, when $(x_1, y_1), \ldots, (x_n, y_n)$ is a bivariate random sample, $\mathbf{x} = (x_1, \ldots, x_n)^t \in \mathbb{R}^n$ and $\mathbf{y} = (y_1, \ldots, y_n)^t \in \mathcal{K}^n = \{k_1, \ldots, k_K\}^n$, if not otherwise stated, the extended point-biserial correlation coefficient measure or short epbccm is given by

$$\rho^{epbccm} := g(\rho^{pbcc}(\mathbf{x}, \mathbb{1}_{\mathbf{y}=k_1}), \ldots, \rho^{pbcc}(\mathbf{x}, \mathbb{1}_{\mathbf{y}=k_K})) = \frac{1}{K} \sum_{j=1}^{K} |\rho^{pbcc}(\mathbf{x}, \mathbb{1}_{\mathbf{y}=k_j})|.$$

For completeness the following lemma shows that epbccm indeed takes values in $[0, 1]$.

Lemma 6.4. *(Value range of epbccm)*
Let $(x_1, y_1), \ldots, (x_n, y_n)$ be a bivariate random sample, where $\mathbf{x} = (x_1, \ldots, x_n)^t \in \mathbb{R}^n$ and $\mathbf{y} = (y_1, \ldots, y_n)^t \in \mathcal{K}^n = \{k_1, \ldots, k_K\}^n$. Then the extended point-biserial correlation coefficient measure defined as

$$\rho^{epbccm} = \frac{1}{K} \sum_{j=1}^{K} |\rho^{pbcc}(\mathbf{x}, \mathbb{1}_{\mathbf{y}=k_j})|,$$

takes only values between zero and one.

Proof.
By Theorem 6.2 each argument of the extended point-biserial correlation coefficient measure takes values between zero and one. And due to the fact that the sum is a monotone increasing function in each argument the following equation holds:

$$0 = \frac{1}{K} \sum_{j=1}^{K} 0 \leq \frac{1}{K} \sum_{j=1}^{K} |\rho^{pbcc}(\mathbf{x}, \mathbb{1}_{\mathbf{y}=k_j})| \leq \frac{1}{K} \sum_{j=1}^{K} 1 = 1.$$

□

6.1.2 Coefficient of Determination R^2

For the next measure or filter we do not focus so much on the correlation between the classes and the feature itself. For this method to each value of the feature $x_i,\ i = 1, \ldots, n$ we assign the corresponding class $y_i \in \mathcal{K}$. Therefore we get $K = |\mathcal{K}|$ different groups where each group consists of $n_j,\ j = 1, \ldots, K$ values of the feature $\mathbf{x}$.

If it is assumed that the feature classifies our problem very well, this is equivalent to saying that their group means $\bar{x}_j,\ j = 1, \ldots, K$, differ significantly from each other. If they do not differ, it can be stated that this feature, only used as available information, is not able to solve the classification problem in a proper way.

For using already established concepts this problem can be formulated as analysis of variance, and therefore as a special case of the classical linear regression analysis discussed in Chapter 1. The additional assumptions following along with this formulation are

- the features are following a Gaussian distribution and
- the variance of all observations is σ^2 (homoscedasticity).

With these assumptions the class memberships can be described by a factor which results in a model equation of the form

$$x_i = \beta_0 + \beta_2 \cdot \mathbb{1}_{k_2}(y_i) + \cdots + \beta_K \cdot \mathbb{1}_{k_K}(y_i) + \epsilon_i,$$

where x_i is the value of the feature for the ith sample element, and y_i is the class for the same.

Remark 6.2.
Notice that the role of the feature and the corresponding class have changed. For the classification problem our response is the class represented by y_i and we have the value of the feature as additional information x_i, with design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$. By using the strategy from above we consider the values of the feature as response and the class as additional information. Therefore, the notation in the following seems unusual when dealing with the classical regression problem since now the design matrix is given by $\mathbf{Y} \in \mathbb{R}^{n \times 1}$.

With this model the group means are estimated by $\bar{x}_j = \hat{\beta}_0 + \hat{\beta}_j$, and the F-test (c.p. Theorem 1.1) can be applied for testing the hypothesis

$$\mathcal{H}_0 : \beta_2 = \cdots = \beta_K = 0 \qquad vs \qquad \mathcal{H}_A : \exists j \in \{2, \ldots, K\} \ s.t.\ \beta_j \neq 0.$$

Now an obvious choice for a measure could be the p-value according to this hypothesis test with an appropriate test statistic. The application of the p-values on the Styrian

wine grape data showed that it tends to be more a zero or one measure. Therefore, this choice seems not appropriate for a measure which should rank, or better distinguish, over 2,000 features.

As an alternative the coefficient of determination will be introduced in the following and is used as a measure for a filter model in this chapter. However, it is a well-known concept which is why the definition along with some associated properties are provided in the following for reasons of completeness.

Theorem 6.5. *(Sum of Squares, see Draper and Smith, 1998)*
For a linear regression model with intercept β_0 and a design matrix $\mathbf{Y}$ the variance can be segmented into

$$\underbrace{\sum_{i=1}^{n}(x_i - \bar{x})^2}_{SST} = \underbrace{\sum_{i=1}^{n}(x_i - \hat{\mu}_{xi})^2}_{SSR} + \underbrace{\sum_{i=1}^{n}(\hat{\mu}_{xi} - \bar{x})^2}_{SSE},$$

where

- *SST is the Sum of Squares Total,*
- *SSR is the Sum of Squared Residuals and*
- *SSE is the Sum of Squared Errors.*

Definition 6.4. *(R^2, see Draper and Smith, 1998)*
For a linear regression model with intercept β_0 and design matrix $\mathbf{Y}$ the coefficient of (multiple) determination is defined as

$$R^2 := \frac{SSR}{SST} = \frac{\sum_{i=1}^{n}(x_i - \hat{\mu}_{xi})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}.$$

By using the identity $SST = SSR + SSE$ the coefficient of determination can also be written as

$$R^2 = \frac{SST - SSE}{SST} = 1 - \frac{SSE}{SST} = 1 - \frac{\sum_{i=1}^{n}(\hat{\mu}_{xi} - x_i)^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}.$$

Remark 6.3.
Notice that by the definition of R^2, $R^2 = 1$ reflects a perfect fit of the data and $0 \leq R^2 \leq 1$. So in this sense the coefficient of determination has the same behavior as the epbccm, which makes the interpretation in the later application easier.

Since R^2 increases with the number of parameters an adjusted version takes care of this problem but the explainability in terms of variance ratios is lost.

Definition 6.5. *(Adjusted R^2, see Draper and Smith, 1998)*
For a linear regression model with intercept β_0 and p the total number of parameters in $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_{p-1})^t$ the adjusted coefficient of (multiple) determination is defined as

$$R^2_{adj} = 1 - \frac{SSE/(n-p)}{SST/(n-1)} = 1 - (1 - R^2)\left(\frac{n-1}{n-p}\right),$$

where n denotes the sample size as usual.

In our case both the R^2 and the R^2_{adj} only differ by a linear transformation, and since p and n are equal for all features there is no difference of them in the later application. For simplicity only R^2 is used in the application part of this work.

6.1.3 Single Predictor Classification Accuracy

When we relax the assumption that filter models evaluate features without utilizing any classification algorithm, the most intuitive filter comes along by using the classifier only with one feature. The use of each feature individually has two advantages, the first one is that it is comparable to the other univariate filter and as a byproduct we ensure that the duration, while still very long, increases linearly in the number of features because each feature only got evaluated once.

In our case the classifier used for the individual classification is given by the multinomial logistic classifier as defined in Definition 2.3, and the only aspect to clarify is how to evaluate the features with this classifier.

In order to get a measure for the ability of the feature to classify the response every appropriate candidate should take values between zero and one to fit into the framework of the other measures presented in this chapter. For simplicity of the interpretation, it would also be profitable that $\rho = 1$ indicates a perfect, or at least the highest possible classification ability, whereas $\rho = 0$ represents a very poor classification ability of the according feature.

Definition 6.6. *(Single Predictor Classification Accuracy)*
Let $(x_1, y_1), \ldots, (x_n, y_n)$ be a bivariate sample, where $\mathbf{x} = (x_1, \ldots, x_n)^t \in \mathbb{R}^n$ are the values of the feature and $\mathbf{y} = (y_1, \ldots, y_n)^t \in \mathcal{K}^n = \{k_1, \ldots, k_K\}^n$ the according classes. Furthermore let M be a family of classifiers with $\hat{c}^M$ the fitted classifier and $\hat{c}^M(x)$ the estimated classes with argument x and values in $\mathcal{K}$. The Single Prediction Classification Accuracy (spca) ρ^{spca} is defined as

$$\rho^{spca} := \frac{|\{y_i = \hat{c}^M(x_i) | i = 1, \ldots, n\}|}{n}.$$

If the classifier M uses the sample $(x_1, y_1), \ldots, (x_n, y_n)$ also for calibration, then the spca is called in-sample single prediction accuracy.

Obviously the single prediction classification accuracy as defined in Definition 6.6 takes values $\rho^{spca} \in [0, 1]$.

Also the two properties

- $\rho^{spca} = 1 \Rightarrow$ perfect classification
- $\rho^{spca} = 0 \Rightarrow$ no classification ability

are fulfilled by this definition.

Since the Definition 6.6 indeed differs between the in-sample spca and the spca, in the following the context should show whether or not the sample is also used to train the classifier. If not otherwise stated, we are talking usually about the in-sample single prediction classification accuracy.

6.2 Application of the Filter Model

After motivating and defining the different filters, this section provides the results when these filters are applied to the Styrian wine grape data. Since there are several different datasets available in this case (three different filters) there is a methodology required to compare the results from these filters with the same underlying data.

Since the result of a filter model is provided by a list reflecting the rank of the individual feature a method to visualize the results requires some restrictions. More or less standard methods can be used if we visualize or classify the data, i.e. if we only use a few of the best features like the 15 or 40 best ones.

Maybe the most common technique to visualize high dimensional data (but with $p < n$) is the principal component analysis. This technique allows to reflect as much variability of the data as possible and projects the features onto a lower dimensional space, i.e. $\mathbb{R}^2$. While this is not a guaranteed way of receiving further insights into the data, it is at least a chance to see some structure in the data. Therefore, and since the filter models are used again afterwards, this method seems reasonable in order to visualize the individual results and allows some kind of comparability.

Another more numerical method comes up by applying all filters on the same dataset and comparing the results. Here comparing means that the different ranks are used to form an overall ranking. Several choices to form this overall ranking are possible, indeed every function $g : \mathbb{N}^f \to \mathbb{N}$, where f is the number of available filter. Since we want explainable results only a few choices are considered in the following.

The first one is to sum up all ranks, which favor features with generally good scores

(ranks). Secondly, multiply all available ranks, this method focuses more on a (great) single performance. The advantage of the product is that the best features are more likely to end up with a high overall score. Therefore, this approach was used in the following.

To summarize, we use the overall ranking calculated by the product of all ranks to get the best features for each available dataset (c.p. Table 17) where only the corrected data are used.

Afterwards we use the principal component analysis to visualize the features which occur at the top 25 features for the Area- and the Height-dataset. This allows to get a heuristic picture of the classification ability when using the most important filters which are also very robust since it is not important what aspect of the underlying peak is used.

Since these filters are used in Chapter 7 there is a lot of information provided rather for completeness then for further insight at this stage of the analysis.

6.2.1 Application to the Variety-Dataset

For this section it is important to know that there are ten different varieties measured in the variety measurement sequence, which means that ten different classes can be observed. Also important for the later analysis is to know that red and white wine grapes are combined in the data. Therefore, the chemical deviation from each other should at least for some sample element be very well observable.

The Variety-Area-dataset

For the Variety-Area-dataset the individual results of the different filters are presented in Table 18. Here the choice for the features presented explicitly was made by the overall rank as described before. This leads to the occurrence of the best feature according to the epbccm and the spca filter.

Table 18 shows that the ranks for the epbccm filter and for the spca filter seem to be of the same magnitude but the ranks for the R^2 filter are quite lower, which means that the R^2 filter captures other properties or at least values the properties of the individual features in a different way than the epbccm or the spca filter does.

Nevertheless, there are some features which have high classification ability according to one filter but a low one when concerning others.

For example, the feature with mass 440.1656 and retention time 8.67 scores at rank

577 for the epbccm but end up at the fourth place when the spca is used. Another example for the difference of the filters is given by the features with mass 532.1189 and retention time 7.49. Here the R^2 filter ranks this feature on the seventh place but for the spca it is only slightly better than half of the features (place 939).

Mass	**RT**	**epbccm**	**epbccm rank**	**R^2**	**R^2 rank**	**spca**	**spca rank**
488.0927	**7.17**	0.2905	2	0.8590	200	0.6947	3
450.1158	**8.26**	0.2597	42	0.9223	76	0.7053	1
472.0975	**8.26**	0.2938	1	0.9389	52	0.5684	72
372.214	**8.26**	0.2653	23	0.8827	150	0.7053	2
304.0581	**7.61**	0.2670	17	0.9325	60	0.6526	10
516.2564	**8.11**	0.2808	3	0.9179	82	0.5789	60
304.058	8.26	0.2606	38	0.9279	67	0.6421	13
304.0582	**7.17**	0.2635	29	0.8465	230	0.6632	7
169.0732	1.58	0.2395	129	0.9269	68	0.6632	8
538.2384	**8.1**	0.2763	6	0.8892	136	0.5474	92
373.1523	7.59	0.2578	52	0.8309	257	0.6632	6
544.1915	7.48	0.2619	33	0.9028	113	0.6211	25
472.0978	6.87	0.2414	120	0.9397	50	0.6316	17
467.2365	7.73	0.2653	22	0.8615	198	0.6211	26
576.1262	9.01	0.2782	4	0.8323	255	0.5053	156
162.0527	1.3	0.2618	35	0.9058	104	0.5895	47
440.1656	**8.67**	0.2036	577	0.9133	87	0.6632	4
548.1868	7.19	0.2149	429	0.9622	28	0.6316	18
479.236	9.47	0.2224	308	0.8856	141	0.6632	5
373.1525	**6.9**	0.2747	7	0.8482	227	0.5053	139
486.206	7.52	0.2260	258	0.9200	79	0.6421	11
532.1189	7.49	0.2601	39	0.9837	7	0.2947	939
472.192	7.55	0.2739	8	0.8132	289	0.5263	111
484.1913	9.6	0.2303	202	0.8782	154	0.6526	9
440.1655	6.42	0.2675	16	0.8536	214	0.5474	84

Table 18: Results for the Variety-Area-dataset, ordered by the overall rank.

Notice that the usage of the multiplication for finding the overall rank tends to favor features which have a high rank for at least two filters. Therefore, we can observe that the epbccm and the spca filter have some similarities.

When we take a closer look at the spca scoring of the features presented in Table 18 we can observe that they lie between 50% and 70%. This means that by using one feature to classify ten classes we can distinguish 50% to 70% of the samples. Nevertheless, the spca measures the in-sample classification accuracy. Therefore a perfect fit is the overall goal and we see that our later models should definitely contain more

than one feature.

The Variety-Height-dataset

If the height of the peaks is analyzed with the predefined filter, then the outcome is provided in Table 19. As before we can observe that the ranks of the R^2 filter are much lower than the ones caused by the epbccm or spca filter.

When the values of the epbccm filter are compared to the area and the height data a slightly lower scoring for the height data can be observed.

Mass	RT	epbccm	epbccm rank	R^2	R^2 rank	spca	spca rank
472.0975	**8.26**	0.2984	1	0.9532	36	0.5474	109
372.214	**8.26**	0.2683	23	0.8747	193	0.7474	1
450.1158	**8.26**	0.2642	39	0.9282	79	0.6842	2
488.0927	**7.17**	0.2941	2	0.8870	169	0.6000	41
304.0581	**7.61**	0.2658	35	0.9366	64	0.6526	10
516.2564	**8.11**	0.2803	5	0.9209	92	0.5895	55
421.1946	7.1	0.2861	3	0.9087	118	0.5474	107
548.1865	7	0.2136	513	0.9791	12	0.6526	8
368.1444	6.99	0.2690	21	0.8798	182	0.6421	13
373.1525	**6.9**	0.2733	10	0.8723	201	0.6105	28
373.1521	7.31	0.2641	40	0.7945	373	0.6632	6
538.2384	**8.1**	0.2748	8	0.8730	200	0.5895	58
373.1518	7.1	0.2640	41	0.8126	332	0.6526	9
144.042	1.31	0.2721	12	0.9069	126	0.5579	87
372.1754	7.31	0.2509	103	0.7601	454	0.6737	3
486.2072	6.92	0.2713	15	0.8436	258	0.6000	42
354.1653	6.68	0.2307	270	0.8979	153	0.6737	4
208.0943	1.3	0.2706	16	0.9135	104	0.5474	100
354.1651	7.1	0.2749	7	0.8822	178	0.5158	141
373.1517	7.1	0.2649	37	0.8204	312	0.6316	17
304.0582	**7.17**	0.2614	53	0.8480	250	0.6421	15
440.1656	**8.67**	0.2057	609	0.9346	66	0.6632	5
548.1864	6.93	0.2112	543	0.9567	27	0.6421	14
373.1523	6.83	0.2662	31	0.8555	237	0.6105	29
373.1524	7.59	0.2624	47	0.8307	294	0.6316	16

Table 19: Results for the Variety-Height-dataset, ordered by the overall rank.

Another interesting point is given when looking at the overlap of the best, according to the overall rank, features when the area or the height of the peaks is considered. Here ten out of 25 features occur when concerning the area or the height of the peaks.

The corresponding features are bold in Table 18 and in Table 19. Therefore, it seems that there is at least some kind of consistence between the area and the height of the peaks along with their ability for the classification.

Conclusion for the Variety-Dataset

After discussing the results for the individual datasets, we want to take a closer look at the features which occur in both datasets. Therefore, Table 20 contains all features which occur in both the Variety-Area- and the Variety-Height-dataset when the 25 best features, evaluated with the overall rank, are selected.

For simplicity only the scores of the spca filter are provided in the following, since this filter can be interpreted in the clearest way.

As we can see, four of the ten features presented at Table 20 have a higher scoring when the area of the peaks is used. Two features have identical values for the spca filter, and four features lead to higher values of the filter when the height of the peaks is considered. Therefore, the features which occur in both Table 18 and 19 have no clear tendency of higher or lower scoring.

Mass	**RT**	**spca Variety-Area**	**spca Variety-Height**
488.0927	7.17	0.6947	0.6000
450.1158	8.26	0.7053	0.6842
472.0975	8.26	0.5684	0.5474
372.214	8.26	0.7053	0.7474
304.0581	7.61	0.6526	0.6526
516.2564	8.11	0.5789	0.5895
304.0582	7.17	0.6632	0.6421
538.2384	8.1	0.5474	0.5895
440.1656	8.67	0.6632	0.6632
373.1525	6.9	0.5053	0.6105

Table 20: Results for the Variety-Area- and Variety-Height-datasets.

If we want a visual impression of the classification ability of the features in Table 20, Figure 28 provides the first two principal components as vertical and horizontal axis with the colors representing the different varieties. Here the area of the peaks was used, but for the height of the peaks the picture is almost identical.

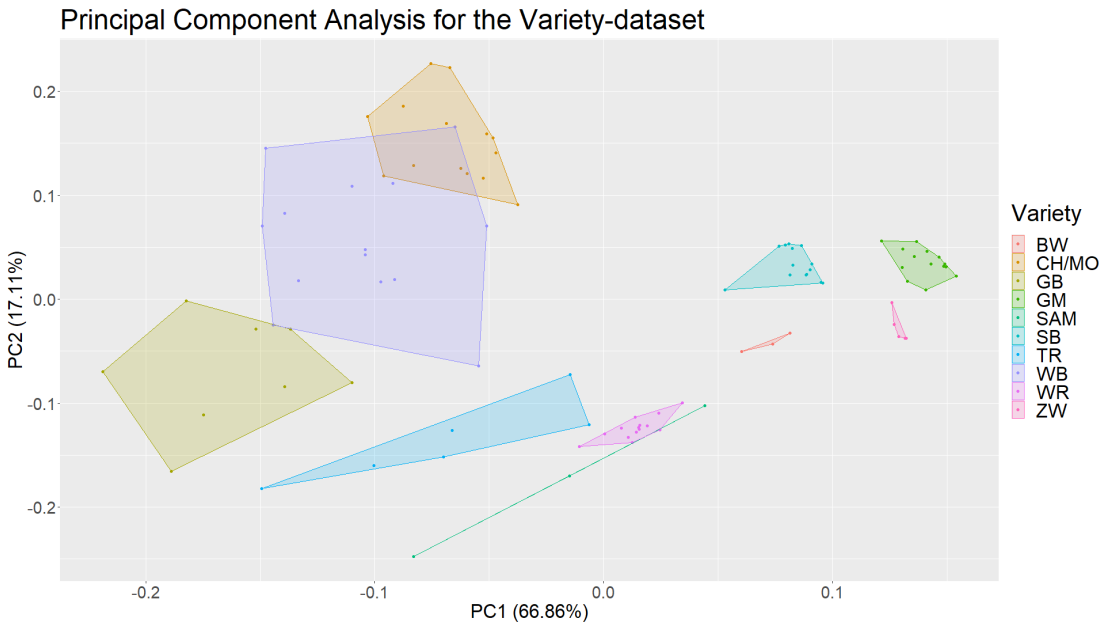

Figure 28: First two PC's by using the area of the features at Table 20.

The advantage of the principal component analysis is that we get rid of redundant information. After all, these filters are univariate and therefore cannot handle correlated data, which provide the same information. We can observe this by the fact that the first two principal components reflect approximately 84% of the total volatility.

After all Figure 28 shows that the classification with the used features should work properly because here clear clusters with only small overlaps were detected.

6.2.2 Application to the Geography-Dataset

After discussing the data generated by the variety measurement sequence, the same type of data but from the measurement sequence designed to analyze the classification ability when the geographical origin is considered remains to be discussed. Therefore, the following subsection provides the same kind of results but according to the geographical measurement sequence.

The Geography-Area-dataset

As mentioned before Table 21 illustrates the three different filters applied on the Geography-Area-dataset containing the values of the rank for the best 25 features, according to the overall ranking system described in the beginning of this chapter. Here for each feature the results of the individual filter along with the rank for the feature are explicitly provided.

The major difference to the results from the variety measurement sequence is given by the fact that here the spca filter seems to evaluate different aspects of the features.

Mass	**RT**	**epbccm**	**epbccm rank**	**R^2**	**R^2 rank**	**spca**	**spca rank**
283.199	**7.15**	0.2678	3	0.4001	9	0.5393	1
517.3612	**19.54**	0.2861	2	0.3977	10	0.5281	2
461.2254	**5.73**	0.2543	9	0.4415	6	0.4719	18
609.2989	**5.66**	0.2603	8	0.4620	4	0.4607	38
99.1045	1.64	0.2984	1	0.4159	7	0.4157	178
247.1303	**8.16**	0.2613	6	0.5790	1	0.3933	398
271.1721	**10.52**	0.2647	4	0.5728	2	0.4045	356
300.209	**15.44**	0.2352	33	0.3776	13	0.4944	8
346.0272	**13.73**	0.2361	31	0.3371	29	0.5056	5
231.141	**7.66**	0.2606	7	0.5529	3	0.4045	292
614.2545	**5.66**	0.2643	5	0.3776	14	0.4382	94
344.0292	**13.73**	0.2339	38	0.3312	32	0.4831	10
342.0321	**13.73**	0.2329	40	0.3281	34	0.4831	9
500.0563	7.97	0.2299	48	0.2884	54	0.5056	6
609.2989	**5.24**	0.2371	29	0.4095	8	0.4494	68
219.126	**11.54**	0.2468	15	0.3928	11	0.4382	102
355.3442	**15.5**	0.2467	16	0.3458	24	0.4494	44
355.3447	**15.7**	0.2452	17	0.3535	21	0.4494	49
329.3288	**15.06**	0.2419	21	0.3400	25	0.4494	41
215.1882	**10.08**	0.2383	25	0.3727	15	0.4494	59
614.2543	**5.23**	0.2322	43	0.3193	40	0.4831	13
279.1471	**11.54**	0.2389	23	0.3684	16	0.4382	70
301.1293	11.54	0.2437	19	0.3834	12	0.4270	147
251.0794	6.62	0.2245	57	0.3082	43	0.4831	14
357.3598	**16.42**	0.2418	22	0.3262	35	0.4494	45

Table 21: Results for the Geography-Area-dataset, ordered by the overall rank.

Taking a closer look at the ranks we can also observe that almost all ranks are high in this case, which means that the filter evaluates in a more consistent way than for the variety measurements.

Nevertheless, there are examples where we can clearly observe that the epbccm and the R^2 filter lead to more similar results than the spca.

For example, the feature with mass 247.1303 and retention time 8.16 is ranked in the first place whereas the R^2 filter is considered as sixth place by the epbccm. In contrast the spca filter only ranks these features at place number 398. So, we can see

that while the contrast between the ranks is not as high as in the Variety-dataset, here a clear difference can be identified.

The Geography-Height-dataset

Table 22 shows the results for the height of the peaks in the Geography measurement sequence and when the ranks of the feature for the different filter are compared, as for the Geography-Area-dataset, the values seem more homogeneous as in the Variety measurement sequence.

Mass	RT	epbccm	epbccm rank	R^2	R^2 rank	spca	spca rank
517.3612	**19.54**	0.2832	1	0.3954	7	0.5281	3
283.199	**7.15**	0.2660	2	0.4010	6	0.5281	2
239.1729	6.63	0.2623	3	0.3844	11	0.5281	1
247.1303	**8.16**	0.2583	6	0.5687	1	0.4494	47
231.141	**7.66**	0.2594	5	0.5478	3	0.4045	266
609.2989	**5.66**	0.2543	10	0.4318	4	0.4270	142
614.2545	**5.66**	0.2620	4	0.3774	14	0.4270	122
609.2989	**5.24**	0.2422	21	0.3932	9	0.4607	38
346.0272	**13.73**	0.2366	34	0.3308	36	0.4944	7
300.209	**15.44**	0.2280	53	0.3534	23	0.4944	8
342.0321	**13.73**	0.2345	39	0.3332	32	0.4831	11
219.126	**11.54**	0.2447	16	0.3863	10	0.4382	91
215.1882	**10.08**	0.2419	22	0.3844	12	0.4494	56
614.2543	**5.23**	0.2271	54	0.3333	31	0.4944	9
344.0292	**13.73**	0.2337	41	0.3324	35	0.4831	12
355.3442	**15.5**	0.2476	15	0.3401	29	0.4494	41
364.1283	15.62	0.2293	49	0.3557	21	0.4831	18
271.1721	**10.52**	0.2575	7	0.5479	2	0.3258	1721
323.1841	11.53	0.2402	24	0.3716	16	0.4494	64
252.1213	1.29	0.2571	8	0.3043	48	0.4494	65
279.1471	**11.54**	0.2402	26	0.3724	15	0.4382	70
357.3598	**16.42**	0.2428	18	0.3303	37	0.4494	42
355.3447	**15.7**	0.2398	28	0.3435	24	0.4494	44
329.3288	**15.06**	0.2401	27	0.3401	28	0.4494	40
461.2254	**5.73**	0.2342	40	0.3948	8	0.4270	106

Table 22: Results for the Geography-Height-dataset, ordered by the overall rank.

Therefore, looking at the individual feature we can observe that the ranks of the filters are closer together. Nevertheless, there are also features where two filter rank them in a similar range but one filter, which is most of the time the spca filter, ranks the same features very low.

For example, the feature with mass equals 231.1410 and retention time 7.66 minutes the epbccm filter ranks it on fifth and the R^2 filter at third place, but the spca ranks it on 266th place.

By taking a closer look on the values of the filters and not the according ranks, we can also observe that for the Geography measurement sequence the results tend to be lower than for the Variety measurement sequence. This is not surprising since from a chemical point of view, it is much easier to analyze the variety than the geographical origin of a wine grape.

As before the features which occur in both Table 21 and 22 are bold and discussed in more detail in the following subsection.

At this point we can see that 21 out of 25 features occur in both tables. This overlap is much larger than the eleven features in the Variety measurement sequence. Up to now we cannot evaluate if this is a good or a bad sign for the later classification.

Conclusion for the Geography-Dataset

As we saw before there were 21 features out of the best 25 features in both the area and the height data. Therefore, it would not lead to a deeper understanding when we analyze the overlapping features in more detail.

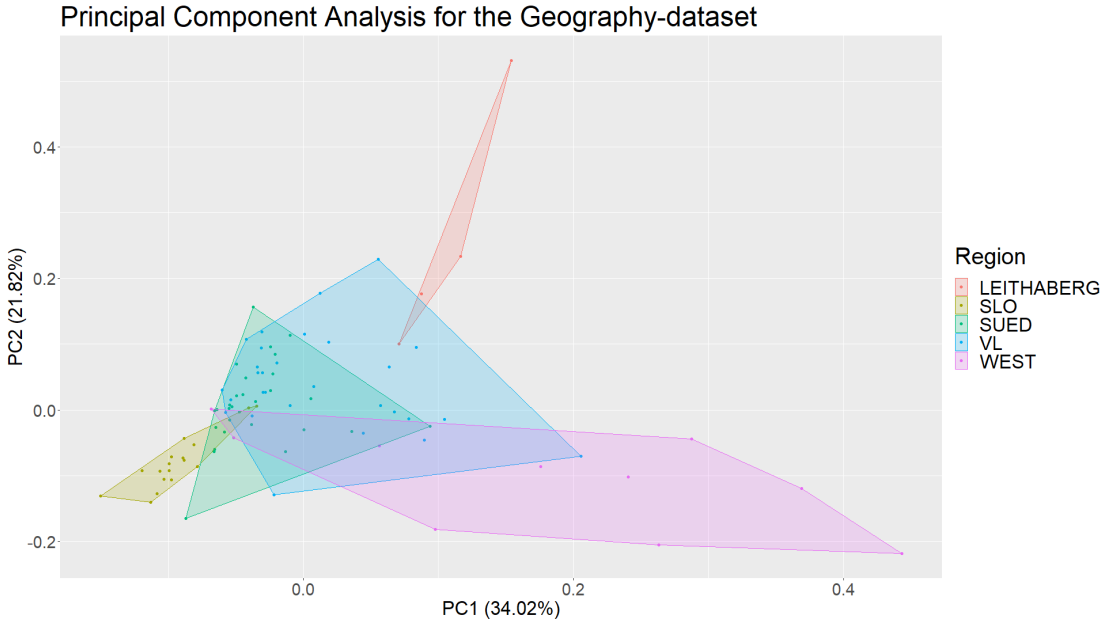

Figure 29: First two PC's by using the area of the features contained in Table 21 and 22.

Nevertheless, Figure 29 provides the first two principal components when the overlapping features are used. The color represents the geographical origin of the according wine grape sample.

Here we can observe three different aspects. The first one can be observed when comparing the clusters in Figure 29 to the ones in Figure 28. Therefore, it seems that the classification problem for the geographical origin is much harder to tackle than the one for the variety. This is also consistent with the chemical intuition of the colleges of the cooperation partner.

By taking a closer look the overlap which can be detected in Figure 29 is quite interesting. We can see that the location VL and SUED overlap the most, which seems reasonable since both are part of the Austrian district Styria. Also, WEST is part of this district and therefore it is not surprising that here an overlap is observable as well. Therefore, it seems as if the overlap indeed corresponds with the geographical distance of the samples.

6.3 Conclusion of Filter Models

As mentioned before only a very heuristic overview was given at this point but two aspects are very interesting up to now.

- The classification ability for both, the Variety- and the Geography-datasets, is at least observable. Since we did not fit models in terms of classical statistical learning were prediction is possible, no further quantification can be provided at this point.
- As mentioned before and also according to the experience of the cooperation partner the classification in terms of the variety should be much easier than the classification of the geographical origin. The results provided before seem to confirm this conjecture.

The next two sections will provide more model based analysis and therefore the possibility of prediction, will allow us to quantify the classification ability along with the specific models.

7 Multinomial Logistic Model and Preselection

As the application of the filter models shows there are indeed features which have a high ability to classify the variety or the geographical origin in our measurement sequences. Due to the fact that there is no statistical model involved, an evaluation of the prediction accuracy of both the in-sample and out-of-sample performance is not possible with this approach. Therefore, only an explorative analysis, as presented above, was possible.

However, this chapter uses the multinomial logistic regression model and the according classifier, as described in Chapter 2, in different variations. The first usage of the multinomial logistic classifier (mlc) is provided in the context of pure wrapper models, as defined in Chapter 3. This represents our starting point and also provides a base line for later variations and approaches.

The operational time of this procedure is quite long which is why the alternative wrapper model with preselection will be introduced and defined in the following. Furthermore, some special cases of filter which have already been discussed in Chapter 6 will be applied to the data at hand.

7.1 Wrapper Model with MLC

The definition of a wrapper model in Chapter 3 is not very detailed, therefore the selected model with its three components (classifier, search method and evaluation criteria) will be specified in the following.

Definition 7.1. *(MLC-Wrapper Model)*
The multinomial logistic classifier - wrapper model (mlc - wrapper model) is a wrapper model with the following specifications:

- *classifier: multinomial logistic classifier*
- *feature search: forward selection*
- *feature evaluation: Akaikes information criterion*

For simplicity the mlc-wrapper model is called wrapper model in the following.

With Definition 7.1 and the following notation we can specify Algorithm 1, described in Chapter 3, which leads to the definition of Algorithm 3, provided below. It is assumed that there are p different features available, which are combined in a design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$. Here n is the sample size as usual.

At this point it is crucial to note which features are used to estimate the parameters for the mlc $\hat{c}$. For the mlc we denote $\hat{c}[\mathbf{x}_1, \ldots, \mathbf{x}_k]$ when the features $\mathbf{x}_1, \ldots, \mathbf{x}_k \in \mathbb{R}^n$ are used for estimating the required parameters. Furthermore, $AIC(\hat{c}[\mathbf{x}_1, \ldots, \mathbf{x}_k])$ represents the value of Akaikes information criterion (c.p. Definition 3.1) where the underlying multinomial model of $\hat{c}[\mathbf{x}_1, \ldots, \mathbf{x}_k]$ is used for the evaluation.

Algorithm 3: MLC - Wrapper Model

$j_1 := \underset{k \in \{1,\ldots,p\}}{argmin} \; AIC(\hat{c}[\mathbf{x}_k])$

$j_2 := \underset{k \in \{1,\ldots,p\} \setminus \{j_1\}}{argmin} \; AIC(\hat{c}[\mathbf{x}_{j_1}, \mathbf{x}_k])$

$i = 2$

while $AIC(\hat{c}[\mathbf{x}_{j_1}, \ldots, \mathbf{x}_{j_i}]) < AIC(\hat{c}[\mathbf{x}_{j_1}, \ldots, \mathbf{x}_{j_{i-1}}])$ **do**

  $i = i + 1$

  $j_i := \underset{k \in \{1,\ldots,p\} \setminus \{j_1,\ldots,j_{i-1}\}}{argmin} \; AIC(\hat{c}[\mathbf{x}_{j_1}, \ldots, \mathbf{x}_{j_{i-1}}, \mathbf{x}_k])$

$return \;\; classifier = \hat{c}[\mathbf{x}_{j_1}, \ldots, \mathbf{x}_{j_{i-1}}]$

In case of the small sample size problem, $n << p$, the forward selection is the only applicable method. This is caused by the fact that for the backward selection it is not possible to fit the first model, since more parameters than observations need to be fitted.

As mentioned before the forward selection seeks for a local minimum of the information criterion. If we would like to ensure a global minimum of the information criterion, we would have to evaluate all combinations of features, which is not reasonable concerning the data. Therefore, it is important to keep in mind that the so found set of features only reflects the optimal choice under some restriction.

If we want to evaluate the performance of the final classifier selected by the wrapper model above, there are several established methods available.

The first one is the in-sample classification accuracy, as defined in Chapter 6. The drawback is that it does not estimate the prediction error when we want to classify new samples. Therefore, it is only a first approach in order to see how the methodology works, since a low in-sample accuracy indicates that the according model is inappropriate.

The second one can be found in the cross-validation method. Here it is possible to estimate the prediction error in a systematic way. In our situation only the leave-one-out cross-validation procedure, also known as Jackknife analysis, is applicable. This comes from the unbalanced design, which would lead, for the regular cross-validation method, to a situation where we have none or only one sample of a specific class in the training data. Therefore, the results would not represent the general data situation and also the used software solution would fail in this scenario.

The last method to evaluate the models would be a bootstrap based approach. In this case the same problem as described in case of the cross-validation method occurs. Also, the required computational resources needed in order to achieve results in a reasonable time would be much higher than for this project available.

7.1.1 Data Standardization

One point worth mentioning is the standardization of the design matrix. Since the fitting step for the classifier is based on numerical optimization, a design matrix where all features have a similar value range could lead to stable results or a shorter run time. Therefore, there are always two versions of the data available and analyzed in the following. The original data, i.e. **Variety-Area-Origin-dataset**, which represents the corrected features (c.p. Chapter 5), and the standardized data, i.e. **Variety-Area-Stand-dataset**. Both versions combined are simply called **Variety-Area-datasets**.

For the standardized datasets we replace the values of the features in the following way. Let $\mathbf{x}_j \in \mathbb{R}^n$ be the j-th column of the design matrix $\mathbf{X}$, then each value $x_{ij}, i = 1, \ldots, n$, is replaced by

$$\tilde{x}_{ij} := \frac{x_{ij} - \bar{x}_j}{\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2}},$$

where $\bar{x}_j$ is the mean of the j-th column,

$$\bar{x}_j := \frac{1}{n}\sum_{i=1}^{n} x_{ij}.$$

Therefore, the standardized data are the ones containing the **centered** and **scaled** values of the corrected features.

After defining the exact methodology and also the available data the following subsections provide the results of the application, where the measurement sequence according to the variety and the geographical origin are discussed separately.

7.1.2 Application to the Variety-Datasets

For the application of the wrapper model on the Variety-datasets two aspects are analyzed. The first one is the in-sample classification accuracy (in-sample ac.), which represents the accuracy when all available samples are used for fitting the model. For these models, also the number of used features is recorded. The second one is the prediction error estimated by the leave-one-out cross validation, here also the run time (loo-cv run time) is provided.

Dataset	**# features**	**classification accuracy**		**run time cross-validation [h]**
		in-sample	**out-of-sample**	
Area-Origin	2	100.00%	84.21%	11.16
Area-Stand	2	100.00%	87.40%	9,81
Height-Origin	2	100.00%	82.11%	11,43
Height-Stand	2	100.00%	86.32%	8,44

Table 23: Results for the wrapper model applied on the Variety-datasets.

As Table 23 shows there are only two features necessary to classify all ten possible classes (in-sample) correctly. The leave-one-out cross-validation method shows that for all types of data in the variety measurement sequence an accuracy of over 82% can be achieved. This shows that there is indeed clear evidence that with this type of chemical analysis the variety can be classified quite well.

The standardization method as described above also seems to slightly increase the out-of-sample classification accuracy. Another interesting fact is that the run time, whenever the standardized data is used, is reduced by approximately 10% to 20%. This can be explained by the fact that the required numerical optimization benefits form a design matrix which contains standardized features.

Remark 7.1.
Notice that the run time here is only presented for comparison purpose and is not optimized for our special case. Also, tasks were not parallelized, which is possible in the case of the leave-one-out cross-validation procedure. Therefore, the provided table entries should only be used to compare the different methods.

Since the final wrapper models generated by using the Variety-Area-Origin and the Variety-Area-Stand-dataset select the same features and only differ by the coefficients. The explicit coefficients for the wrapper model using the original area of the peaks are provided in Table 24.

	Intercept	m=304.0581 rt=7.61	m=428.1654 rt=7.39
CH/MO	-4934.667	0.1979	-0.0061
GB	-11,947.227	0.2602	0.0381
GM	1,458.622	-0.3020	0.0478
SAM	-4,396.428	0.02736	0.0685
SB	1,203.782	-0.1703	0.0455
TR	-6,433.976	0.2632	-0.1069
WB	-6,559.410	0.1321	0.0678
WR	-2,559.158	0.1569	-0.0558
ZW	-1,215.878	-0.4010	0.0676

Table 24: Coefficients for the wrapper model using the area of the peaks.

In this special situation where two features are enough to classify the data perfectly, when using all available information, we can visualize the data by a two dimensional plot. Here each axis represents one relevant feature.

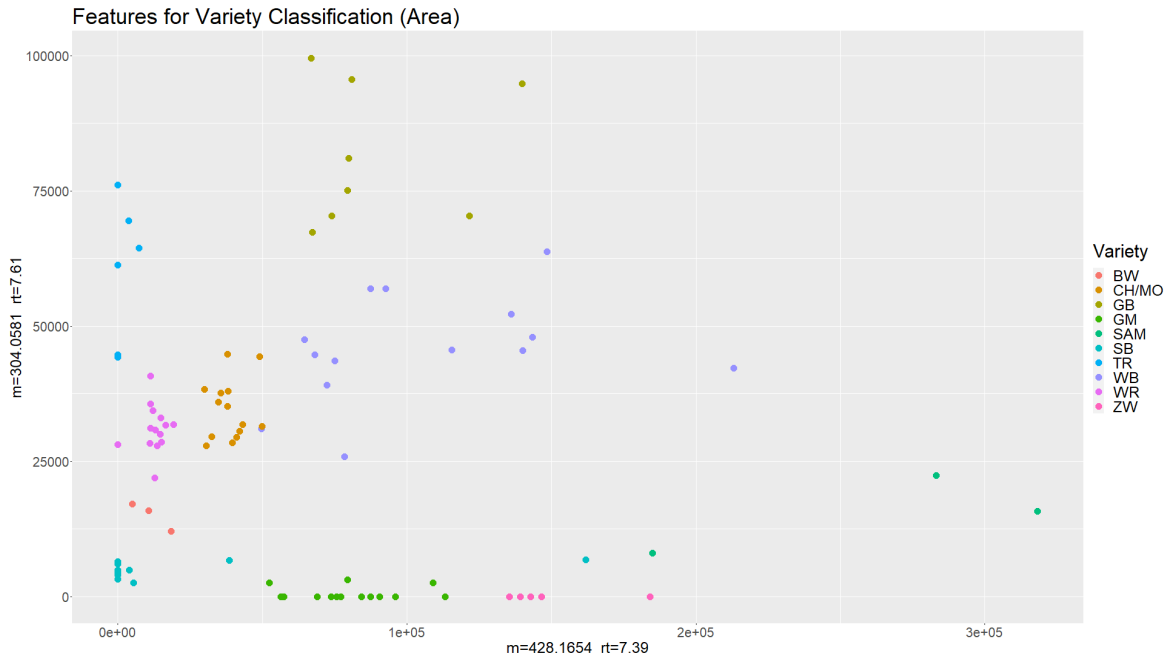

Figure 30: Relevant features for variety classification (Area).

In Figure 30 we can observe that a clear clustering in terms of the varieties exists. Only a few varieties like the GM or the SB seem to have values which are far away from their center points.

If the height of the peaks is used for the modeling, the final wrapper models again coincide in terms of selected features when the original or standardized data are used. As before, the coefficients of the model, using the original data, are provided in Table 25.

	Intercept	**m=304.0581 rt=7.61**	**m=423.21 rt=7.39**
CH/MO	-896.1302	0.1901	-0.0475
GB	-4,888.6459	0.3630	0.0021
GM	957.7213	-0.5965	0.0107
SAM	-1,302.1644	-0.0071	0.0333
SB	445.1296	-0.1397	-0.0018
TR	-2,728.0577	0.4163	-0.2677
WB	-1,713.1770	0.1997	0.0179
WR	-381.1036	0.2451	-0.2549
ZW	-362.1630	-0.7199	0.0541

Table 25: Coefficients for the wrapper model using the height of the peaks.

The results for the height data are very similar to the ones for the area data and do not provide further insights. But after discussing the accuracy of the model and the selected features the following subsection will mention the sensitivity of the methodology.

Sensitivity of the Results

To discuss the robustness of the methodology we again use the results generated by the leave-one-out cross-validation procedure. Here, for each single sample element the whole procedure of feature selection is performed on all data except for the selected sample element. Therefore, the prediction error can be estimated by predicting the class for the selected sample element and comparing it to the observed class.

Another information which can be used from this procedure are the selected features. Here, we get an idea whether or not a feature is stable, which means that it occurs almost in every model, or if the methodology is very sensitive to the according data.

Table 26 provides the features, which are contained in the final models of the leave-one-out cross-validation method, along with the number of models where they were contained. Since the results from the original and the standardized data are very similar only the results for the original ones are provided in the following.

Mass	Retention Time	Absolute Frequency	Relative Frequency
304.0581	7.61	94	98.95%
426.1654	7.39	92	96.84%
440.1656	8.67	2	2.11%
450.1158	8.26	1	1.05%
525.209	7.24	1	1.05%

Table 26: Absolute and relative frequencies of the selected features in the final models (Variety-Area-Origin-dataset).

As Table 26 shows there are two features which occur in almost every model, the first feature was also selected when all samples are used in the estimation step. However, we see that small changes in the data do not have much influence on the feature selection. Therefore, we can say that the feature selection using the wrapper model is quite robust.

Together with the results discussed before, it seems that the wrapper model is a good choice for the classification of the variety. Here we observed that the data standardization indeed improves the classification accuracy in terms of the out-of-sample prediction from around 82% up to over 86% and also the methodology itself seems to be quite robust in terms of the selected features.

7.1.3 Application to the Geography-Datasets

Analogous to the variety measurement sequence the data for the geographical measurement sequence is analyzed in this section. A summary containing the in-sample and out-of-sample accuracy is given in Table 27.

Dataset	# features	classification accuracy		run time cross-validation [h]
		in-sample	out-of-sample	
Area-Origin	4	84.27%	66.29%	10,59
Area-Stand	4	84,27%	64,04%	9,27
Height-Origin	5	87,64%	64,04%	11,12
Height-Stand	5	87,64%	55,06%	10,30

Table 27: Results for the wrapper model applied on the Geography-datasets.

For this section the quantity of interest is the geographical origin also called region and there were five different locations available. These five regions split up into two countries (Austria and Slovenia) and four different DAC-regions for the Austrian samples.

Also keep in mind that, as mentioned before and verified by the first results of the

filter model, the difference in terms of chemically measurable variation should be much lower than for the variety setting.

As Table 27 shows, more complex models are required, and even those models do not result in an tremendous in-sample classification accuracy. Also noticeable is the fact that the leave-one-out cross-validation performs quite bad in comparison to the results of the variety. However, this is not really surprising if again the complexity of the chemical task is considered.

One interesting point worth to be mentioned is that for the Geography-datasets the standardization of the data decreases the performance of the wrapper model, when the prediction accuracy estimated by the leave-one-out cross-validation method is considered as performance measure.

This could have several reasons. One is that the different classes stratify the data in a more sensitive way, which means that by the transformation of the features small changes get lost, or at least are reduced. This will be discussed later when the sensitivity and the leave-one-out cross-validation procedure is discussed in more detail.

First, we want to take a look at the wrong classified samples. Here all available samples were used to fit the model. The confusion table provides the true and the predicted classes and a perfect fit would be represented by non-zero values only in the diagonal.

Predicted \True	**LEITHABERG**	**SLO**	**SUED**	**VL**	**WEST**
LEITHABERG	4	0	0	0	0
SLO	0	17	0	0	0
SUED	0	0	25	10	0
VL	0	0	4	19	0
WEST	0	0	0	0	10

Table 28: Confusion table for the Geography-Area-Origin and Geography-Area-Stand-dataset.

As Table 28 shows the regions which are not distinguished correctly are SUED and VL. This coincides with the results of the filter models in Chapter 6. Therefore, we again observe that these regions are very close in terms of their the chemical measurements.

For reasons of completeness the significant features for all models along with the affiliation for the models are provided in Table 29.

Mass	Retention Time	Area-Origin	Area-Stand	Height-Origin	Height-Stand
517.3612	19.54	Yes	Yes	Yes	Yes
283.199	7.15	Yes	Yes	No	No
271.1721	10.52	Yes	Yes	No	No
301.2979	14.31	Yes	Yes	No	No
215.1886	8.65	No	No	Yes	Yes
231.141	7.66	No	No	Yes	Yes
687.5127	15.92	No	No	Yes	Yes
252.1213	1.29	No	No	Yes	Yes

Table 29: Features in the final wrapper models.

As Table 29 shows there is only a small overlap (i.e., only one feature) of features in the final wrapper model when the different data belonging to the geography measurement sequence are considered. This indicates that the classification problem of the geographical origin is not that easy to handle with the methodology developed so far.

A visualization as in Figure 30 is not possible since every model used more than two or three features.

Sensitivity of the Results

As for the Variety-datasets the leave-one-out cross-validation method for the Geography-datasets was used to analyze the sensitivity of the results. Here we observed that much more features at least appeared in one model during the cross-validation procedure. Since the results for the different data were quite similar, only the ones when using the Geography-Height-Stand-dataset are provided in the following.

Therefore, Figure 31 shows the relative frequency of the feature on the vertical axis and on the horizontal axis the rank of the feature, when they are ordered by their relative frequency, is provided. This means that for example the point with rank 10 on the horizontal axis represents the feature with the 10th highest relative frequency. Only features, or respectively their rank, which appear at least once are shown in the graphic.

In Figure 31 we can observe that there are a few features which occurs in almost all final models, but a much wider number of features occurs only a few - or in extreme cases - only one time. This means that the wrapper models used for the classification of the geographical origin are much more sensitive to small changes in the data than for the variety classification.

This could also be a reason why the results for the standardized data are not as

good as the original data. Here we see that more research in terms of models or even data preparation is required to capture the complexity of this issue.

After all we can say that for the classification of the geographical origin there is clear evidence that the chemical analysis has the potential to be a reasonable methodology, but as mentioned above further development in terms of modeling and or data preparation is required to provide satisfying results.

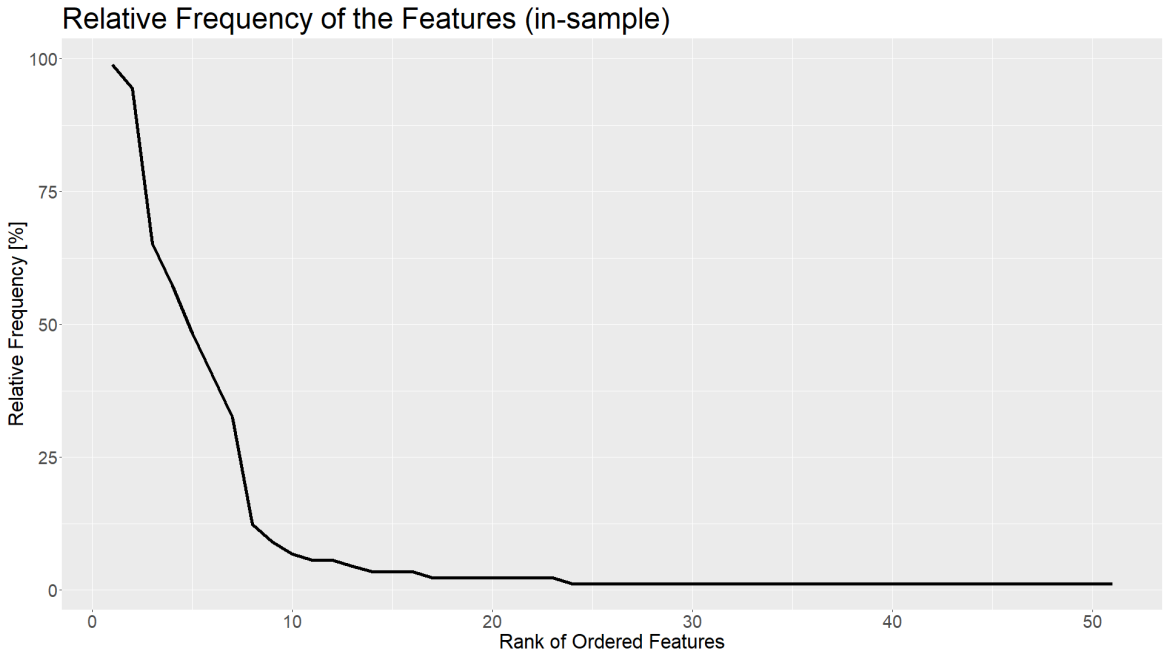

Figure 31: Relative frequency of the features in the final model (Geography-Height-Stand-dataset).

After getting some kind of base line for every classification problem and according dataset the following section describes the concept of preselection for wrapper models.

The application of this methodology is also available in later parts of the section and we will see that an improvement in terms of accuracy, estimated by the leave-one-out cross-validation method is possible in some cases.

7.2 Wrapper Model with Preselection

The general idea presented in this section is based on the combination of a filter model with the application of a wrapper model on a reduced number of features. For a general framework of wrapper models with preselection it is assumed that the available

input data consists of a design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ and a response vector $\mathbf{y} \in \mathbb{R}^n$.

This input data is used to perform a filter model, afterwards the rank of the features is described by the permutation $\tau : \{1, \dots, p\}^n \to \{1, \dots, p\}^n$. This means that $\tau(1)$ is the rank for the first feature, which is the first column of the design matrix $\mathbf{X}$. The inverse of the function τ at a point i, i.e. $\tau^{-1}(i)$, describes the location, or column, of a feature with rank i, $i \in \{1, \dots, p\}$, in the design matrix. With this function we can rearrange the design matrix $\mathbf{X}$ by defining

$$\tilde{\mathbf{X}} := (\mathbf{x}_{\tau^{-1}(1)}, \dots, \mathbf{x}_{\tau^{-1}(p)}).$$

For the wrapper model only the best, according to the filter model, m features are used. By definition these are the first m columns in the rearranged design matrix $\tilde{\mathbf{X}}$ and in the following summarized as $\tilde{\mathbf{X}}_m \in \mathbb{R}^{n \times m}$. For a better understanding of the procedure, Figure 32 provides an overview of the general framework of wrapper models with preselection.

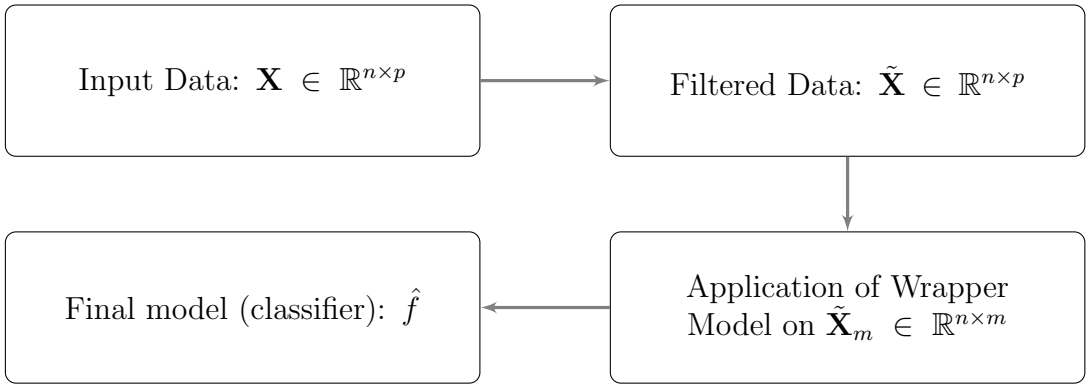

Figure 32: A general framework of wrapper models with preselection.

With this framework the wrapper model with preselection contains of three parts:

- the filter model for the preselection and therefore an underlying filter,
- the wrapper model for the final model search, along with a classifier and a search method,
- the hyperparameter m which needs to be specified.

The main advantage of the preselection is given by the parameter m which is able to control the run time. The restriction of using the best m features, according to the filter model, should allow similar performances compared to the unrestricted case. However, by using the preselection the run time increases linearly in the number of available features, which makes this method useful in situations where the number of available features is quite large.

Remark 7.2.
Since only univariate filter are used, the filter step could be parallelized. Nowadays when multi-core computing is indeed very common this could be an additional run time boost for these types of models.

In the following the results for the wrapper model with preselection, applied on all available datasets and all three filters defined in Chapter 6, are compared. The wrapper model used in the following also uses the mlc as classifier but in contrast to before here the backward selection will be utilized.

7.2.1 Application to the Variety-Dataset

Since the results of the pure wrapper model are quite good, especially when applied on the Variety-dataset, there is no intention to achieve a remarkably better performance. However, always keep in mind that for datasets with very large features this method is faster, even if it is not optimized. Also notice that for the variety classification problem the number of features in the pure wrapper model is quite low. Therefore, this is indeed not the best-case scenario for the preselection application.

In-Sample Classification Accuracy

As for the pure wrapper models, the first quantity we want to discuss is the in-sample classification accuracy. Therefore, Figure 33 provides the in-sample classification accuracy depending on the underlying dataset (shape of the points), the filter used for the preselection (color of the points) and the maximum number of available features m (horizontal axis).

Here we can observe two major aspects. The first one is the general tendency that a higher number for the maximum number of available features, m, corresponds to a higher in-sample classification accuracy. This behavior is clear since with the parameter m we control the maximum number of features used. Therefore, with a higher number m we "allow" the model to be more complex, which is the reason why the in-sample classification accuracy tends towards 100% in all situations.

The second point of view refers to the performance of the different filters. Here we see that the R^2 filter starts with very poor results and needs a larger maximum number of available features to achieve an in-sample classification accuracy which is of the same magnitude as the results when the epbccm or the spca is applied.

Figure 33: In-sample classification accuracy for the Variety-Area and the Variety-Height-datasets.

Another interesting aspect is the number of features in the final model. Here all available information is used to train the classifier. Therefore, Figure 34 uses the same method (shape, color, etc.) to provide the number of finally used features for the different settings.

To provided a reference value for the model complexity, measured by the number of included features, the natural upper bound of the maximum number of available feature is used. Therefore, in Figure 34 the first median, in the geometric sense, shows this upper bound.

In Figure 34 a similar picture as for the in-sample classification accuracy is observable. Here we can verify that the R^2 filter only performs similar to the other filters, in terms of the in-sample classification accuracy, when the maximum number of available features is high. In this case much more features are selected for the model, which obviously increases the in-sample classification accuracy.

So far we can conclude that the R^2 filter does not perform as good as the other filters, and in general the preselection method results in much more complex models, considering the number of selected features. Furthermore, most of the settings result in a lower in-sample classification accuracy compared to the pure wrapper models

(100%).

Figure 34: Number of finally selected features for the Variety-Area- and the Variety-Height-datasets.

Leave-One-Out Cross-Validation

For estimating the prediction error, or classification accuracy when new samples are classified, we again use the leave-one-out cross-validation method. Beside the comparability of the results for the pure wrapper models the reasons for this choice as a performance measure are identical to the ones mentioned before.

Figure 35 provides the classification accuracy of the leave-one-out cross-validation procedure, where the different settings and circumstances are identified by the color or by the shape of the points. Additionally, the results of the pure wrapper models are visualized by horizontal lines with different line types.

In Figure 35 we observe that again the preselection with R^2 performs really poor compared to all other methods. Beside this we get a very homogeneous picture. The major part of the results are between a classification accuracy from approximately 75% up to 85%. This means that the preselection method can achieve similar results as for the pure wrapper model.

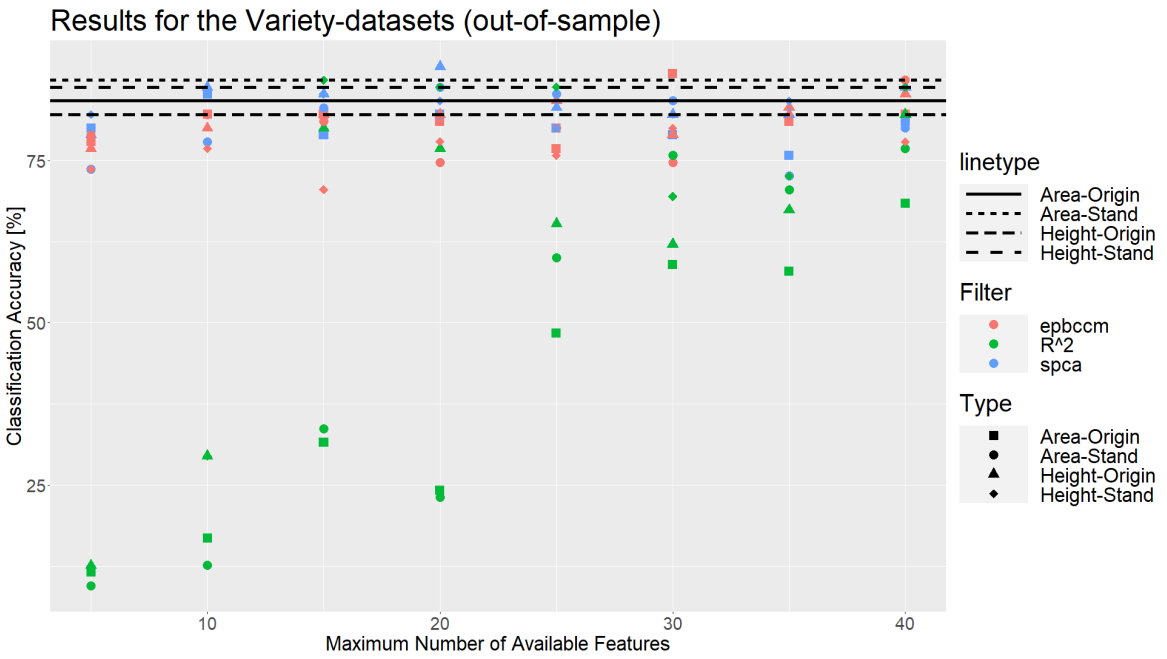

Figure 35: Leave-one-out cross-validation for the Variety-Area- and the Variety-Height-datasets.

One drawback can be observed when a specific setting (i.e. a dataset, a filter and a maximum number of available features m) is chosen and the sensitivity of the feature selection is considered. Therefore, the spca filter was applied on the Variety-Height-Origin-dataset and a maximum number of 20 features was fixed. Analog to the discussion of the sensitivity for the feature selection using the pure wrapper models, Figure 36 shows the frequency of the finally chosen features during the leave-one-out procedure.

Here again the relative frequency for the individual features is provided on the vertical axis and the horizontal axis shows the rank of the features, when they are ordered by their relative frequency.

In Figure 36 we see that the selected features are not that clear for this method compared to the pure wrapper models. Here the feature which occurs in most of the final models only achieves a relative frequency of slightly less than 40%. Therefore, we can conclude that this methodology is much more sensitive against changes to the data. This could be caused by the sensitivity of the filter according to changes of the data which result in different sets of features for the wrapper model and obviously different features in the final model.

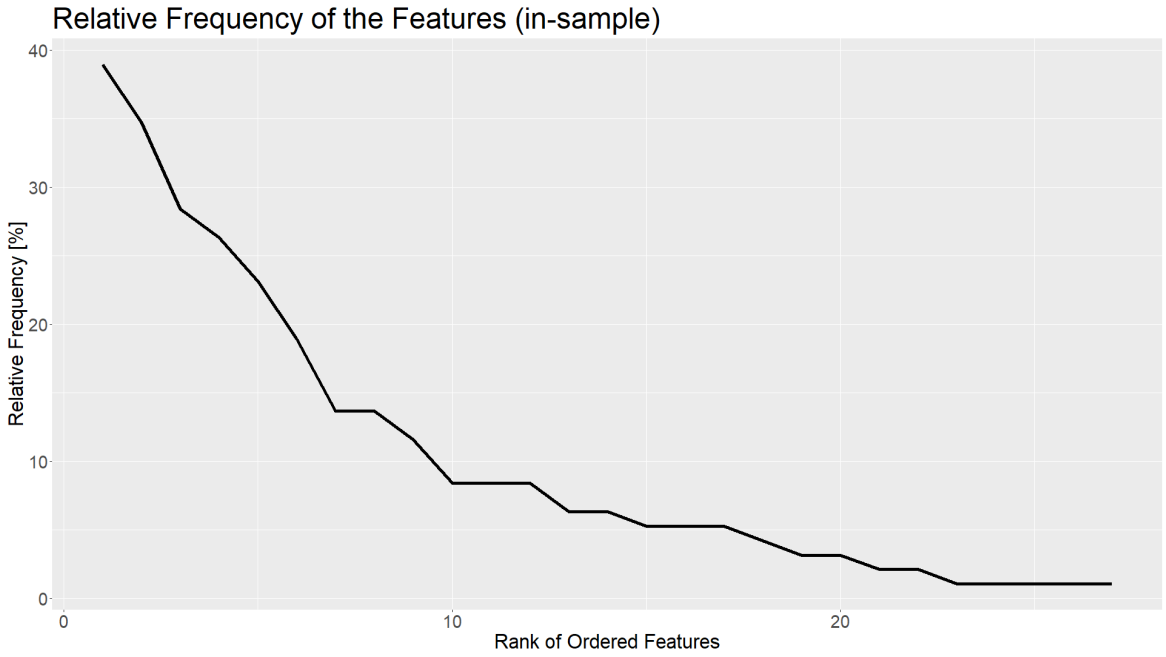

Figure 36: Relative frequency of the features in the final model (Variety-Height-dataset, spca filter, at most 20 features available).

This example shows that we can indeed achieve an out-of-sample classification accuracy, estimated with the leave-one-out cross-validation method, of 89.47% which is slightly higher than for the best results concerning the pure wrapper model. However, the drawback that the methodology has is that it seems to be very unstable and data dependent.

Conclusion for the Variety-Dataset

After discussing the performance of the wrapper model with preselection and illustrate the abilities and also limits of this method, applied on the Variety-datasets, one aspect has not been discussed in detail before. This is the aspect of run time for this methodology. Therefore, Figure 37 provides the different run times for performing the leave-one-out cross-validation procedure and as we can see in almost all cases a run time of less than three hours was required.

Furthermore, we can also observe that the spca needs more time to do the calculations than the epbccm or the R^2 filter. This is not surprising since the spca needs to numerically solve an optimization problem, whereas the other two filters can be calculated directly.

Remark 7.3.
Formally the R^2 filter also solves an optimization problem, but for classical linear regression this problem can be solved analytically.

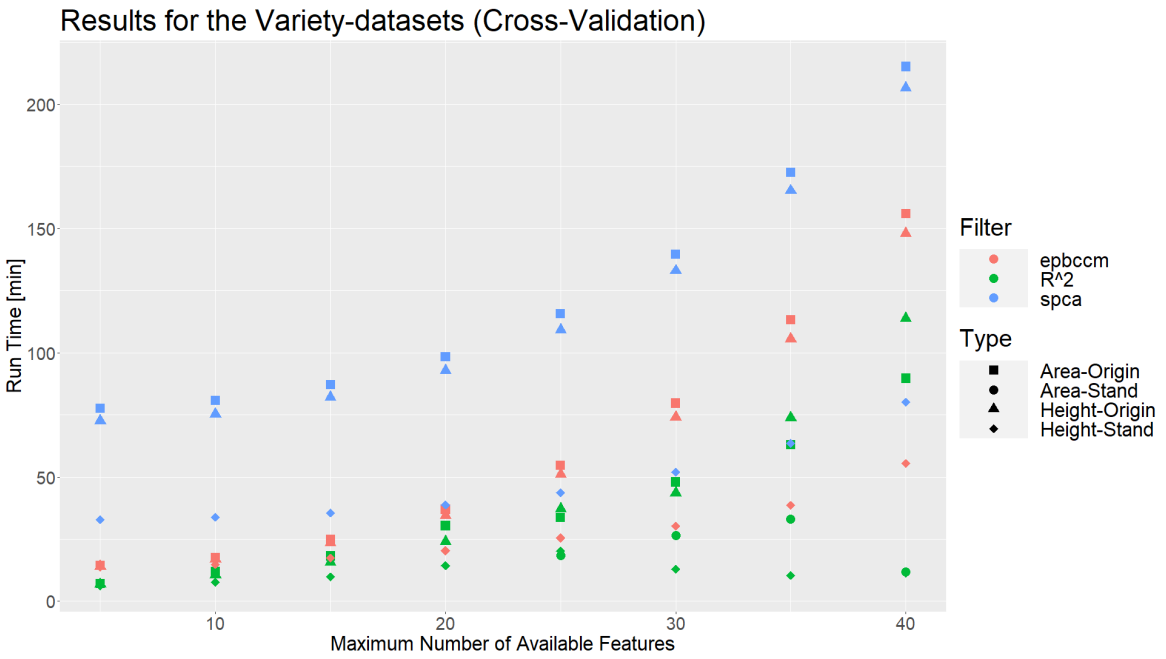

Figure 37: Run time comparison for the Variety-datasets.

After all we can conclude that for the Variety-datasets the method of preselection can improve the results, or at least produce similar results in a much shorter run time. However, the drawback of this method is that it is very sensitive to the data and in general not really robust concerning the Variety-datasets.

7.2.2 Application to the Geography-Dataset

As we could see for the Variety-datasets, the wrapper model with preselection can achieve quite reasonable results but in the previous setting the pure wrapper model had already achieved quite good results either. In this section the wrapper model with preselection is applied to the Geography-datasets for which we already know that the general classification ability is not as good as for the variety classification problem.

In-Sample Classification Accuracy

Analogous to the Variety-datasets we start with a brief discussion of the in-sample classification accuracy. Figure 38 provides the results where again the different filter

and the data used are visualized by the color and the shape of the points. Additionally, the in-sample classification accuracy of the pure wrapper models is also provided by horizontal lines with different line types.

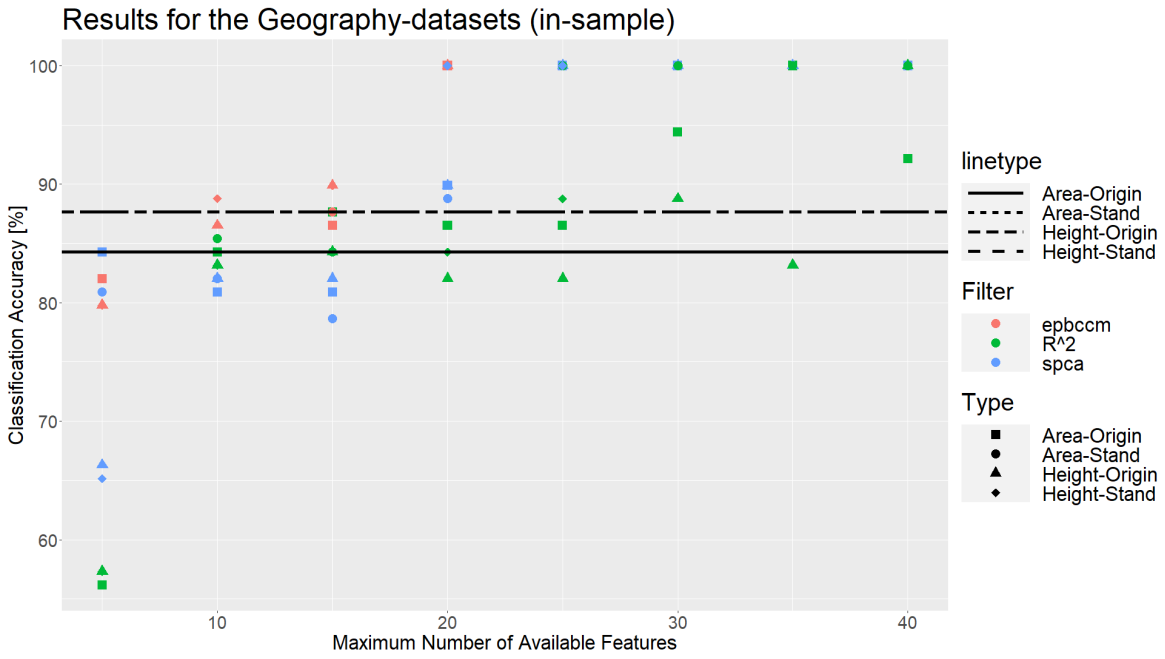

Figure 38: In-sample classification accuracy for the Geography-Area- and the Geography-Height-datasets.

As mentioned before we can observe that the maximum number of available features directly corresponds to the in-sample classification. This means that for this setting we can indeed reach 100% for the in-sample classification accuracy.

Nevertheless, the required number of selected features, as shown in Figure 39, verifies that we can only achieve this high in-sample classification accuracy with very complex models, compared to the four or five features selected by the pure wrapper model.

Furthermore, an interesting point worth to be mentioned is that for the Geography-datasets the number of features in the final model differs widely between the different filters used for the preselection. Especially when the maximum number of available features is equal or greater than 20.

Here we can see that the classification problem for the geographical origin of the grape samples is indeed more difficult than for the variety. This also coincides with the aforementioned results from before and the conclusion in Chapter 6. Also, the fact

that the R^2 filter again produces the most complex models indicates that this filter is not really practicable.

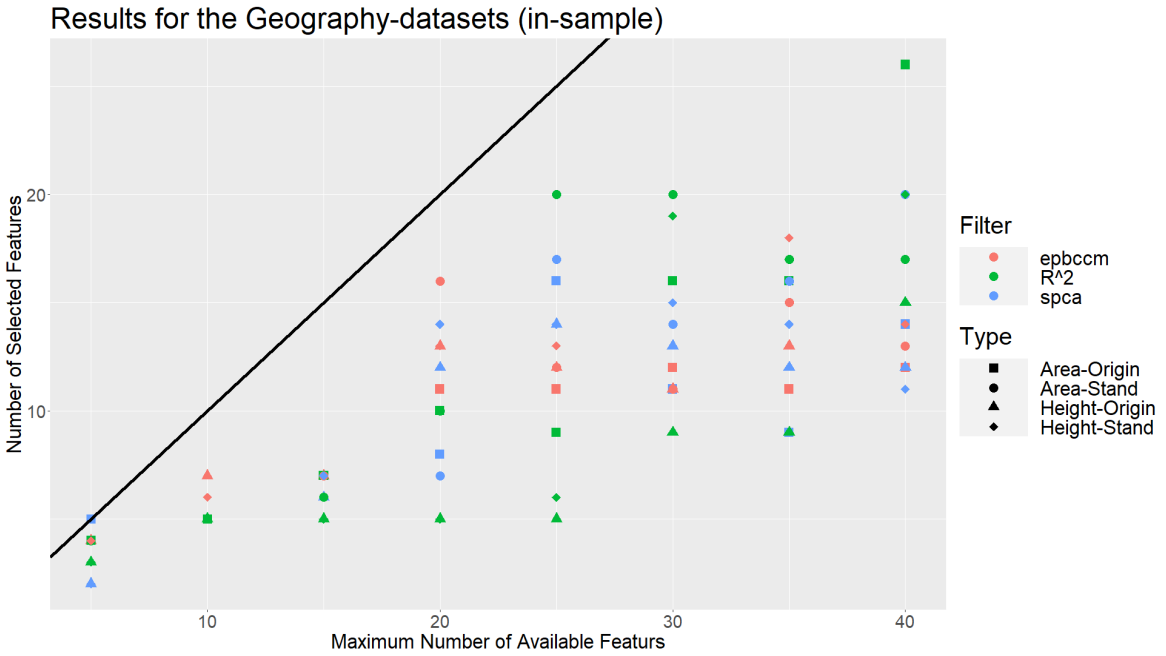

Figure 39: Number of finally used features for the Geography-Area- and the Geography-Height-datasets.

From this first analysis we receive a similar picture as before but in this case the direct control of the maximum number of available features and the backward selection method of the wrapper model allows us to have models which are able to perfectly separate our five different geographical origins when all samples are used to fit the model.

Leave-One-Out Cross-Validation

Since the in-sample classification accuracy is only a first and not a final performance measure, the prediction error or out-of-sample classification accuracy, estimated by leave-one-out cross-validation, is provided in Figure 40 as for the Variety-datasets.

In Figure 40 we can observe two different findings. The first one is that in most cases the wrapper model with preselection does not perform as well as the pure wrapper model which is why we actually lose performance by using this preselection method.

The other one is that a larger maximum number of available features leads to a decrease of the classification accuracy, which indicates again that the method is quite

sensitive and not robust against changes of the data. To see this, Figure 41 shows, as for the Variety-datasets before, the relative frequency of the finally chosen features during the leave-one-out cross-validation procedure.

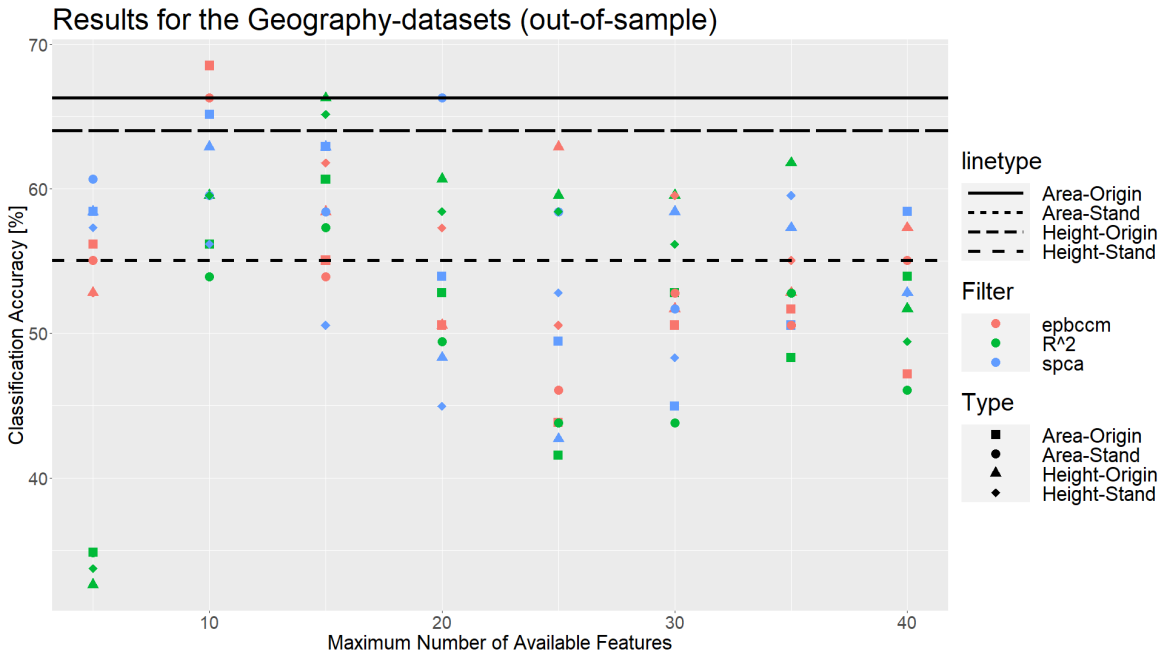

Figure 40: Leave-one-out cross-validation for the Geography-Area- and the Geography-Height-datasets.

The model chosen to be analyzed in more detail is specified by using the Geography-Area-Origin-dataset with the epbccm as filter for preselection and a maximum number of available features of 10. This model achieves with its 68.54% the highest out-of-sample classification accuracy, which is even higher than the 66.29% achieved by the pure wrapper model applied on the same data.

Figure 31 shows that when a maximum number of 10 features is allowed for every step of the leave-one-out cross-validation, only one specific feature is selected in each step and 15 different features are selected, at least once, during the procedure.

Compared to the results from the Variety-dataset this seems quite stable but caution is required since around five features are selected in the final model (c.p. Figure 39) which means that an overlap of the features is much more likely than in the case of the Variety-datasets were less features where selected in every step of the leave-one-out cross-validation procedure.

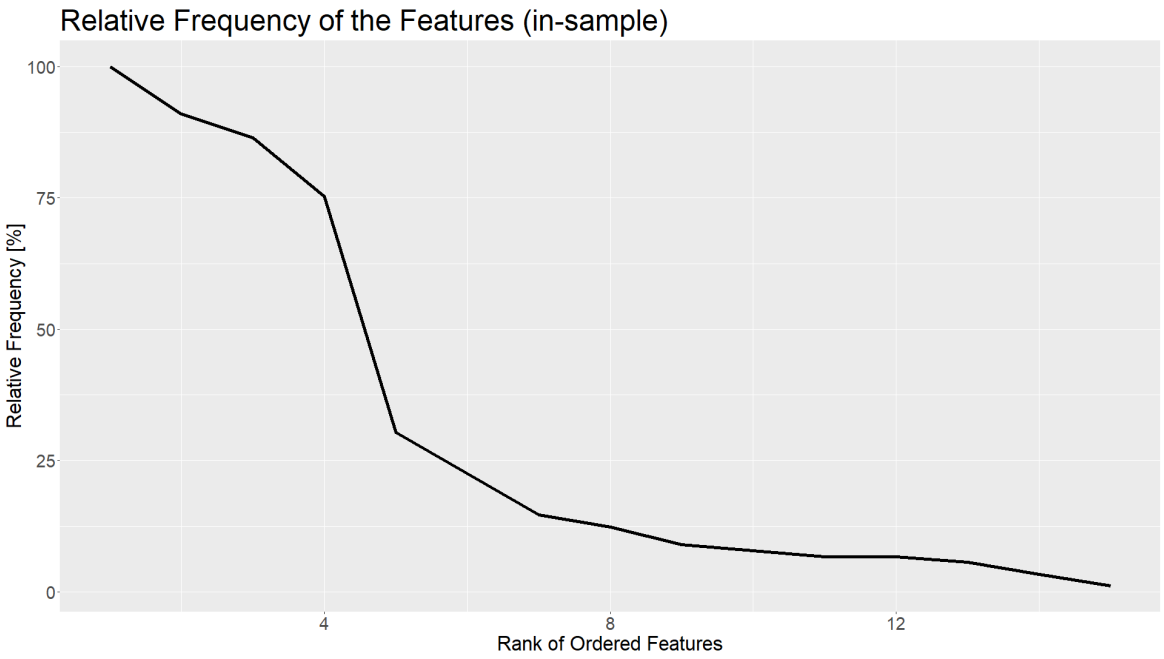

Figure 41: Relative frequency of the features in the final model (Geography-Area-dataset, epbccm filter, at most 10 features available).

Conclusion for the Geography-Dataset

The results of the Geography-datasets are very similar compared to the ones of the Variety-dataset. This means that the general methodology works very well in specific settings. For the Geography-datasets we saw that a smaller maximum number of available features for the wrapper model performs better in terms of the out-of-sample classification accuracy but a higher maximum number of available features increases the in-sample classification accuracy up to 100%.

As for the Variety-datasets we observed for the Geography-datasets that the methodology is very sensitive to small changes to the data which makes it difficult to receive clear, stable and well-proven comparable results.

To conclude this section, we can say that the main idea of preselection works and achieves results which are in some cases very close or even better than the ones of the pure wrapper models. But we also saw that the R^2 filter performs very badly and the reduction of the run time comes at the cost of instability.

To compare the run time for the Geography-datasets, Figure 42 provides the same for the leave-one-out cross-validation procedure under different settings.

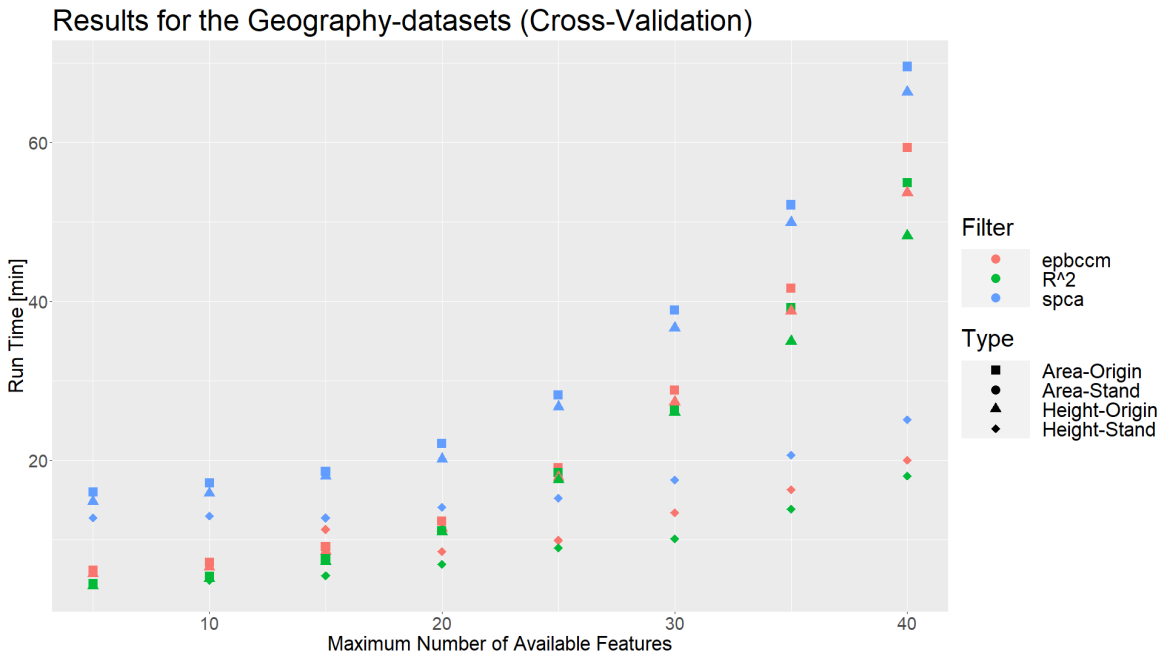

Figure 42: Run time comparison for the Geography-Area- and the Geography-Height-datasets.

7.3 Conclusion of Wrapper Models

In this chapter we can confirm the findings of Chapter 6 where we saw that the classification according to the variety of the grapes is much easier than the classification according to the geographical origin. With the usage of the wrapper models, we found a methodology which is able to achieve an out-of-sample classification accuracy of approximately 85% for the variety and around 65% for the geographical origin.

However, the usage of standardized data does not lead to clearly better results and in the case of the Geography-datasets it decreases the out-of-sample classification accuracy.

The preselection method shows the possibility of reducing the run time but is not fully developed at this point. Therefore, we observed instability in the selection procedure and further research in terms of used filters and an appropriate choice for the maximum number of available features m would be required to get a working alternative. Nevertheless, we saw that with this procedure the out-of-sample classification accuracy indeed can be increased with certain settings.

To conclude this chapter, we can say that so far the results are very promising and that the chemical analysis is capable of stratifying the samples according to their variety. However, for the classification of the geographical origin, up to now, we can say that further work would be required to achieve satisfactory and practicable results.

8 Multinomial Logistic Model with Penalization

As we have already observed the wrapper models generate quite reasonable results, at least for the classification of the variety, but it also requires a lot of computational resources and therefore its run time is quite long. In order to reduce the amount of time, the method of preselection has not worked out in a satisfactory way which is why the application of embedded models is provided in this chapter. Here, we will use the lasso regularization because of its properties and widely usage in the case of high dimensional classification problems ($p >> n$).

Before discussing the results for the Variety- and the Geography-datasets separately, we will formulate and describe the optimization problem in our case and also mention the used software for training and testing of the according classifier. As a last point, a short discussion of the evaluation methods is provided, which basically coincides with the ones of the wrapper model.

For the formulation of the optimization problem in the framework of embedded models we start with Equation (3.4) from Chapter 3, stated in Definition 3.4 as

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^q}{\operatorname{argmin}} \; V(\mathbf{y}, \boldsymbol{\beta}, \mathbf{X}) + \alpha \times \text{penalty}(\boldsymbol{\beta}),$$

where $\mathbf{y}$ is the response vector, $\mathbf{X}$ the design matrix and $\boldsymbol{\beta}$ the parameter vector.

Here, we see that the loss function $V(\mathbf{y}, \boldsymbol{\beta}, \mathbf{X})$ and the penalization term penalty($\boldsymbol{\beta}$) need to be specified and additionally to the parameter vector $\boldsymbol{\beta}$, the regularization parameter α needs to be estimated as well.

Due to the fact that we want to use the multinomial logistic classifier (mlc) as classifier (c.p. Definition 2.2) for the embedded model a natural definition of the loss function is the negative log-likelihood function according to the underlying multinomial distribution. If we want to formulate this properly, we have to start with the logit link as defined in Chapter 2 by

$$\log \frac{p_{ik}}{p_{i1}} = \mathbf{x}_i^t \boldsymbol{\beta}_k \qquad \Leftrightarrow \qquad p_{ik} = p_{i1} \exp(\mathbf{x}_i^t \boldsymbol{\beta}_k) \qquad\qquad k = 2, \ldots, K.$$

With the restriction that $\sum_{k=1}^{K} p_{ik} = 1$ for $i = 1, \ldots, n$, which is clear by the definition of the multinomial distribution, we get an explicit expression for the probability of the reference class in the ith sample (p_{i1}) by

$$1 = \sum_{k=1}^{K} p_{ik} = p_{i1} + \sum_{k=2}^{K} p_{i1} \exp(\mathbf{x}_i^t \boldsymbol{\beta}_k) = p_{i1} \left(1 + \sum_{k=2}^{K} \exp(\mathbf{x}_i^t \boldsymbol{\beta}_k) \right)$$
$$\Rightarrow p_{i1} = \frac{1}{1 + \sum_{k=2}^{K} \exp(\mathbf{x}_i^t \boldsymbol{\beta}_k)} \qquad i = 1, \ldots, n.$$

With this expressions for p_{ik} where $k \in \{1, \ldots, K\}$ and $i \in \{1, \ldots, n\}$ the equation for the relevant part of the log-likelihood of a multinomial distribution and therefore the relevant part for the loss function can be written as

$$V(\mathbf{y}, \boldsymbol{\beta}, \mathbf{X}) := -l(\boldsymbol{\beta}|\mathbf{y}, \mathbf{X}) = \sum_{i=1}^{n} \sum_{k=1}^{K} y_{ik} \log(\frac{1}{p_{ik}}) = \sum_{i=1}^{n} \left(y_{i1} \log(\frac{1}{p_{i1}}) + \sum_{k=2}^{K} y_{ik} \log(\frac{1}{p_{ik}}) \right)$$
$$= \sum_{i=1}^{n} y_{i1} \log \left(1 + \sum_{k=2}^{K} \exp(\mathbf{x}_i^t \boldsymbol{\beta}_k) \right) + \sum_{i=1}^{n} \sum_{k=2}^{K} y_{ik} \log \left(\frac{1 + \sum_{k=2}^{K} \exp(\mathbf{x}_i^t \boldsymbol{\beta}_k)}{\exp(\mathbf{x}_i^t \boldsymbol{\beta}_k)} \right).$$

Notice that by using the notation of Chapter 2 we describe the parameter vector $\boldsymbol{\beta}$ by $\boldsymbol{\beta} = (\boldsymbol{\beta}_2^t, \ldots, \boldsymbol{\beta}_K^t)^t$, where $\boldsymbol{\beta}_k = (\beta_{0k}, \ldots, \beta_{pk})^t$ is the parameter vector according to the class k. This is also important for the specification of the penalization term as described in the following.

Since the lasso penalization favors sparse models and is widely used for high dimensional problems we also use this penalization as defined in Chapter 3 for this work. Therefore, we can write the penalization term as

$$penalty(\boldsymbol{\beta}) := ||\boldsymbol{\beta}||_1 = \sum_{m=1}^{q} |\beta_m| = \sum_{k=2}^{K} \sum_{j=0}^{p} |\beta_{jk}| = \sum_{k=2}^{K} ||\boldsymbol{\beta}_k||_1,$$

which leads to the final formulation of the cost function for our optimization problem as

$$\sum_{i=1}^{n} y_{i1} \log \left(1 + \sum_{k=2}^{K} \exp(\mathbf{x}_i^t \boldsymbol{\beta}_k) \right) + \sum_{i=1}^{n} \sum_{k=2}^{K} y_{ik} \log \left(\frac{1 + \sum_{k=2}^{K} \exp(\mathbf{x}_i^t \boldsymbol{\beta}_k)}{\exp(\mathbf{x}_i^t \boldsymbol{\beta}_k)} \right) + \alpha \sum_{k=2}^{K} ||\boldsymbol{\beta}_k||_1.$$

Since there is no analytic solution for this optimization problem available only a numerical approximation is available.

One algorithm used to find such numerical solutions for the optimization problem is based on a coordinate descent approach but since a further discussion of the same

would be inappropriate, a detailed description can be found in Friedman, Hastie, and Tibshirani (2010).

Furthermore, the function `cv.glmnet` from the R package `glmnet`, implemented by Friedman, Hastie, and Tibshirani (2010), allows us to use this algorithm and therefore provides results for this type of estimation. Additionally, there is already a way of estimating the value of the penalization parameter α via the cross-validation method implemented.

Before the results of the application are provided some remarks on the evaluation of the classifier are made. The first one is that the methods for evaluation are in general the same than for the wrapper models; this should make the results more comparable and easier for understanding.

The second one deals with the unbalanced design and the impact for the training of the classifier, especially during the leave-one-out cross-validation procedure. As we already mentioned the estimation of the penalization parameter α is done with a cross-validation method, more precisely a leave-one-out cross-validation procedure. This choice was made since for the Variety-datasets there are varieties available which are only represented by three samples which cause computational issues when a classical k-fold cross-validation procedure is used. With the method described above we get a special situation when performing the leave-one-out cross-validation procedure in order to estimate the out-of-sample performance.

Let us assume we apply the general leave-one-out cross-validation procedure for estimating the out-of-sample classification accuracy. Then we have the case that one of these three available samples for a specific variety is left out because of the evaluation procedure. Another one is left out due to the estimation of the penalization parameter. This means that only one sample representing the specific variety is left in the training set. Moreover, with the implementation of the `glmnet` function this causes an error because it is not reasonable to fit a model on a data set where one class is represented only by one sample.

Therefore, a leave-one-out cross-validation procedure for estimating the out-of-sample classification accuracy is not possible and therefore also not provided for the Variety-dataset in the following. Due to a lack of other methods to estimate the out-of-sample performance with this data setting and since the results using the wrapper models worked very well for the classification of the variety of the grapes we will focus more on the improvement of the classification accuracy for the geographical origin in the following sections.

8.1 Application to the Variety-Dataset

As mentioned before only the in-sample classification accuracy and the number of non-zero features can be provided for the Variety-datasets. Therefore, Table 30 provides these quantities and also shows a problem concerning the number of features which are finally selected. For this type of model the term selecting a feature means that the according parameter value is non-zero.

Dataset	# features	classification accuracy (in-sample)
Area-Origin	77	100.00%
Area-Stand	77	100.00%
Height-Origin	68	100.00%
Height-Stand	68	100.00%

Table 30: Results for the embedded model applied on the Variety-datasets.

Since 68 or 77 features are finally selected for the classifier, we can observe a scenario where the lasso penalization fails in terms of selecting an appropriate number of features. Due to the fact that this is not only a large number compared to the results for the wrapper model, but also compared to the sample size itself the results are at least questionable.

Also notice that each feature is linked to nine values in the parameter vector since every class, except for the reference class, gets its own parameter vector under the multinomial logistic model (c.p. Chapter 2). This shows that the model we are dealing with is clearly a case of overparametrization.

Therefore, with this number of selected features a detailed discussion of the coefficients is not constructive and we can only say that the approach using the embedded model does fail because of the experimental design in general and the shape of the problem ($p >> n$) for this case in particular.

8.2 Application to the Geography-Dataset

As for the Variety-datasets an overview of the results, containing the number of non-zero features, the in- and out-of-sample classification accuracy along with the run time for the leave-one-out cross-validation procedure, for the Geography-datasets is provided in Table 31. In this case it is possible to perform the leave-one-out cross-validation procedure because all regions are represented by at least four samples.

Dataset	# features	classification accuracy		run time cross-validation [min]
		in-sample	out-of-sample	
Area-Origin	91	100.00%	80.90%	41.86
Area-Stand	91	100.00%	80.90%	40.99
Height-Origin	74	98.88%	70.79%	42.23
Height-Stand	74	98.88%	70.79%	42.43

Table 31: Results for the embedded model applied on the Geography-datasets.

If the number of features in the final model is considered, we can observe that the number of selected features is even larger than for the Variety-datasets. This again shows that we are dealing with models that are very likely to be overparameterized. But keep in mind that the final number of parameters is the product of the number of selected features and the number of available classes. Therefore, we observe that a model with less parameters is required to classify the geographical regions of the grapes rather than the variety. However, this statement must really be handled with care since both classification problems and the according data differ in many ways and the observed results should not be used for clear statements since the models itself do not seem to be as reliable as the pure wrapper models.

However, the out-of-sample classification accuracy of over 70% and up to 80% is quite impressive and proves to be the best result we achieved for this problem throughout all discussed methods. Also, the run time of under one hour is quite impressive compared to the pure wrapper models. This shows the advantage of the one step procedure even when the optimization step is more complex and the additional regularization parameter α needs to be estimated. We are of course much faster than when a sequence of several optimization steps is required.

As a final point a short sensitivity analysis in form of the relative frequency of the selected features in the final models during the leave-one-out cross-validation procedure is presented in Figure 43.

There we can observe that the frequencies of features selected in the final models are identical for the original and standardized versions of the data. Combined with the results from Table 31 we can conclude that the data standardization does not affect the results of the embedded models.

For putting the relative frequency of the selected features in the final model into context we have a problem because so many features were selected. Therefore, a number of around 200 features which occur at least in one model during the leave-one-out cross-validation procedure is very low compared to the number of features in the final model when using all data between 74 and 91.

Notice that in our case many features are selected which means that every feature which is slightly significant occurs in the model. This means that the results seem to be stable but are difficult to compare to the results of the wrapper model.

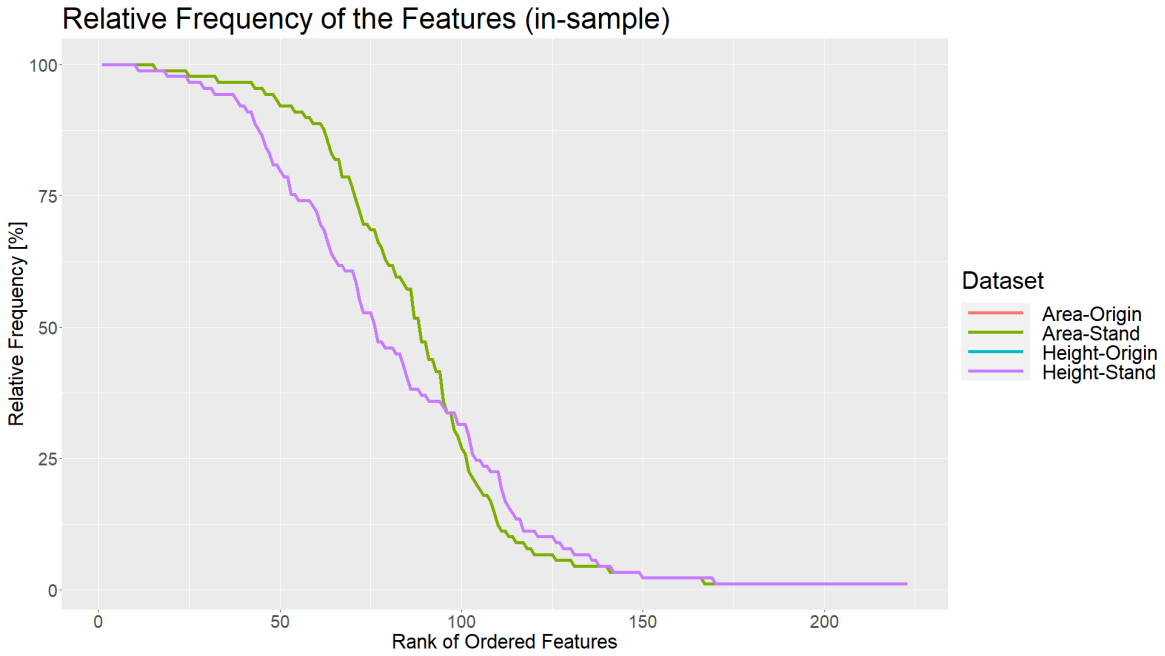

Figure 43: Relative frequency of the features in the final model (Geography-datasets).

As a last point we want to take a closer look at the wrong classifier samples during the leave-one-out cross-validation procedure. Therefore, Table 32 provides the confusion table when the embedded model is applied on the original Geography-Area-dataset with the difference that here for each sample a individual model was fitted.

Predicted \ True	**LEITHABERG**	**SLO**	**SUED**	**VL**	**WEST**
LEITHABERG	1	0	0	0	0
SLO	0	17	0	0	0
SUED	0	0	24	3	2
VL	3	0	5	24	2
WEST	0	0	0	2	6

Table 32: Confusion table for the original Geography-Area-dataset.

As Table 32 shows most samples from LEITHABERG are classified wrong which could be caused by the small sample size for this region. Therefore, the classifier underestimated the according probability. As mentioned in previous chapters the regions SUED

and VL seem to have grapes of similar chemical structures and are difficult to distinguish under our circumstances, this can also be observed when the embedded model is applied. As already mentioned in Chapter 6, this coincides with the geographical distance of these regions.

Another point worth to be mentioned in this context is that the region VL has different geological properties which are not discussed in further detail, but these results could also be a hint that a stratification according to the soil and not the political regions would lead to a better determination of the geographical origin.

But all these conclusions and statements are based on the results of a class of models where the results are questionable. Therefore, further investigation and research would be required to verify or disprove these statements.

8.3 Conclusion of Embedded Models

The approach using the embedded models is clearly the one with the shortest run time, when a model with the ability to predict the classes is used. Furthermore, we also observed that this approach selects the most features, which is the problem with this type of models in our application. This is because here clearly more features are selected than reasonable, especially when the number of fitted parameters is compared to the sample size.

The number of parameters indeed exceeds the number of available samples which is always a problem in the context of statistical modeling because this indicates that the results, which are based on numerical computation, do not approximate a valuable or practicable solution.

Another problem we observe in this chapter is the drawback due to the unbalanced design. This restricts the methods for evaluation and testing of the according classifiers and the general methodology in a way that we have not observed in other chapters before. Maybe a setting where more observations are available could lead to stable and more meaningful results, but with this setting the high number of out-of-sample classification accuracy achieved by the embedded models is very questionable.

Furthermore, for the search of very significant features which should in a next step allow to identify special chemical compounds, it seems that under the provided circumstances this method is not really constructive and practicable. This means that either further research for the embedded models or other approaches, like the wrapper models, are required.

Appendix

Abbreviation	German Name	Further Information
SLO	Slowenien	https://en.wikipedia.org/wiki/Slovenia
AUT	Österreich	https://en.wikipedia.org/wiki/Austria
STMK	Steiermark	https://en.wikipedia.org/wiki/Styria
BGLD	Burgenland	https://en.wikipedia.org/wiki/Burgenland
NÖ	Niederösterreich	https://en.wikipedia.org/wiki/Lower_Austria
SÜD	Südsteiermark	https://en.wikipedia.org/wiki/Districtus_Austriae_Controllatus#DAC_regions
WEST	Weststeiermark	https://en.wikipedia.org/wiki/Districtus_Austriae_Controllatus#DAC_regions
VL	Vulkanland	https://en.wikipedia.org/wiki/Districtus_Austriae_Controllatus#DAC_regions
EISEN	Eisenberg	https://en.wikipedia.org/wiki/Districtus_Austriae_Controllatus#DAC_regions
LEITHA	Leithaberg	https://en.wikipedia.org/wiki/Districtus_Austriae_Controllatus#DAC_regions
NSEE	Neusiedlersee	https://en.wikipedia.org/wiki/Districtus_Austriae_Controllatus#DAC_regions
WV	Waldviertel	https://en.wikipedia.org/wiki/Districtus_Austriae_Controllatus#DAC_regions

Table 33: All geographical regions with their abbreviations.

Abbreviation	**German Name**	**Further Information**
BF	Blaufränkisch	https://en.wikipedia.org/wiki/Blaufr%C3%A4nkisch
BW	Blauer Wildbacher	https://en.wikipedia.org/wiki/Wildbacher
CH/MO	Chardonnay/Morillon	https://en.wikipedia.org/wiki/Chardonnay
CON	Concorde	-
CS	Carbernet Sauvignon	https://en.wikipedia.org/wiki/Cabernet_Sauvignon
FU	Furmint	https://en.wikipedia.org/wiki/Furmint
GB	Grauburgunder	https://en.wikipedia.org/wiki/Pinot_gris
GM	Gelber Muskateller	https://en.wikipedia.org/wiki/%0D%0AMuscat_Blanc_%C3%A0_Petits_Grains
Gold-M	Gold Muskateller	https://en.wikipedia.org/wiki/Moscato_Giallo
GV	Grüner Veltliner	https://en.wikipedia.org/wiki/Gr%C3%BCner_Veltliner
MER	Merlot	https://en.wikipedia.org/wiki/Merlot
PN	Blauburgunder	https://en.wikipedia.org/wiki/Pinot_noir
RADRAM	Radgonska Ramina	-
RIES	Riesling	https://en.wikipedia.org/wiki/Riesling
R-RIES	Rheinriesling	https://en.wikipedia.org/wiki/Riesling
SAM	Sämling	https://en.wikipedia.org/wiki/Scheurebe
SB	Sauvignon Blanc	https://en.wikipedia.org/wiki/Sauvignon_blanc
SIL	Grüner Sylvaner	https://en.wikipedia.org/wiki/Silvaner
ST-L	Sankt Laurent	https://en.wikipedia.org/wiki/St._Laurent_(grape)
SYR	Syrah	https://en.wikipedia.org/wiki/Syrah
TR	Traminer	https://en.wikipedia.org/wiki/Savagnin
WB	Weißburgunder	https://en.wikipedia.org/wiki/Pinot_blanc
WR	Welschriesling	https://en.wikipedia.org/wiki/Welschriesling
ZW	Zweigelt	https://en.wikipedia.org/wiki/Zweigelt

Table 34: All grape varieties available in this project with their abbreviations.

System Part	Specification
Hardware	
Processor	Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80GHz
Installed RAM	8.00 GB (7.88 GB usable)
System type	64-bit operating system, x64-based processor
Graphics card (on-board)	Intel(R) UHD Graphics 620
Graphics card (dedicated)	NVIDEA GeForce MX150
Software	
Operating system	Windows 10 Home (Version 1909)
Statistic software	R version 3.6.3 (2020-02-29)
Integrated development environment (IDE)	RStudio Version 1.1.463

Table 35: Technical specification of the laptop used to perform the analysis.

```
correct_features <- function(df)
{
 # Rename the columns for input df to standardize the names
 colnames(df) = c('Samples', 'Attribute', 'Time', 'Feature')

 # Only use the qc-measurements (no attributes available)
 df_qc = df[!complete.cases(df), 3:4]

 # Search for the appropriate model
 threshold_p_value = c(0.05, 0.05, 0.01)
 model_degree = 0

 for(i in 3:1)
 {
  model = lm(
    formula = Feature ~ poly(Time, degree = i, raw = TRUE),
    data = df_qc
    )

  if(summary(model)$coefficients[i+1, 4] <= threshold_p_value[i])
  {model_degree = i
   break
   }
  }

  # Adjust the output according to the model degree
  if(model_degree == 3)
  {return(rep(NA, times = length(df$Feature)))}
  if(model_degree %in% 1:2)
  {
   predict_qc = predict.lm(
    object = model, newdata = df, type = 'response'
    )

   center = floor(mean(df$Time)):ceiling(mean(df$Time))
   center_point = round(mean(predict_qc[center]))
   correction_factor = center_point / predict_qc
   return(correction_factor * df$Feature)
  }
  if(model_degree == 0)
  {return(df$Feature)}
}
```

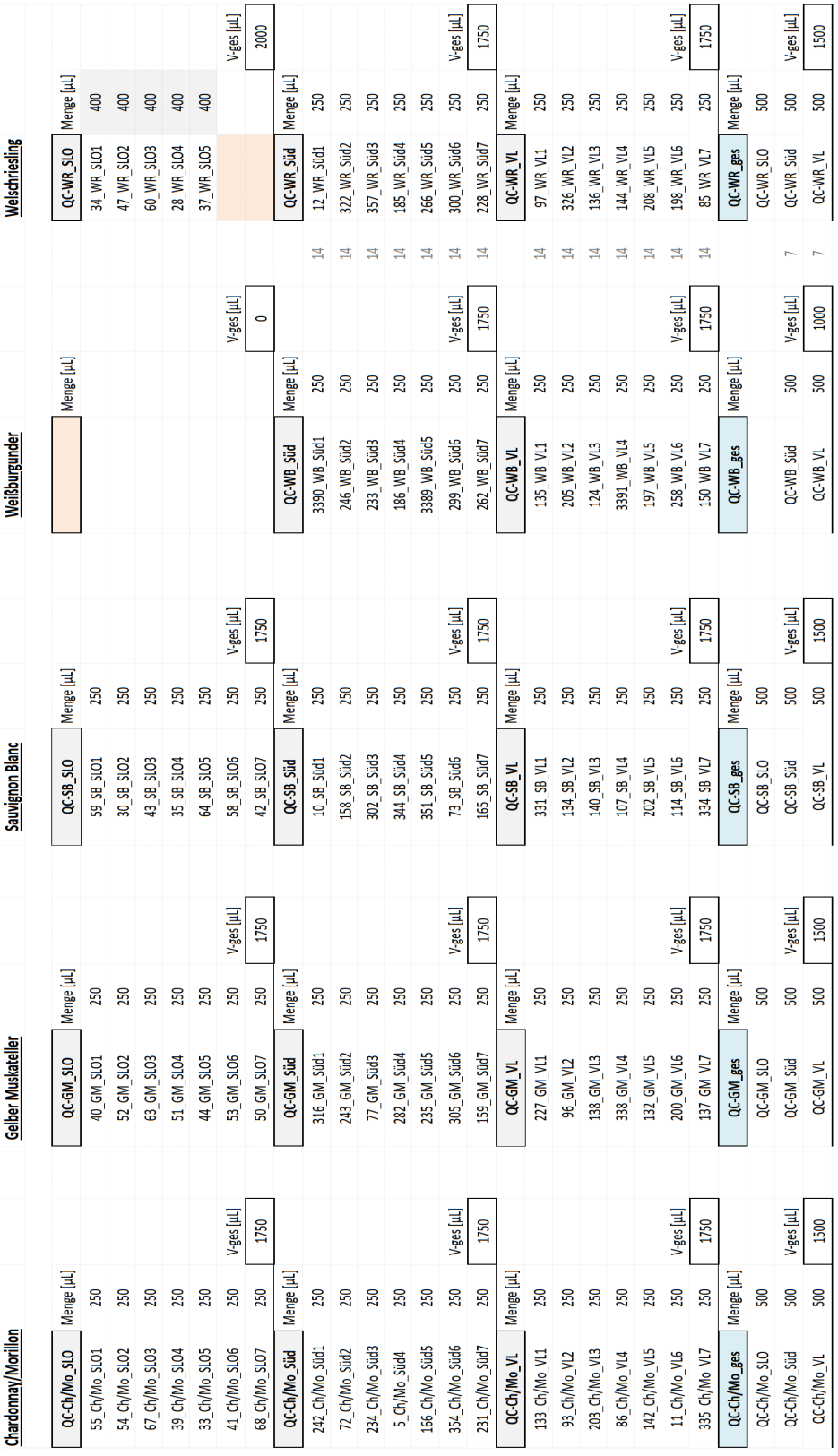

Figure 44: Mixture for the compounds used to generate the substance for the QC measurements.

Bibliography

Casella, G., & Berger, R. (2002). *Statistical inference* (2nd ed.). Pacific Grove, California: Duxbury Press.

Draper, N., & Smith, H. (1998). *Applied regression analysis*. New York: John Wiley and Sons.

Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, *33*, 1–22.

Fuchs, F., & Friedl, H. (2020). *Authenticity of styrian wine grapes*. Institut of Statistics at the Graz University of Technology. Unpublished report.

Guyon, I., & Elisseeff., A. (2003). An introduction to variable and feature selection. *The Journal of Machine Learning Research*, *3*, 1157–1182.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference and prediction* (2nd ed). New York: Springer Science+Buisness Media.

Herbert, C. (2003). *Mass spectrometry basics*. Boco Raton, Florida: CRC Press.

Ma, S., & Huang, J. (2008). Penalized feature selection and classification in bioinformatics. *Briefings in Bioinformatics*, *9*, 392–403.

McCullagh, P., & Nelder, J. (1989). *Generalized linear models* (2nd ed.). London: Chapman and Hall.

Meyer, V. (2013). *Practical high-performance liquid chromatography*. New York: John Wiley and Sons.

R Development Core Team. (2008). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria.

Rosasco, L., De, E., Caponnetto, V. A., Piana, M., & Verri, A. (2003). Are loss functions all the same. *Neural Computation*, *16*.

Shao, J. (1998). *Mathematical statistics*. New York: Springer.

Tang, J., Alelyani, S., & Liu, H. (2014). Feature selection for classification: A review. *Data Classification: Algorithms and Applications* (pp. 37–64). Boco Raton, Florida: CRC Press.

Venables, W., & Ripley, B. (2002). *Modern Applied Statistics with S* (4th ed.). New York: Springer.

Wollschläger, D. (2014). *Grundlagen der Datenanalyse mit R* (3rd ed.). Berlin: Springer.

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitely indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master‘s thesis.

Date	Signature