Spring 2018
Master's Thesis

AARHUS UNIVERSITET

Martin Højland Hansen
Supervisor: Associate Professor Anders Bredahl Kock

Deal or no deal:
How much is your house worth according to machine learning prediction algorithms

Danish title: Deal or no deal: Hvor meget er dit hus værd ifølge machine learning prædiktions algoritmer
Degree programme: International Master of Science in Quantitative Economics (cand. oecon - IMSQE) | Study ID: 201305187
Submission deadline: 01/06-2018
Subject area: Econometrics
This thesis may be published

DEPARTMENT OF ECONOMICS AND BUSINESS ECONOMICS
AARHUS UNIVERSITY • DENMARK

DEAL OR NO DEAL

How much is your house worth according to machine learning algorithms

Author:
Martin Højland Hansen
201305187

Abstract

This thesis investigates the scope for using machine learning algorithms within the context of creating an automated valuation model for single-family houses in Denmark. It outlines why machine learning yield better predictions, and how it breaks the dogma of unbiasedness in econometrics by concentrating solely on prediction. This means that it needs to allow for more complex modelling of inputs, while regularising the model to reduce the variance of the predictions and also possibly combatting model uncertainty by creating an ensemble of models. Several different types of models are compared based on their predictive abilities, and it is found that models which inherently model complex relationships while still successfully reducing prediction variance have superior performance. This is done on self-gathered publicly available data, which have been extracted from several sources. The backbone of the data is from OIS, consisting of Danish register databases, and is gathered in cooperation with the land surveying company LIFA A/S. In total, the thesis includes sales of 179952 single-family houses and has 151 input variables for the generalised linear models. In order to alleviate spatial autocorrelation, which is found to be substantial within mass appraisal, locational data is gathered and exploited. This data is used to get distances to locational amenities and to create neighbourhood square metre prices as an input, which mimics how buyers assess relative prices, but also includes some, otherwise unobserved, heterogeneity idiosyncratic to certain areas - both locational amenities but also house characteristics. The model with the best performance is the extreme gradient boosted regression tree, which incorporates several features from other models in a seamless and efficient algorithm. Finally, this thesis concludes that since the machine learning algorithms substantially outperform both the linear model and SKAT's suspended valuations, they should be considered more heavily in Danish mass appraisal models.

Indeholder data fra Styrelsen for Dataforsyning og Effektivisering
The data set GeoDanmark is downloaded the 11th of April 2018

Contents

List of Figures

List of Tables

1 Introduction

1.1 Housing valuations: a difficult task

Long has the Danish real-estate valuations been heavily criticised for being unfair and poorly estimated. In 2003, SKAT, who has jurisdiction over collecting taxes in Denmark, was assigned to the job. However, according to a report in 2013 by Rigsrevisionen, the Danish government's independent audit authority, SKAT has not lived up to its responsibility of providing good and fair valuations [Rigsrevisionen, 2013]. According to an article in the news section of the Danish ministry of taxation's webpage, the report from Rigsrevisionen partly led to the suspension of further valuations from SKAT[1]. The former government then appointed a committee, the so-called Ekspertudvalg om Ejendomsvurdering (Engbergudvalget) to investigate the scope for a new model for calculation. In 2014 they submitted their final report, which showed that it is possible to establish a more transparent valuation system based on better data and with starting point in statistical methods and laid out the groundwork for a given statistical model. The work was followed up by Skatteministeriets (Ministry of Taxation's) Implementeringscenter for Ejendomsvurdering (ICE), who plans to implement the new system in 2019 [ICE, 2016]. This is a new era of Danish valuations, since SKAT has not previously set out specifically to determine their valuation of a single-family house based on the selling prices of similar houses. In fact, Rigsrevisionen [2013] writes that SKAT, at that point, actually found this approach wrong. But Rigsrevisionen [2013] finds it very important to have a viable frame of reference, especially since citizens have a right to complain if they are dissatisfied with their valuation. Furthermore, a reasonably large portion of Danish taxes are collected as housing taxes; 38 billion Danish kroner yearly according to Rigsrevisionen [2013]. Thus, it is very relevant to be able to provide a just, more transparent and precise basis for collecting these taxes.

A statistical model that finds the relationship between house prices and characteristics influencing the price is called an Automated Valuation Model (AVM) or Computer Assisted Mass Appraisal (CAMA), and the literature investigating these relationships is fairly extensive, as seen in the literature review section, Section 2. The benefits of a good AVM are not only useful for taxation purposes but also useful for banks, rating agencies and aspiring homeowners or sellers. These other stakeholders favour precision in estimates comparatively more than the government, and so the scope for this thesis is to investigate statistical modelling with out-of-sample (and therefore prediction) focus. This thesis is not the first to explore these techniques, which goes

[1] *www.skm.dk/aktuelt/nyheder/2016/oktober/danskerne-faar-nye-og-mere-retvisende-ejendomsvurderinger-i-2019*

under the name of machine learning, on housing data. In fact, there are several websites that provide this service, such as *www.zillow.com* for the US market, although none does this in Denmark. Moreover, the scope for using these techniques in economics in general is vast as found in Athey [2017]. Athey [2017] finds the transformation of economics given these techniques important, all the while making it more relevant for economists to worry about what they have always worried about in regards of the data's capabilities of identifying causal effects. Therefore, this thesis will motivate the use of recent developments in prediction modelling using machine learning in general with the Danish housing market as a very relevant case study. In addition, it will explore the scope for using machine learning methods for mass appraisals with a special focus on the Danish government's use of these predictions for taxation purposes.

In the next subsection, the differences between traditional statistical models and machine learning models are introduced, and how these differences make machine learning models better for prediction purposes.

1.2 Making better predictions with machine learning

Machine learning techniques might seem similar to standard statistical models, such as the linear regression model, arguably the most frequently used statistical model type used, but they are fundamentally different. As standard econometric models are concerned with uncovering structural or causal relationships, unbiasedness is paramount. Simply, they aim to minimise the difference between the estimator's expected value and the "true" value of said parameter within a given function class. Biases can stem from many different sources, one of them being omitted variable bias, which occurs when we leave out one or more relevant variables. A strategy can then be to include all possible variables in the model to eliminate unbiasedness given the function class is correctly specified, but that requires a large degree of freedom which increases the prediction variance. Moreover, as convenient it may seem to specify the model as linear, due to the easily interpretable results it yields, as unlikely is it that the true data generating process is linear in general. These problems are the focal points of machine learning, and the trade-off can be mathematically expressed; that expression is called the bias-variance decomposition (explained in Subsection 3.2). It shows us that there is a trade-off between bias and variance when predicting an uncertain event. Thus, machine learning is breaking with the dogma of unbiasedness and focusing primarily on prediction[2]; so the techniques are trying to minimise both bias and variance simultaneously, leading to possibly far better predictions. The scope for using these

[2]As a side note, Explainable Artificial Intelligence (xAI) is a sub-element of machine learning trying to alleviate the interpretability problem inherent to the techniques, or using machine learning predictions in auxiliary predictions problems as well as main predictions problems.

methods specifically for real-estate valuation is large; the amount of data available is huge, the complexity of the relationships between the input variables and the outcome variable is immense, and the focus should mainly be on prediction rather than estimation of parameter values to extract causal relationships. To add to the point of the applicability of machine learning techniques to housing valuations, Mullainathan and Spiess [2017] uses that exact example throughout their article. They look directly at the differences between traditional econometric approaches and machine learning. They describe the main difference between supervised machine learning and traditional econometric approaches as the former revolving around prediction, and how it manages to uncover *generalisable* patterns, and the latter revolving around parameter estimation; finding good estimates of parameters, β, that underlie the relationship between the dependent and independent variables. Cherkassky and Mulier [2007] also make a point out of distinguishing between traditional statistical model estimation and machine learning. They explain that classical statistics assume that the data is generated from some distribution with *known* parametric form, and the goal is to estimate certain properties of that distribution. So classical statistics rely heavily on parametric assumptions and asymptotic arguments. For example, applying the maximum-likelihood approach to linear regression with normal independent and identically distributed (niid) noise leads to parameter estimation via least squares. Once again, they emphasise that machine learning techniques are explicitly created to fit models to have good generalisation capabilities within finite samples.

1.3 Data

The data is derived from several Danish register databases as well as relevant open data in order to include as much relevant information as possible affecting house prices. The main part of the data is gathered in cooperation with the land surveying company LIFA A/S through "Den Offentlige Informationsserver" (OIS), which includes the register databases "Bygnings- og Boligregistret" (BBR), "Statens Salgs- og Vurderingsregister" (SVUR) and "Det Fælleskommunale Ejendomsstamregister" (ESR). They jointly contain information on all sales and valuations of Danish real-estate property including building-specific information such as construction materials, location and size. In addition to this, public data from GeoDanmark and Statistics Denmark is gathered, which gives information on locational amenities, such as distance to the coast and the nearest train station, and municipality characteristics, such as the local unemployment level. In the house pricing literature, there are reported problems about spatial autocorrelation, an effect stemming from spatially close houses being similar in appearance and structure while also sharing the same locational amenities, such as shopping opportunities etc. This problem is alleviated by

creating input variables that measure local square metre prices in a given year. In total, the thesis includes sales of 179952 single-family houses and has 151 input variables for the generalised linear models. This data is gathered for the years 2005-2016 and can easily be extended. Noteworthy is it that 2016 is used as a test year since SKAT's appraisals, to which the models are held up against, are updated to be relevant for this year - but not other years.

1.4 Structure of the thesis

In Section 2, the preceding literature on housing valuations and the prediction of house prices is reviewed. The "why" and the "how" of automated valuation models is looked into, as well as the progress in Danish context. The Danish market is briefly investigated using literature primarily from the National Bank of Denmark. Moreover, the recent developments around property appraisals and their uses, in e.g. ad valorem taxation, are presented.

In Section 3, an introduction to machine learning is given in a model-free way, where the primary focus will be on the bias-variance trade-off and how to (and to what extent one can) estimate generalisation error.

In Section 4, how we can use the generalisation error to tune model parameters, and hence the optimisation of the model for prediction in a data-driven way, is investigated. Parallels to Bayesian model estimation is drawn, and how several of the regularisation techniques can be viewed in a Bayesian way are outlined throughout the section. Later, Bayesian methods of model estimation and model averaging methods are also considered. In this section, several ensemble methods and other methods of optimising the prediction model using more models are introduced.

In Section 5, individual supervised learning techniques for regression are introduced. Namely, tree-based methods such as the random forest and gradient boosted trees, K-nearest neighbour regression and several types of neural network architectures are explained.

In Section 6, the data set is described together with the processing it has undergone; descriptive statistics on the included variables is given, and a description of how the data set is cleaned. Furthermore, summary statistics are presented to give an understanding of what the data looks like.

In Section 7, the results of the developed models are presented and compared to SKAT's valu-

ations. They are compared with SKAT's valuations in the same manners as in ICE [2016]. How some of the methods are regularised, and what effect that has on their generalisation abilities, is evaluated. Finally, the models are also compared to each other.

In Section 8, concluding remarks from the analysis is given. It makes remarks on the predictive ability of machine learning algorithms in contrast to generalised linear models, and how either the public or the government could use these. Furthermore, suggestions on how to improve upon the model, and what else should be evaluated for successful implementation, are considered.

2 Literature Review and Danish Property Appraisals

2.1 An Automated Valuation Model (AVM)

Market value predictions for residential properties are important for investment decisions and risk management of households, banks and real estate developers [Schulz et al., 2014]. Furthermore, the government uses them for ad valorem tax purposes, where it is commonly split between lot value and property value. Schulz et al. [2014] provide a comprehensive insight into the data needs and the stages involved in developing an Automated Valuation Model (AVM)[3]. They also examine the statistical model development and how to validate the predictive model. They find that one of the most important features of mass appraisal is the availability of good data, although some real-estate properties are too distinct to be properly appraised in automatic manners when the objective is providing a good basis for taxation. Hence, it is okay for the AVM to focus on real-estate properties where structural and location characteristics are easily observed and homogenised with the spatially close properties, since properties that fall out of this category can be valued by a professional valuer (or appraiser) in a full physical inspection. The physical inspection may be more precise than the AVM but is also time consuming and expensive. In many instances, market participants are prepared to trade off predictive accuracy for cost. First, banks can use low-cost appraisals when underwriting loan advances, home equity withdrawals and remortgaging, or they could use it to manage risk as a tool to monitor collateral values underlying the bank's portfolio of mortgage loans. Second, rating agencies may request information about current loan-to-value ratios to estimate loss severity and probability. Third, aspiring property owners (or sellers) will want to get an idea of the cost of a home (get a feeling of how much they can get for the home). Sellers would then be able to use the information to inform the decision to relocate. And, of course, fourth, government agencies can use AVM's as cost-effective appraisals for taxation, planning and land-use regulation. The literature on AVM's is large enough to have a defined standard for mass appraisal of real property, and IAAO [2013b] defines this standard. The standard is focused on appraisal for ad valorem tax purposes, but they deem it relevant for other sakes as well. The reason for a specific standard related to mass appraisal for ad valorem tax purposes is the additional requirements of the government related to the defence of the given appraisal of a property. In any given case, the appraiser should be able to defend and explain the given appraisal to the property owner. McCluskey et al. [2013, 2014] indeed name that a critical factor for the AVM is the explainability of the model in terms of being able to defend the estimates in a formal setting, such as an appeal tribunal or court. It is given in the committee report of the

[3]The term is also interchangeably used with Computer Assisted Mass Appraisal (CAMA).

Danish government investigating the scope for a new mass appraisal model (Engbjergudvalget's rapport, Jensen et al. [2014]) that an overriding principle is that of greatest possible transparency. The committee sees this as a prerequisite for the general acceptance of the model as grounds for taxation as well as comprehension of the results and in extension hereof optimal opportunity to complain. Kauko and d'Amato [2008] set up further criteria for comparison between modelling approaches: predictive accuracy, conceptual integrity, internal consistency of the model, reliability and robustness of the model, and feasibility in terms of cost and time efficiency. Even though single predictions are difficult to explain within the (relative) "black-box" nature of some machine learning algorithms [Friedman et al., 2009], McCluskey et al. [2013] still evaluate models based on predictive accuracy on grounds that it is considered the fundamental component of an AVM; as will be done in this thesis. Previous studies of machine learning algorithms for mass appraisal have focused on the performance of Artificial Neural Networks, see for example Tay and Ho [1992], Kathmann [1993], Limsombunchai [2004], Peterson and Flanagan [2009], but there are also attempts using other techniques, such as using boosted regression trees in McCluskey et al. [2014] and random forest regression in Antipov and Pokryshevskaya [2012]. With almost no exceptions (none is found), they find that machine learning algorithms are superior in predictive accuracy in the housing markets, which is aligned with the general literature on machine learning.

These techniques stand in contrast with the traditional approach of a hedonic price based on the work of Sherwin Rosen in Rosen [1974], which is then primarily estimated using OLS as in Goodman and Thibodeau [1995], Ottensmann et al. [2008], Payton et al. [2008]. Though some of these applications of OLS are primarily for the use of assessing the causality of explanatory factors within the house pricing equation, as it is for Ottensmann et al. [2008] in estimating the effect of spatial proximity of houses and employment centres and for Payton et al. [2008] in estimating the effect of green areas in urban environments on housing prices, these methods are considered the method of choice for an AVM [Gloudemans and Miller, 1976, Mark and Goldberg, 1988, McCluskey et al., 2014]. So much so that many countries, including Australia; New Zealand; Canada; USA; Netherlands and Denmark, use versions of it for property tax assessment [McCluskey et al., 2014]. However, as aforementioned, this might be due to the explainability factor especially relevant for government issues. There is an extensive literature of different problems related to the estimation of prediction equations using linear OLS models in the area of property pricing. The most prominent problem is that of spatial autocorrelation. This problem is well explained in Basu and Thibodeau [1998], who argue that spatial dependence exists because nearby properties will often have similar structural features (they are often developed at the same time) and also share locational amenities, such as having equally good grocery shopping possibilities, equally good access to job opportunities or the same waterfront view. This

problem has led to some model specifications actively seeking to alleviate it while still staying in the realm of an explainable model, such as the hierarchical model of Goodman and Thibodeau [1998], the simultaneous autoregressive model and the geographically weighted regression both evaluated in McCluskey et al. [2013]. Goodman and Thibodeau [1998] also give an excellent review of what factors will influence the creation of sub-markets of similar houses, and how it becomes to be so that spatially close houses have some of the same features. For example, they find that if the hedonic price of fireplaces is to exceed the cost of building one, then suppliers will build houses with fireplaces, and landlords with units that do not have fireplaces will install them.

Implementeringscenter for Ejendomsvurdering (ICE), who took up the job of implementing the new mass appraisal model after Engbjergudvalget, writes in ICE [2016] that they have created a neighbour model, which by that time was expected to constitute Denmark's new mass appraisal model for taxation of property. Their neighbour model is a local weighted average, where spatially close houses are given more weight, determining the average log square metre price in the area and a hedonic price model similar to that in Ottensmann et al. [2008]. These models are combined so that the weighted log square metre price is the intercept in the model, and dissimilarities between the given house and the spatially close ones are adjusted using the estimates from the hedonic model.

In a personal email, the head of ICE has told me that they now want to apply a K-nearest neighbour for determining area square metre prices for similar properties for use in their regression. Furthermore, they want to add a generalised additive model where a two-dimensional spline constitutes the geography, and the rest of the input variables is modelled using a hedonic regression model adjusted for neighbour prices and neighbour house specifications. Furthermore, they apply an eXtreme Gradient boosting (XGboost) model to compare results with their model and make further investigations into the houses where the models disagree. In ICE [2016], they compare their results to SKAT's older model, which is still the one used in practice, and find that they have more precise results based on the objective of estimating realised prices. However, SKAT's results are a bit different, as they are not designed explicitly to predict house prices, but they have also had manual corrections on specific houses that are difficult to predict [ICE, 2016]. Even though ICE will not directly implement machine learning algorithms directly in their predictive equation, the scope for doing so is immense since the taxes amount to a significant amount of the government proceeds, and there are a lot of other applications where predictive accuracy is paramount. Machine learning techniques are specifically designed for good predictive ability, as they in a smart way try to minimise generalisation error [Mullainathan and Spiess, 2017]. Friedman et al. [2009] is a good place to start learning about machine learning, as it

is a fairly straightforward and comprehensive book on the motivation for, applications of and methods of machine learning. In this thesis, models from two genres with different methods of creating ensembles from those genres will be applied. These models then include a version of boosted regression trees, based on the developments of the boosting technique in Schapire and Freund [2012] which has shown great predictive abilities in general and good promise within house predictions in McCluskey et al. [2014]. They also include a random forest model, which works by constructing several regression trees where only a randomly chosen subset of the input variables can be selected for splitting in each tree and finally averages them. This procedure decreases the similarities between models and hence decreases the variance of the final predictions. Finally, these models will be compared with the neural network model which has been the primary choice of statistical academics trying to predict house prices due to its ability to estimate highly complex and non-linear relationships [Peterson and Flanagan, 2009]. The literature surrounding machine learning will be investigated further in the coming sections.

As we will be making appraisals for Danish single-family houses it is valuable to look into that market on a general level; how has the market been moving? which factors affect the market? what are the dynamics of the markets? how is the legislation designed? and which changes are expected? The next subsection will briefly sum up the literature on these issues.

2.2 A primer on the Danish real-estate market the past decade

In 2011, after the supposed housing bubble burst, the Danish central bank published an extensive article analysing the Danish housing market up until that point. In an abbreviated form, the article Dam et al. [2011] will be used to understand the dynamics of the market.

First of all, like any other market, it is driven by the forces of supply and demand. The demand on a macroeconomic level depends on the expectations of the stability of the market, the borrowing costs, the households' disposable income and much more. The supply on a macroeconomic level is seen to be responsive to the housing prices, and so when the house prices are appreciating a lot of new construction is beginning. However, of course, new buildings are not built instantaneously and so increasing demand for housing, driven by, for example, the increase in disposable income, is not entirely swamped up by new development. Furthermore, there are several reasons to why a stable housing market is to be preferred. For example, the government will want to have ad valorem taxes on housing not to prefer this type of investment against other investment vehicles, and so the housing market is also dependent on the taxation scheme and the general legislation [Klein et al., 2016].

In 2004-06 the Danish economy was booming with relatively low borrowing costs and an increasing disposable income for the majority of households, which was reflected in the property market. At its highest in 2007, the average price for single-family houses was 54 percent higher in real terms than 2003 [Dam et al., 2011]. This led to a highly increased building activity as predicted by Tobin's Q of the housing market[4], where several contractors were drawn to the market in search of profits. Furthermore, the availability of plots of land for development varies across the country, and it is namely in Copenhagen that the highest price increases have occurred lately. Due to several legislative as well as natural courses, the development here is still sluggish and therefore inelastic compared to the rising demand [Dam et al., 2011]. Several authors including Dam et al. [2011], Isaksen [2015], Klein et al. [2016], Hviid and Kramp [2017] believe that the recent price increases in the Copenhagen area is partly due to the 2002 tax freeze, where the taxable value of a house was frozen, and so increases in the value of a house does not lead to higher taxes - the effective tax rate was declining. This had re-distributional effects as the house prices in Copenhagen rose more than in the rest of the country, and so the majority of the tax break was distributed to Copenhagen homeowners[5]. Furthermore, the natural stabilising effects housing taxes was reversed, as appreciating property prices yield lower effective tax rates instead of higher taxes. Some of these authors, such as Isaksen [2015], are warning against a local house price bubble in the Copenhagen area, which has not burst as of this moment.

In recent years the housing market has rebounded from the recession, as seen on the figure on the next page, and in the Copenhagen area the single-family houses investigated in this thesis is almost at the height of the peak in 2007. In the second graph, we can also see that the Copenhagen housing market has grown a lot more since 1993 than the whole of Denmark. It also seems to be more volatile, which is supported by the paper Hviid [2017]. This paper investigates regional differences and how the sub-markets are interconnected and finds that the Copenhagen area creates a ripple effect in prices to other parts of Denmark that is much stronger than the ripple effect from the opposite direction.

[4]Which is defined as the ratio between the established housing prices and their replacement value.

[5]In addition to this, then, as of this moment, the marginal tax on houses worth more than 3.040.000 kr is to be taxed three times higher than otherwise. However, as of the tax freeze, houses increasing beyond that level received an even greater amount of the tax break, and these houses were mainly in the Copenhagen area [Dam et al., 2011].

Figure 2.1: Danish House Price Indices

Note: The displayed house price indices is divided by the Danish consumer price index to present them in real terms. The indices are using 2015 as base year. The first graph displays the House Price Index of Denmark and is used in the algorithms to give an indicator of the level of prices in the housing market. The second graph displays the price indices for single-family houses sold in Copenhagen and the whole of Denmark respectively .
Source: Statistics Denmark.

According to Hviid and Kramp [2017], the Danish government is reinstating variable taxes on housing (varying with respect to the valuation and not frozen at some level), which both

Hviid and Kramp [2017] and Klein et al. [2016] believe is sound policy, since it will dampen the volatility of the market. Furthermore, Klein et al. [2016] suggest that it will have an impact on the price level of housing in the Copenhagen area due to higher taxation costs, as the benefits of the artificially lowered effective housing tax are annulled.

It is important to notice that none of the results of this paper have any dynamical features, and so they do not have any abilities to predict an economic downturn or housing market collapse but are merely a reflection of the housing market as of today, and how the market will price a given house in the population. Furthermore, it does not take into account the increased supply relating to the increasing house prices following a demand increase.

3 Prediction in the Era of Big Data

The economic science has evolved rapidly over the recent decades along a number of dimensions, from previously being driven mainly by theoretical work to subsequently being driven highly by empirical work. The data revolution of the past decade is likely to continue as the quality and quantity of data on economic activity are expanding rapidly [Einav and Levin, 2014]. Examining the changes in patterns of coauthorship, age structure and methodology in the three top general economics journals for one year in each decade from 1960s-2010s, Hamermesh [2013] concludes that top journals are publishing a decreasing number of papers that represent pure theory and an increasing number of empirically based studies of phenomena. The economic literature using large data sets is still primarily relying on traditional econometric techniques, where the researchers put considerable thought into controlling for heterogeneity, limiting bias or obtaining carefully constructed standard errors for the main parameters of interest. In addition, although using several different specifications of one's model to assess robustness, a single preferred specification, more likely than not linear, is the focus. This approach, both in conception and execution, stands in contrast to some of the recent developments in statistics and computer science [Einav and Levin, 2014]. This view is backed up by Cherkassky and Mulier [2007], who believes that these classical methods are unsuited for many applications because the parametric modelling in finite samples imposes too rigid assumptions about the unknown dependency between dependent and independent variables. This imposed rigid parametric form tends to introduce a large modelling bias (discrepancy between the assumed parametric model and the unknown truth). These approaches, focusing more on model uncertainty than on statistical uncertainty, is the main emphasis of this section.

3.1 An admission of uncertainty

One of the certainties we have in this world is the randomness of a fair coin. Coin tossing is one of the most basic examples of a random phenomenon. A single toss can be modelled as a Bernoulli random variable since the outcome of the toss is success/failure, and we would expect the probability of both heads and tails to be 50%. However, coin tossing is not random - it obeys the laws of mechanics, and a coin's flight is determined by its initial conditions [Diaconis et al., 2007]. So the parameter of the model, p, is not fixed. It can be modelled itself to give a better guess on the probability of heads coming up than by mere chance. The following distinction between notions of probabilities is set up for the reader to notice the subtleties of uncertain data, uncertain parameters, and uncertain models. The solution to the above-mentioned coin tossing problem could be, as always, to collect more data, and thus, by tossing the coin many times and

noting the result of each coin toss under different circumstances, and using this data to estimate the conditional probabilities of the coin ending up heads. In a frequentist sense, the probability is the limit of a long-run relative frequency:

$$\mathbb{P}(A|B) = \lim_{n\to\infty} \frac{m}{n}$$

where m is the number of heads under a given precondition, B, and n is the number of trials under that precondition. In contrast, Bayesian probability statements are about states of mind over the state of the world, and not about the states of the world per se. This subjectivist view of the world allows us to attach probabilities to propositions reflecting our degree of belief in the proposition. Bayes Theorem then tells us how to rationally revise our belief about said proposition in the light of data [Jackman, 2009]. Following Jackman [2009], we say θ is some object of interest subject to uncertainty - a parameter, a hypothesis, a model, a data point - then Bayes Theorem tells us how to rationally revise our prior beliefs about θ, $\mathbb{P}(\theta)$, in the light of data y, to yield *posterior beliefs* $\mathbb{P}(\theta|y)$.

Bayes' Theorem:

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)} = \frac{\mathbb{P}(B|A)\mathbb{P}(A)}{\mathbb{P}(B)}. \tag{3.1}$$

Bayes' Theorem tells us, then, how to change *any* belief about a given proposition and provide us with means to summarise our idea of said belief in a probability distribution. We are therefore allowed not to think about parameters as fixed objects to be truthfully estimated but to be objects used to reflect our degree of belief about a proposition, themselves being objects of uncertainty. In the coin tossing example, we are allowed to have a prior on the probability of success to be 50% with a low uncertainty on that proposition and then collecting data that might disprove the prior. If enough evidence suggesting that a low toss changes the probability of success is gathered, we can use this information to revise the posterior probability.

Furthermore, if one then wants to use said parameters of a model to predict an event, it is not only the mode of the parameter distribution that is of interest, it is the whole distribution[6]. That is, we want to integrate all our beliefs about the event to form our prediction. Now, if we are also subject to model uncertainty, as we most frequently are, then what is to say that we should not

[6]This is not necessarily true for simpler linear models, where we have an uninformative prior and where the parametric assumptions imply that the parameters are normally distributed - in this case, the mode and the mean of the distribution is the same, and the tails of the distribution will equal out their contributions to the predictions of the expected effect. Then, the distribution of the prediction can also be analytically derived in easier manners than Markov Chain Monte Carlo (MCMC) methods of Bayesian statistics. This follows the general common-sense principle that one should not attempt to solve a specified problem by indirectly solving a harder general problem as an intermediate step.

make use of all our models predictions to help predict a certain event instead of picking a single one out? Talking about model uncertainty, Varian [2014] notes that an important insight from machine learning is that averaging over many small models tends to give better out-of-sample prediction than choosing a single model. Furthermore, he ponders how econometricians have not taken up the habit of using these methods in the statement:

"Ironically, it was recognised many years ago that averages of macroeconomic model forecasts outperformed individual models, but somehow this idea was rarely exploited in traditional econometrics."

[Varian, 2014][7]

In contrast, it might be wise enough to retain model interpretability, which will undoubtedly be squandered when averaging over models, if the goal is extracting causal relationships for inference. However, if the goal is prediction, we should be very wrong to do so.

So, in summary, when we want to predict outcomes the focus should not be on sampling or statistical uncertainty, as in traditional econometrics, but on model uncertainty. The focus on sampling uncertainty over model uncertainty when having vast amounts of data in our sample seems strange. The adaptive methods in machine learning have the aptitude of achieving greater flexibility by specifying a broader class of approximating functions in contrast to the rigid parametric modelling of classical statistics [Cherkassky and Mulier, 2007]. The next section will focus on the bias-variance trade-off, which will help us quantify and find data-driven methods of alleviating model uncertainty, especially when we generalise a model from one sample to an independent test data.

3.2 The bias-variance trade-off

Following Friedman et al. [2009], we first consider an unknown vector of the target variable Y, an unknown matrix of inputs X, and a prediction model $\hat{f}(X)$ that has been estimated from a training set $\mathcal{T}$. The loss function for measuring errors between Y and $\hat{f}(X)$ is denoted by $L(Y, \hat{f}(X))$. For real-valued outcomes, some typical choices are

$$L(Y, \hat{f}(X)) = \begin{cases} (Y - \hat{f}(X))^2 & \text{squared error} \\ |Y - \hat{f}(X)| & \text{absolute error.} \end{cases} \tag{3.2}$$

The test error (also named generalisation error), is the prediction error one makes when generalising one's model over an independent test sample

[7] As an exception, Varian [2014] finds that Bayesian model averaging methods have seen a steady flow of work.

$$\text{Err}_{\mathcal{T}} = \mathbb{E}\left[L(Y, \hat{f}(X))|\mathcal{T}\right] \tag{3.3}$$

where both X and Y are random samples from their joint distribution (Y, X). This is not to be mistaken from the expected prediction error

$$\text{Err} = \mathbb{E}\left[L(Y, \hat{f}(X))\right]. \tag{3.4}$$

The expectation here averages over all that is random, including the randomness in the training set $\mathcal{T}$, that produced $\hat{f}(X)$; which becomes more apparent through the relation $\text{Err} = \mathbb{E}\left[\text{Err}_{\mathcal{T}}\right]$. So Err, consequently, is the error that does not account for the fact that the training set is drawn from a larger, population, data set.

Most statistical methods effectively estimate the expected error, Err, instead of the goal $\text{Err}_{\mathcal{T}}$. It does not seem possible to estimate the conditional error effectively, given only the information in the same training set [Friedman et al., 2009]. Methods to try to do so is developed, such as the cross-validation method of Subsection 3.3.4 or simply retaining a part of the data for testing purposes. They will be discussed later on in this section.

The sample equivalent of the expected prediction error is the training error, and that is the average loss over the training sample

$$\overline{\text{err}} = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}(x_i)). \tag{3.5}$$

Here, as the model becomes more complex, it maps closer to the data and is able to adapt to more complicated underlying structures, but also more underlying noise. Hence, there is a decrease in bias and an increase in variance.

In order to formally investigate this matter further, let us make a decomposition of the error. First, let us assume that $Y = f(X) + \epsilon,\ \epsilon \sim N(0, \sigma_\epsilon^2)$, then we can derive an expression for the expected prediction error, Err, of a regression fit $\hat{f}(X)$ at an input point $X = x_0$ under squared-error loss as

$$\begin{aligned}
\text{Err}(x_0) &= \mathbb{E}\left[(Y - \hat{f}(X))^2|X = x_0\right] \\
&= \mathbb{E}\left[Y^2 + \hat{f}(x_0)^2 - 2Y\hat{f}(x_0)\right] \\
&= \mathbb{E}[Y^2] + \mathbb{E}[\hat{f}(x_0)^2] - 2\mathbb{E}\left[Y\hat{f}(x_0)\right]
\end{aligned}$$

and using the definition of variance[8] to get

$$\begin{aligned}
\mathrm{Err}(x_0) &= \mathrm{Var}\left[Y\right] + \mathbb{E}\left[Y\right]^2 + \mathrm{Var}\left[\hat{f}(x_0)\right] + \mathbb{E}\left[\hat{f}(x_0)\right]^2 - 2\mathbb{E}\left[Y\hat{f}(x_0)\right] \\
&= \sigma^2 + \mathrm{Var}\left[\hat{f}(x_0)\right] + f(x_0)^2 + \mathbb{E}\left[\hat{f}(x_0)\right]^2 - 2f(x_0)\mathbb{E}\left[\hat{f}(x_0)\right] \\
&= \sigma^2 + \mathrm{Var}\left[\hat{f}(x_0)\right] + \left[\mathbb{E}\left[\hat{f}(x_0)\right] - f(x_0)\right]^2 \\
\mathrm{Err}(x_0) &= \text{irreducible error} + \text{variance} + \text{bias}^2. \qquad (3.6)
\end{aligned}$$

Here the first term is the variance of the outcome variable around its true mean $f(x_0)$. This factor cannot be avoided, and in order to make good predictions out-of-sample, we must be careful not to map to close to the outcome variable, as we will be mapping some of this noise in the training data as well. The third term is the squared bias. It is the amount by which we expect our estimate to deviate from the true mean. If we do not make our model complex enough to capture the important relationships in the data, the model will not be generalising well either, as it will introduce too much bias. The second term is the prediction's variance. Variance is the part of prediction error related to the complexity of one's model; formally it is the expected squared deviation of the prediction around its mean under squared-error loss. Then, typically, the more complex we make the model, the lower the bias, the higher the variance [Friedman et al., 2009].

As an example, let us think about the curse of dimensionality, a phenomenon arising from analysing data in high-dimensional spaces, and a term coined by Richard E. Bellman in Bellman [1961]. Let us first consider a unit cube with inputs uniformly distributed within a p-dimensional hypercube in the nearest-neighbour problem. Say we want to capture a fraction r of the observations, then we require an expected range in each of the dimensions in the order $e_p = r^{1/p}$. That is, it is exponentially increasing in the number of dimensions. With p increasing, the range of the variation necessary in each variable is getting larger, and so assigning a value to the nearest neighbour might seem unwise, if the nearest neighbour is not "local" anymore [Friedman et al., 2009].

The idea can be extended to other model types and is definitely also prevalent within the linear framework. So, although one might want to include all possible variables, comprising also generalisations of the linear model such as interactions and powers, in one's model to increase flexibility, it might not be the best way to increase that flexibility at the expense of variance. In the case of housing prices, we could consider looking at only houses with brick walls and houses with wooden walls, and for the sake of argument, let us still go under the belief that

[8] $\mathrm{Var}\left[Y\right] = \mathbb{E}\left[Y^2\right] - \mathbb{E}\left[Y\right]^2$.

each variable is uniformly distributed between 0 and 1. The range in each variable required in order to capture 10% of the data in order to form a local average is 31.5%[9], which is not an insurmountable requirement. But if we also consider concrete walls, light concrete walls or built with half-timbering, and we then include size variables, such as size of building ground, total building area on all floors or the size of the ground, and we could include location variables, other house variables or somewhat miscellaneous variables, then it is another story. Let us say these variables number up to $p = 150$, then the range required in each variable will be 98.5%[10]. In the example of house prices given by Mullainathan and Spiess [2017], they also rhetorically ask why interactions between variables should not also be included as the effect of the number of bedrooms may well depend on the base area of the unit, and the added value of a fireplace may be different depending on the number of living rooms. However, simply including all pairwise interactions would be infeasible as it produces more regressors than possible to fit[11]; in our case that would amount to 11325[12], but other interactions or functional forms of specific variables could also be of interest. In such situations the available data quickly becomes sparse.

Some machine learning models automatically look for complex relationships between the independent variables in different ways according to the model class and are designed not to overfit the data - to fit too tightly to the in-sample training set, such as to fit also the idiosyncratic errors contained in it. A distinction between classical statistics and machine learning can be made here; the goal of learning (estimation) within statistical modelling is accurate identification of the unknown system, whereas under predictive learning, the goal is accurate imitation of the system's output [Cherkassky and Mulier, 2007]. This means that, for example, machine learning does not have to include both of two highly correlated variables, which would increase variance, and it does not necessarily have to choose between them either - in fact, it can remain rather agnostic about that kind of choice.

These machine learning models are built from the idea of regularising the model in order to minimise out-of-sample error. Machine learning models, therefore, include parameters to *tune* model complexity according to a given appropriate loss function. This tuning parameter should be based on an estimate of prediction error [Friedman et al., 2009], and there are several ways to

[9] $e_2 = 0.1^{1/2} = 0.3162$.

[10] $e_{100} = 0.1^{1/150} = 0.9848$.

[11] When having a large amount of observations as well, we can even run into problems of too low computer memory since having 11325 variables and 179952 observations will exhaust all computer memory plus some. As some of the variables are one-hot encoded, such as the walling in the previous example, they are mutually exclusive, and as they are also dummy variables, then they cannot be included with squares. However, even with these exceptions, the computer memory is still exhausted.

[12] From the formula expressing the sum of a series, $S = \frac{N(N+1)}{2} + N$; in this case $\frac{150(150+1)}{2} + 150 = 11325$.

do so. The next subsection is dedicated to an overview and discussion of these estimates.

3.3 The estimation of prediction error

The estimation of prediction error can be handled in several ways according to practical preference. Direct estimates of generalisation error, such as cross-validation or bootstrap methods, are computationally heavy and require the model to be estimated a number of times. As an alternative, there are several statistics based in the in-sample error, and although this error is not of direct interest since future values of the features are unlikely to coincide with the training set values, these methods are convenient and often lead to effective model selection. The reason is that the relative rather than absolute size of the error is most important when selecting models [Friedman et al., 2009]. The downside to this, which will be seen later, is that it requires us to know the degree of freedom, which is awkward in non-parametric or regularised models. Nonetheless, it is a good starting point for the introduction of prediction error estimates.

So, in addition to the error statistics mentioned in the previous subsection, we should also mention the in-sample error

$$\mathrm{Err}_{\mathrm{in}} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{Y^0}\left[L(Y_i^0, \hat{f}(x_i))|\mathcal{T}\right], \tag{3.7}$$

where the Y^0 notation indicates that we observe N new response values at each of the training points x_i, $i = 1, 2, ..., N$, and that is what the expectation is over; so that we in this procedure have averaged out idiosyncratic error related to the outcome variable as N goes to infinity according to the law of large numbers [Friedman et al., 2009]. We can then define the *optimism* as

$$\mathrm{op} \equiv \mathrm{Err}_{\mathrm{in}} - \overline{\mathrm{err}}. \tag{3.8}$$

This value is typically positive as $\overline{\mathrm{err}}$ is usually biased downwards as an estimate of the prediction error because the model frequently fits the in-sample irreducible error as well. In the estimate of $\overline{\mathrm{err}}$ this will show through a lower predicted error. Now, the average optimism is the expectation of the optimism over several training sets, much like the relationship between the expected prediction error and the actual prediction error,

$$\omega \equiv \mathbb{E}_y(\mathrm{op}). \tag{3.9}$$

Here the predictors in the training set are fixed, and the expectation is over the training set outcome values. Hence, in relation to $\mathrm{Err}_{\mathcal{T}}$, we do not observe a new set of outcome values, y,

to each x_i, but instead only have the outcome values within the training set, and so the notation $\mathbb{E}_y[\cdot]$ is used.

For squared-error loss, the average optimism is derived in the Appendix section A.1, and it is given as

$$\omega = \frac{2}{N}\sum_{i=1}^{N}\text{Cov}(\hat{y}_i, y_i). \tag{3.10}$$

Thus, the amount by which $\overline{\text{err}}$ underestimates the true error depends on how strongly a single observation y_i affects its own prediction.

From Equation (3.8) and the estimate of optimism in Equation (3.10) an obvious estimate of Err_{in} is

$$\mathbb{E}_y\left[\text{Err}_{\text{in}}\right] = \mathbb{E}_y\left[\overline{\text{err}}\right] + \omega. \tag{3.11}$$

This expression simplifies if $\hat{y}_i$ is obtained by a linear fit with d inputs [Friedman et al., 2009], which reflects the degree of freedom in the model. For the additive error model $Y = f(\boldsymbol{X}) + \epsilon$ then $\omega = \frac{2}{N}d\sigma_\epsilon^2$, and so

$$\mathbb{E}_y\left[\text{Err}_{\text{in}}\right] = \mathbb{E}_y\left[\overline{\text{err}}\right] + 2\frac{d}{N}\sigma_\epsilon^2 \tag{3.12}$$

Thus, we can see that the optimism increases linearly with the number of inputs, but decreases as the training sample size increases. A number of different, frequently applied, methods rely on Equation (3.11), and they will be discussed in the next section.

3.3.1 Estimates of in-sample prediction error

The general form of the in-sample estimate is

$$\widehat{\text{Err}}_{\text{in}} = \overline{\text{err}} + \hat{\omega}. \tag{3.13}$$

When d parameters are fit under squared-error loss, we have a version of the so-called Mallows' $\mathcal{C}_p$ statistic

$$\mathcal{C}_p = \overline{\text{err}} + 2\frac{d}{N}\hat{\sigma}_\epsilon^2,$$

where $\hat{\sigma}_\epsilon^2$ is an estimate of the noise variance, which should be obtained from a low-bias model [Friedman et al., 2009]. Motivated by the unfortunate subjective judgment required in the for-

mulation of $\hat{\sigma}_\epsilon^2$, Akaike [1974] proposed another, now frequently applied, method of estimating Err_{in}: Akaike information criterion (AIC), which is more generally applicable than Mallows' $\mathcal{C}_p$. AIC relies on a asymptotic relation for the average log-likelihood of all observations of Y similar to Equation (3.12). When N is increased indefinitely, the average tends, with probability 1, to

$$\frac{1}{N}\sum_{i=1}^{N}\log[\mathbb{P}_{\hat{\theta}}(y_i)] = \int \mathbb{P}(Y)\cdot\log[\mathbb{P}_{\hat{\theta}}(Y)]\cdot dY$$

where $\log[\mathbb{P}_{\hat{\theta}}(Y)]$ is a maximised log-likelihood of a parametric model, and $\mathbb{P}(Y) = \mathbb{P}_{\theta_0}(Y)$ is the assumed to be the true density of Y in the same parametric family with θ_0 as the true parameter values, and the existence of the integral is assumed. The difference $I[\mathbb{P}(Y);\mathbb{P}_\theta(Y)] = \frac{1}{N}\sum_{i=1}^{N}\log[\mathbb{P}(y_i)] - \frac{1}{N}\sum_{i=1}^{N}\log[\mathbb{P}_\theta(y_i)]$ is known as the Kullback-Leibler mean information for discrimination between $\mathbb{P}(Y)$ and $\mathbb{P}_\theta(Y)$ and is positive unless $\mathbb{P}(y_i) = \mathbb{P}_\theta(y_i)$ holds almost everywhere, which is known as Gibbs inequality [Akaike, 1974]. Under the situation where $\mathbb{P}(Y) = \mathbb{P}_\theta(Y)$, then a perturbation in $I[\mathbb{P}(Y);\mathbb{P}_\theta(Y)+\triangle]$ admits an approximation

$$I[\mathbb{P}(Y);\mathbb{P}(Y)+\triangle] \approx \frac{1}{2}\left[(\triangle\theta)^T J(\theta)(\triangle\theta)\right], \tag{3.14}$$

where $J(\theta)$ is the Fisher information matrix. Notice also that if the likelihood of θ is restricted to lie in a lower dimensional subspace than the true θ_0, and so cannot encompass it, then for the maximum likelihood estimate $\hat{\theta}$ it can be shown that $N\left[(\hat{\theta}-\theta)J(\theta)(\hat{\theta}-\theta)\right]$ for sufficiently large N is approximated under certain regularity conditions by a chi-squared distribution with the degree of freedom equal to the dimension of the restricted parameter space, d. Thus, it holds that

$$2N\mathbb{E}\left[I[\mathbb{P}_{\theta_0}(Y);\mathbb{P}_{\hat{\theta}}(Y)]\right] \approx N\left[(\theta-\theta_0)^T J(\theta)(\theta-\theta_0)\right] + d$$

Moreover, using $2\left(\sum_{i=1}^{N}\log\left[\mathbb{P}(Y_i)\right] - \sum_{i=1}^{N}\log\left[\mathbb{P}_{\hat{\theta}}(y_i)\right]\right)$ as an estimate of $N\left[(\theta-\theta_0)^T J(\theta)(\theta-\theta_0)\right]$, and adding d to the approximation in order to adjust for the downward bias introduced by the replacement of θ with $\hat{\theta}$, the maximum likelihood estimate, yields the result when isolating $\sum_{i=1}^{N}\log\left[\mathbb{P}(Y_i)\right]$. Hence, with the objective of minimising the Kullback-Leibler divergence between our model, assuming the correct model is within our specification, and the true data generating process, AIC is defined as

$$\text{AIC} \equiv -\frac{2}{N}\sum_{i=1}^{N}\log\left[\mathbb{P}_{\hat{\theta}}(y_i)\right] + 2\cdot\frac{d}{N}. \tag{3.15}$$

For the Gaussian model and with variance $\sigma_\epsilon^2 = \hat{\sigma}_\epsilon^2$ assumed known, the AIC is equivalent to Mallows' $\mathcal{C}_p$.

For non-linear and other complex models, we need to replace d by some measure of model complexity, and these are discussed in Subsection 3.4. If we want to use this estimate, then say we have a set of models $f_\alpha(\boldsymbol{X})$ indexed by a tuning parameter α, and denote the training error by $\overline{\text{err}}(\alpha)$ and the number of parameters for each model by $d(\alpha)$. Then for this set of models we have

$$\text{AIC}(\alpha) = \overline{\text{err}}(\alpha) + 2 \cdot \frac{d(\alpha)}{N} \hat{\sigma}_\epsilon^2.$$

The function $\text{AIC}(\alpha)$ is then an estimate of the test error curve, and we can tune α to find an $\hat{\alpha}$ that minimises it [Friedman et al., 2009]. A note of caution is that if we have a total of p possible input variables, and we choose the best-fitting linear model with $d < p$ inputs, the optimism will exceed the otherwise correctly estimated optimism of $(2d/N)\sigma_\epsilon^2$, because the effective number of parameters fit is more than d [Friedman et al., 2009].

Other methods of model selection are featured in Subsection A.2 of the appendix. Here we also see the Bayesian Information Criterion (BIC), that, although not motivated by the estimation of Err_{in}, is similar to AIC. Instead, BIC is motivated by the posterior odds of two competing models, which is a very Bayesian way to look at the problem of model selection. Unlike AIC, BIC is asymptotically consistent, and for $N > e^2 \approx 7.4$ BIC tends to penalise complex models more heavily than AIC. However, for finite samples, BIC often chooses models that are too simple, because of its heavy penalty on complexity [Friedman et al., 2009].

Now we will shift the focus to the out-of-sample prediction error in the next subsections.

3.3.2 Setting some data aside for testing purposes

If we are in a data-rich situation, the best approach is to randomly divide the data set into three parts: a training set, a validation set, and a test set [Friedman et al., 2009]. The training set is used to fit the models; the validation set is used to estimate prediction error for model selection; the test set is used for assessment of the generalisation error of the final chosen model. It is difficult to give a general rule on how to choose the number of observations in each of the three parts, as this depends on the signal-to-noise ratio in the data and the training sample size. However, a typical split is 50% for training, and 25% each for validation and testing [Friedman et al., 2009].

Next are some methods of efficient re-use of the sample in order to both estimate the model, tune the parameters and estimate the prediction error. Besides their use in model selection, it is also examined to what extent these methods provide a reliable error of the final chosen model as

these would most likely be downward biased estimates.

3.3.3 Bootstrapping

Bootstrapping is a familiar tool for assessing the statistical errors of any model. It relies on sampling with replacement, and the idea is to create an approximating distribution for the parameter of interest by assuming the sample is the whole population, and then iteratively sample with replacement from that distribution [Efron, 1979]. For example, say we create $B = 100$ samples of the same size as the original sample from our original sample, we can estimate a model $f(\boldsymbol{X}; \hat{\theta})$ B times. The distribution of the models' predictions can be used to make statistical inference about the prediction accuracy, and the distribution of the parameter estimates of the model fits can be used to make statistical inference about their distributions. Of course, as a means for estimating the conditional error $\text{Err}_{\mathcal{T}}$, the method might be underestimating the actual error, as individual predictions can be helping to predict themselves; a positive optimism as shown in Equation (3.10). This problem exists in any use of this method, or any efficient re-sampling scheme, but although it provides inferior model assessment qualities, it has some good qualities when it comes to smaller samples and overall model selection. More formally, we follow Friedman et al. [2009] to state the estimate of the error given by bootstrapping

$$\widehat{\text{Err}}_{\text{boot}} = \frac{1}{B}\frac{1}{N}\sum_{b=1}^{B}\sum_{i=1}^{N} L(y_i, \hat{f}^{*b}(x_i)), \tag{3.16}$$

where $\hat{f}^{*b}(x_i)$ is the predicted value at x_i from the model fitted to the bth bootstrap data set.

As means of mitigating the problem of positive optimism in the bootstrapping re-sampling scheme, there are several corrections to be made in the bootstrapping scheme. Firstly, the leave-one-out bootstrap estimate keeps track of all those samples drawn that do not include observations i, and then use them to estimate the prediction error of observation i. Secondly, the average number of distinct observations in each bootstrap sample is $0.632 \cdot N$ which creates a bias; the ".632 estimator" is designed to alleviate this bias and does so by a weighted average of $\overline{\text{err}}$ and the leave-one-out bootstrap estimate. Further information, including the definition of these estimates, can be found in the Appendix A.3.

In contrast, cross-validation explicitly uses non-overlapping data for the training and test samples, although when selecting and tuning the model this distinction becomes less apparent. The method of cross-validation is described in the next subsection.

3.3.4 Cross-validation

According to Friedman et al. [2009], cross-validation is the simplest and most widely used method of estimating prediction error. Originally, Stone [1974] creates N data sets by leaving one observation of the original data set out of each data set, and then fits the model on each data set to predict the left out observation. This method is subsequently being named leave-one-out cross-validation (LOOCV). From those prediction errors, Stone [1974] can make an estimate of the average generalisation error and therefore tune the shrinkage parameter. Golub et al. [1979] motivate the invention of generalised cross-validation by the non-necessity of estimating $\hat{\sigma}_\epsilon^2$, but also recognises some sensitivities to extreme values in Allen's PRESS (similar to LOOCV in Stone [1974]). Golub et al. [1979] then derive a rotation-invariant form of cross-validation, which, simply put, is a reweighting of the errors in each data set used to derive the statistic to tune from.

However, although it is nice that LOOCV is approximately unbiased for the true (expected) prediction error, this means that it can have high variance since the N "training sets" are so similar to one another. Instead, K-fold cross-validation splits the data into K non-overlapping, roughly equal-sized parts, and fits the model to $K-1$ parts of the data to predict the kth part for $k = 1, 2, \dots K$. We are then able to average these predictions over the K parts to give an estimate of prediction error [Geisser, 1975]. Following Friedman et al. [2009], we formally let $\kappa : \{1, \dots, N\} \mapsto \{1, \dots K\}$ be an indexing function indicating the partition to which observation i is allocated to by randomisation. We denote the fitted function with the kth part removed as $\hat{f}^{-k}(x)$, and we can define the cross-validation estimate of prediction error as

$$\mathrm{CV}_K(\hat{f}) = \frac{1}{N} \sum_{i=1}^{N} L(y_i, \hat{f}^{\kappa(i)}(x_i)). \tag{3.17}$$

Here we are able to decrease variance, but we also increase bias. So in the choice of K, we should have this trade-off in mind as suggested in Kohavi et al. [1995]. Notice, also, that the case $K = N$ is the LOOCV cross-validation estimate. Friedman et al. [2009] compare different values of K to see how well they estimate Err and $\mathrm{Err}_{\mathcal{T}}$ respectively. They find that CV_{10} estimates both Err and $\mathrm{Err}_{\mathcal{T}}$ better than LOOCV, and that both are approximately unbiased estimates of Err but fail to predict the generalisation error very well. Furthermore, they suggest that $K = 5$ or $K = 10$ are both common choices, and that one should look at variances between the data sets to pick one; the higher the variance between data sets, the higher the number of folds you choose.

In order to make the methods of estimating prediction error without re-sampling relevant even in highly complex and non-linear model frameworks, the next subsection is dedicated to the effective

number of parameters used.

3.4 The effective number of parameters

The concept of "number of parameters" can be generalised, which is especially necessary for models where regularisation is used in the fitting [Friedman et al., 2009]. Firstly, within linear model frameworks which include features based on a derived basis set or smoothing methods using quadratic shrinkage (see Subsection 4.1), we can write a linear fitting as

$$\hat{y} = \boldsymbol{S}y,$$

where $\boldsymbol{S}$ is a $N \times N$ matrix depending on the input vectors x_i but not on y_i. The effective number of parameters is then defined as

$$\mathrm{df}(\boldsymbol{S}) = \mathrm{trace}(\boldsymbol{S}),$$

e.g. the sum of the diagonal elements of $\boldsymbol{S}$. If Y arises from an additive-error model $Y = f(X) + \epsilon$ with $\epsilon \sim N(0, \sigma_\epsilon^2)$, then one can show that $\sum_{i=1}^{N} \mathrm{Cov}(\hat{y}_i, y_i) = \mathrm{trace}(\boldsymbol{S})\sigma_\epsilon^2$, which resembles the relation used in Equation (3.12) but with $\mathrm{trace}(\boldsymbol{S})$ in place of d [Friedman et al., 2009]. This motivates the more general definition

$$\mathrm{df}(\hat{y}) = \frac{\sum_{i=1}^{N} \mathrm{Cov}(\hat{y}_i, y_i)}{\sigma_\epsilon^2}.$$

In the case of models like neural networks described in Subsection 5.3, where we minimise an error function $R(w)$ subject to weight decay regularisation $\alpha \sum_m w_m^2$, the effective number of parameters has the form

$$\mathrm{df}(\alpha) = \sum_{m=1}^{M} \frac{\theta_m}{\theta_m + \alpha},$$

where θ_m are the eigenvalues of the Hessian matrix $\partial^2 R(w)/\partial w \partial w^T$. The result follows from a second-order Taylor approximation to the error function at the solution [Bishop, 1995].

This type of measurement of the effective number of parameters is very useful, but the method described here is not fully general, as it is limited to a narrow class of models. The Vapnik-Chervonenkis (VC) theory provides such a general measure of complexity. It gives associated analytical bounds on the optimism, and although it will not be implemented in this thesis and has its hardships with regression, it is worthwhile to describe in the next subsection.

3.4.1 Vapnik-Chervonenkis Dimension

In order to follow Devroye et al. [2013] in showing an influential result in VC-theory using the VC-dimension in the next subsection, we must start out by changing our mindset from function classes, $\mathcal{F}$, to collections of measurable sets, $\mathcal{A}$. We start with a formal definition of a shatter coefficient.

Definition 1. (Shatter coefficient)

Let $\mathcal{A}$ be a collection of measurable sets. Then for any arbitrary set $\mathcal{C} \subset \mathcal{X} \in \{\mathbb{R}^d\}^N$, let $N_{\mathcal{A}}(\mathcal{C})$ be the number of different sets in $\{\mathcal{C} \cap A; A \in \mathcal{A}\}$, the nth shatter coefficient of $\mathcal{A}$ is $s(\mathcal{A}, N) = \max_{\mathcal{C} \in (\mathbb{R}^d)^N} N_{\mathcal{A}}(\mathcal{C})$

Now, less formally, we note that the maximal number of shatter coefficients is bounded from above by $s(\mathcal{A}, N) \leq 2^N$, which makes sense in a combinatorial way since it is the maximal number of combinations one can pick out either 1 or 0. Then, if $s(\mathcal{A}, k) < 2^k$ for some integer k, then $s(\mathcal{A}, N) < 2^N$ for all $N > k$, and so if we can find a smaller number of different sets in $\mathcal{A}$ to shatter $\mathcal{C}$, then the maximal number of shatter coefficients is clearly smaller than 2^N. The VC-dimension is then the largest integer $k \geq 1$ for which $s(\mathcal{A}, k) = 2^k$ is denoted $\text{VCdim}(\mathcal{A})$. Devroye et al. [2013] also states that, by definition, if $s(\mathcal{A}, N) = 2^N \; \forall \, N$, then $\text{VCdim}(\mathcal{A}) = \infty$. Having defined the shatter coefficient and the VC-dimension using set-families, we turn our attention to the VC-dimension in function classes.

Suppose we have a class of functions $\mathcal{F} = \{f(\alpha) : \mathcal{X} \mapsto \mathcal{Y}\}$ indexed by a parameter vector α, with $\mathcal{X} \in \mathbb{R}^p$. Assume for now that $f(\cdot)$ is an indicator function so that it takes either the value of 0 or 1. If $\alpha = (\alpha_0, \alpha_1)$ and $f(\cdot)$ is the linear indicator function $I(\alpha_0 + \alpha_1^T x > 0)$, then it seems reasonable to say that the complexity of the class $f(\cdot)$ is the number of parameters $p+1$ [Friedman et al., 2009]. But if we instead consider another function $f(\cdot) \in \mathcal{F}$, where $f(\alpha) = I(\sin\alpha \cdot x)$ and α is a scalar coefficient, then, because of the wiggly form of the function that gets even rougher as the frequency, α, increases, we cannot conclude that this function is less complex than the linear indicator function. This is the reasoning that leads us to the definition of the VC-dimension within function classes, where the translation of sets to function classes is straightforward: a given classification function $f(\cdot)$ with parameter vector θ is said to shatter a set $\mathcal{C} \subset \mathcal{X}$ if, for all assignments of classifications to those points, there exist a θ such that the function $f(\cdot)$ makes no errors [Friedman et al., 2009]. The VC-dimension is then defined as:

Definition 2. (VC-dimension)

The VC-dimension of the class $\mathcal{F}$ is defined to be the largest size of a set $\mathcal{C} \subset \mathcal{X}$ that can be shattered by members of $\mathcal{F}$. If $\mathcal{F}$ can shatter sets of arbitrarily large size we say that $\mathcal{F}$ has

infinite VC-dimension.

That is, as an example, say $\mathcal{F}$ is a class of threshold function over $\mathbb{R}$, and say we have an arbitrary set of one point $\mathcal{C} = \{c_1\}$, then $\mathcal{F}$ shatters $\mathcal{C}$; and therefore the VC-dimension of $\mathcal{F}$ is greater than or equal to 1 ($\text{VCdim}(\mathcal{F}) \geq 1$). But we can also show that for an arbitrary set $\mathcal{C} = \{c_1, c_2\}$ with $c_1 \leq c_2$, $\mathcal{F}$ does not shatter $\mathcal{C}$, where we must note the importance of the possible equality of c_1 and c_2. We can therefore conclude that $\text{VCdim}(\mathcal{F}) = 1$ [Shalev-Shwartz and Ben-David, 2014].

Next, we generalise the concept of VC-dimension to *real-valued*, and therefore unbounded, loss functions. Consider a set of real-valued functions $L(Y, \hat{f}(X))$ bounded by some constants $A \leq L(Y, \hat{f}(X)) \leq B$. For each function, we can form the indicator function showing whether $L(Y, \hat{f}(X))$ is greater or smaller than some level β ($A \leq \beta \leq B$):

$$I(y, \boldsymbol{Z}; \beta) = I[L(Y, \hat{f}(X)) - \beta > 0].$$

The VC-dimension of a set of real-valued functions $L(Y, \hat{f}(X))$ is, by definition, equal to the VC-dimension of the set of indicator functions with parameter β [Evgeniou et al., 2000, Cherkassky and Mulier, 2007]. According to personal communication between Cherkassky, Mulier and Vapnik, one can use

$$h \approx h_f,$$

where h is the VC-dimension of $L(Y, \hat{f}(X))$ and h_f is the VC-dimension of $\hat{f}(X)$, for practical regression applications using squared-error loss [Cherkassky and Mulier, 2007].

After having established the notion of the VC-dimension, we can use it construct an estimate of the generalisation prediction error in different ways. One can prove results about the optimism of the training error when using a class of functions and within specific loss function. The next subsection is dedicated to these bounds and discusses another method of model selection based on the bounds.

3.5 Upper bounds on the rate of uniform convergence

To start this section off, we look at the beginning of VC-theory which is structured around the idea of empirical risk minimisation (ERM)[13] mostly developed by Vapnik [Cherkassky and Mulier, 2007]. The bounds evaluate the difference between the (unknown) true risk and the known empirical risk as a function of the sample size N, properties of the unknown distribution $\mathbb{P}(Z)$,

[13]The risk part of ERM is another way to view loss. In fact, it is the integral with respect to the joint distribution function of the iid realisation of the outcome variable and the predictors of the loss function.

properties of the loss function and properties of approximating functions [Cherkassky and Mulier, 2007]. Within regression the bounds on the true function or the additive noise are not known, so we cannot provide finite bounds for such loss function as there is always a small probability of observing enormous output values. This is why the first developments of generalisation bounds were within classification and primarily binary classification. An example of such a bound for binary classification, which draws on a result in the original paper Vapnik and Chervonenkis [1971], is given in [Devroye et al., 2013]. It gives some general bounds on sums of random variables in specific sets. The theorem is stated as such: for any probability measure of the random variables $\mathbb{P}(Z)$ and class of sets $\mathcal{A}$, and for any n and $\epsilon > 0$,

$$\mathbb{P}\left\{\sup_{A\in\mathcal{A}}\left|\frac{1}{N}\sum_{i=1}^{N} I_{(Z_i\in A)} - \mathbb{P}(Z\in A)\right| > \epsilon\right\} \leq 8s(\mathcal{A},N)e^{-N\epsilon^2/32} \tag{3.18}$$

where $Z_1,\ldots,Z_N$ are independent identically distributed random variables in $\mathbb{R}^d$. It is a generalisation of the Glivenko-Cantelli theorem, stating uniform almost sure convergence of the empirical distribution function, the one we have estimated using our data within a specific function, to the true one [Devroye et al., 2013]. Thus, this theorem gives us an upper bound within a certain probability threshold on the loss incurred from fitting our model rather than having the true data generating process.

However, for real-valued loss functions, we need a description of the tails of the distribution, namely the probability of observing large values. For distributions with so-called "light tails", a fast rate of convergence is possible. For such a distribution, the bounds on generalisation are given in Cherkassky and Mulier [2007]. They find that the bound that holds with probability at least $1-\eta$ simultaneously for all loss functions (including the one that minimises the empirical risk) and all members of $\mathcal{F}$ is

$$\mathrm{Err}_{\mathcal{T}} \leq \frac{\overline{\mathrm{err}}}{(1-c\sqrt{\epsilon})_+} \tag{3.19}$$

where

$$\epsilon = a_1\frac{h[\log(a_2N/h)+1]-\log(\eta/4)}{N},$$

and

$$0 < a_1 \leq 4,\ 0 < a_2 \leq 2,$$

and also $\mathrm{VCdim}(\mathcal{F}) = h$. Furthermore, c depends on the tails of the distribution, and for most practical regression applications we can safely assume $c = 1$ [Cherkassky and Mulier, 2007]. For regression they suggest practical values of the constants $a_1 = a_2 = 1$. They also give an alternative

practical bound for regression, arguing that the confidence level $1-\eta$ should depend positively on the sample size N, and using the values of the constants as given before:

$$\text{Err}_{\mathcal{T}} \leq \overline{\text{err}} \left(1 - \sqrt{\rho - \rho\log\rho + \frac{\log N}{2N}}\right)_{+}^{-1},$$

where $\rho = \frac{h}{N}$, which is free of tuning constants. These bounds suggest that optimism increases with complexity (h) and decreases with N, which is what we would have thought a priori. The result given in Equation 3.19 is strong; rather than given the expected optimism for each fixed function, they give probabilistic upper bounds for all functions in a class $\mathcal{F}$, and hence allow for searching over the class [Friedman et al., 2009]. It could be another criterion for model selection to pick the model with the least probabilistic upper bound of the optimism, which is exactly what Vapnik's structural risk minimisation of Vapnik and Chervonenkis [1974] does. Furthermore, Cherkassky and Mulier [2007] note that in situations when the VC-dimension can be accurately estimated, the analytic bounds of SRM may provide better complexity control than resampling approaches. However, as found in [Friedman et al., 2009], this often is not the case, and so this upper bound view of complexity control will not be pursued in this thesis.

4 Reducing Uncertainty and Variance

The former section described methods of estimating generalisation error or other forms of error for model selection, which are both relevant within functional classes and outside functional classes. This section will show us how to use that information to regularise and shrink our models to give better generalisation abilities and possibly to make a simpler (more parsimonious) model. The separate goal of a simpler model stems partly from the desire to make an interpretable model, but also for its own sake. This is something that is often motivated by Occam's razor (such as in Cherkassky and Mulier [2007], Schapire and Freund [2012]), but to do it differently let us follow the tale of the Columbus egg: after having returned from discovering the Americas, Columbus was told that the discovery was inevitable and no great accomplishment. After hearing this Columbus challenges his critics to make an egg stand on its tip. His critics find this task hard and give up, and then Columbus does so himself by tapping the egg on the table to flatten its tip and thereby make it stand. The analogy to our problem is, therefore, that we should not necessarily seek to make the problem harder and more complex if a simpler solution exists. The section will also consider ways of reducing model uncertainty, such as to average models from several functional classes to enhance predictive accuracy. Furthermore, this section will introduce Bayesian methods as alternative means of model averaging according to the posterior probability of the models.

4.1 Regularisation

The regularisation approach provides a formalism for adjusting the complexity of approximating functions to fit finite data. It takes starting point in a specific adaptive (flexible) model and then by reducing the number of parameters, by shrinking them, by constraining the optimisation procedure, or by model specific regularisations seeking to reduce the variance of the predictions for better generalisation ability. The model specific regularisations can be choosing the number of hidden layers (in a neural network) or picking the number of nearest neighbours (K-nearest neighbours). As an example, one form of this is to retain only a subset of the predictors and discarding the rest. This is the way best subset selection produces a model that is both interpretable and possibly has lower prediction error than a full model. However, also here there is a price paid in variance for selecting the best subset of each size of each model; and so other methods constraining the ability to test all models against each other are made - such as forward-stepwise and forward-stagewise regression. Although we will not use or go more in-depth with these models, the procedure of forward-stepwise regression will be briefly explained: first, it starts with the intercept at the mean of the outcome variable and all other parameters at zero, and then sequentially add into the model the predictor that improves fit the most [Friedman et al., 2009].

This procedure of only looking at what improves fit the most in the next step of the algorithm grants it the general adjective of being a *greedy* algorithm.

Within the generalised linear model framework, where we as usually seek an estimator that minimises a specified loss function, we can apply a penalty term

$$L(y, f(\boldsymbol{X}; \beta)) + \lambda J(\beta),$$

where $\lambda \geq 0$ is a tuning parameter, $J(\cdot)$ is the penalty functional, and $\beta \in \mathbb{R}^p$ contains all of our parameter estimates besides the intercept. Here we can fit the model using different values of the tuning parameter and choose between the different models $f_\lambda(\boldsymbol{X}; \hat{\beta})$ based on the predictive accuracy as estimated in the previous section. A very general form of regularisation is the ℓ_q penalty function

$$J(\beta) = ||\beta||_q^q,$$

where $||\cdot||_q^q$ is the q-norm to the qth power, i.e. $||\beta||_q = \left(\sum_{j=1}^p |\beta_j|^q\right)^{1/q}$. This penalisation term leads to the minimisation problem for β giving the *bridge* estimate

$$\tilde{\beta} = \arg\min_{\beta} \left\{ L(y, f(\boldsymbol{X}; \beta)) + \lambda ||\beta||_q^q \right\}. \tag{4.1}$$

To see how different values of q affect the values of the parameters in the case of two inputs, consider Figure 4.1 from Friedman et al. [2009] that show the contours of a constant value of $\sum_{j=1}^2 |\beta_j|^q$.

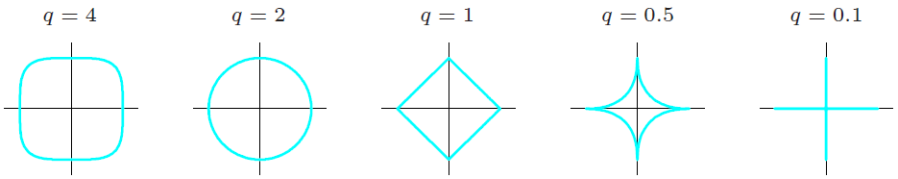

Figure 4.1: Contours of a constant value of $\sum_{j=1}^2 |\beta_j|^q$

Consider $q = 1$ as an example, here we are more likely to shrink the coefficients a lot or set them equal to zero compared with $q = 2$, and with $q < 1$ we are more and more likely to set variables to zero. The case of $q = 0$ corresponds to variable subset selection, which selects a subset of the total number of possible variables to keep and set the rest to zero, and so creates a *sparse* model. We can also view these estimates as Bayes estimates using different prior densities

yielding the same results[14]. In general format, the bridge estimator can be viewed as the Bayes posterior mode under the prior $\mathbb{P}(\beta|\lambda;q) \propto e^{(-\lambda||\beta||_q^q)}$, where $q = 1$ corresponds to a Gaussian prior and the $q = 1$ corresponds to a Laplacian prior under relevant scaling of the tuning parameter, λ, which can be seen in Appendix A.4.

Frank and Friedman [1993] are credited with the invention of the bridge regression technique, while they did not solve for the estimator for any given q, they suggested that it would be desirable to optimise the parameter q [Fu, 1998].

Although it is considered estimating q from data, it might also seem counterproductive, as the methods are deployed to decrease variance, and estimating q will definitely increase variance, and so Zou and Hastie [2005] find that it does not improve predictions compared to $q = 1$ or $q = 2$. These two frequently used special cases, namely $q \in \{1, 2\}$, is what we will look into in the following two subsections. In Subsection 4.1.3, we will also briefly discuss a combination of the two.

4.1.1 Ridge regression (ℓ_2 penalty)

First, let us consider the ridge regression of Hoerl and Kennard [1970], who employ the ℓ_2 penalty term, and let us also consider squared-error loss within a multivariate regression framework as they do, then we have the following problem

$$\hat{\beta}^{\text{ridge}} = \arg\min_{\beta} \left\{ ((y - 1_N\beta_0) - \boldsymbol{X}\beta)^T ((y - 1_N\beta_0) - \boldsymbol{X}\beta) + \lambda||\beta||_2^2 \right\}, \tag{4.2}$$

where 1_N indicates a vector of ones with length N. This minimisation problem admits the neat closed form solution

$$\hat{\beta}^{\text{ridge}} = \left(\boldsymbol{X}^T\boldsymbol{X} + \boldsymbol{I}_p\lambda\right)^{-1}\boldsymbol{X}^T y, \tag{4.3}$$

where $\boldsymbol{I}_p$ is the $p \times p$ identity matrix. Here we clearly see the distinction that shrinks the estimates compared to the usual OLS estimate

$$\hat{\beta}^{\text{ols}} = \left(\boldsymbol{X}^T\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T y.$$

Notice how in Equation (4.2) β_0 is not contained in the penalty functional; penalisation of the intercept would make the procedure depend on the origin chosen for y [Friedman et al., 2009]. Also, a minor problem with ridge regression is that the estimates of β are not scale invariant - if we measure weight in kilogram rather than gram our standard OLS estimate would be a factor

[14] When using either the mean or mode of the posterior distribution of the variable in the Bayes estimate.

1000 larger, but if we use ridge regression, we will penalise the estimate measured in kilogram much more. A simple fix to this problem is to standardize each of the inputs such that each input is normally distributed with zero mean and unit variance, i.e. $\mathcal{N}(0,1)$. When we have standardized inputs, we can estimate β_0 by $\bar{y}$, the mean of the outcome variable.

In the case of orthonormal inputs (independence between the independent variables[15]), we have a simple relationship between the ridge estimate and the OLS estimate

$$\hat{\beta}^{\text{ridge}} = \frac{\hat{\beta}^{\text{ols}}}{1+\lambda},$$

where the effect of adding the penalisation is easily seen. We can also see that ridge regression does not produce sparse models, as it does not shrink estimators to zero. However, it still reduces the effective degrees of freedom defined in Subsection 3.4 as exemplified in the following figure found in Friedman et al. [2009].

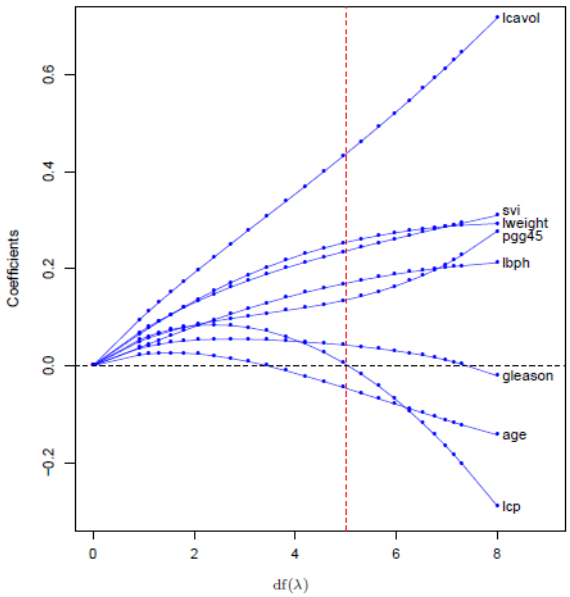

Figure 4.2: Profiles of ridge coefficients as the tuning parameter λ is varied. Coefficients are plotted against $\text{df}(\lambda)$, the effective degrees of freedom. A vertical line is drawn at $\text{df} = 5$, the value chosen by cross-validation.

Another frequently deployed penalty function is the Least Absolute Shrinkage and Selection Operator (LASSO), which does make a more sparse model. This type of penalty will be the

[15]One could also make a singular value decomposition of the input matrix with a relevant rescaling to induce orthonormality and find a similar relationship between ridge regression and OLS, where the individual $\hat{\beta}^{\text{ridge}}$ estimate is scaled by the magnitude of the singular value d_j corresponding to their (normalised) principal component u_j (vector), hence $\boldsymbol{X}\hat{\beta}^{\text{ridge}} = \sum_{j=1}^{p} u_j \frac{d_j^2}{d_j^2+\lambda} u_j^T y$. To compare, the estimate from OLS would be $\boldsymbol{X}\hat{\beta}^{\text{ridge}} = \sum_{j=1}^{p} u_j u_j^T y$.

subject of the next subsection.

4.1.2 LASSO regression (ℓ_1 penalty)

The formulation that follows directly from deploying $q = 1$ from Equation 4.1 is the LASSO regression of Tibshirani [1996]. The estimates are the ones that solve

$$\hat{\beta}^{\text{LASSO}} = \arg\min_{\beta} \left\{ ((y - 1_N\beta_0) - \boldsymbol{X}\beta)^T ((y - 1_N\beta_0) - \boldsymbol{X}\beta) + \lambda ||\beta||_1 \right\}. \tag{4.4}$$

The latter constraint now makes the problem non-differentiable, and so we cannot find a closed form solution. But there are several means of estimating β efficiently, such as Least Angle Regression (LAR) of Efron et al. [2004], that actually computes the entire LASSO path (for the different values of the tuning parameter λ) in an extremely efficient manner [Friedman et al., 2009].

Just as for ridge regression, we need to notice the exclusion of β_0 in the penalty term and the scaling problem. Instead of picking out variables to keep in the model as in subset selection, LASSO translates each coefficient by a constant factor λ and then truncating at zero. This is called "soft thresholding" in contrast to the "hard thresholding" of best-subset selection.

When Tibshirani [1996] introduced the LASSO, he motivated the development by comparing the sparsity of the model with ridge regression and found that the LASSO gives some more interpretable models that enjoy some of the same favourable properties. However, the favourability might extend beyond that. Friedman et al. [2009] consider two scenarios; either we are in a sparse scenario, such that in the true model only a small number of coefficients are nonzero; or in a dense scenario, such as if the coefficients are draws from a Gaussian distribution. In the first scenario, ℓ_1 penalty would give the best model fit, and in the second scenario ℓ_2 penalty would give the best model fit. However, in the sparse scenario, only ℓ_1 penalty yields a good model, but in the adverse scenario ℓ_1 does not do too bad - it is more robust towards different types of data. The use of ℓ_1 penalty, therefore, follows what has come to be known as the "bet on sparsity" principle for high-dimensional problems: use a procedure that does well in sparse problems since no procedure does well in dense problems.

4.1.3 Elastic net regression

Zou and Hastie [2005] introduce the elastic net, which is a compromise between ridge regression and LASSO regression. They claim that elastic net often outperforms the LASSO although enjoying a similar sparsity of representation. The second nicety of this formulation is the computational

tractability; it has considerable computational advantages over the ℓ_q penalties [Friedman et al., 2009]. The formulation of the elastic net is as such

$$\hat{\beta}^{\text{elastic net}} = \arg\min_{\beta} \left\{ ((y - 1_N\beta_0) - \boldsymbol{X}\beta)^T ((y - 1_N\beta_0) - \boldsymbol{X}\beta) + \lambda \left(\alpha||\beta||_2^2 + (1-\alpha)||\beta||_1 \right) \right\}. \tag{4.5}$$

Thus the penalty term is a convex combination of both the LASSO and the ridge penalties. They find that the problem can be turned into an equivalent LASSO problem on augmented data. They, then, owe the computational tractability to the LAR algorithm previously discussed, as they can leverage its ability to estimate the LASSO regression efficiently. Furthermore, they find that solving the problem as stated will, although still having features of both LASSO and ridge regression, make the βs appear to incur double shrinkage. However, they find that this solution will do well in situations that are similar to either ridge or LASSO, but not both. The elastic net also has a Bayesian interpretation. The prior here is rather unusual, but it is simply the compromise between Gaussian and Laplacian priors of ridge and LASSO regression,

$$\mathbb{P}(\beta|\lambda; q, \alpha) \propto e^{(-\lambda\left[\alpha||\beta||_2^2) + (1-\alpha)||\beta||_1\right])}.$$

According to Zou and Hastie [2005], they solve the double shrinkage problem by rescaling the solution to β by $1+\lambda_2$, which comes from a reformulation of Equation 4.5 and the defined relation $\alpha = \lambda_2/(\lambda_1 + \lambda_2)$. λ_2 was the ridge penalty coefficient while λ_1 was the LASSO coefficient in the reformulation. Zou and Hastie [2005] see their contribution as a generalisation of the LASSO, as they still want to have the feature of producing a sparse solution. They explain that bridge regression with $q > 1$ always keeps all variables in the regression, and as does ridge. So why is the elastic net superior to the LASSO? Especially for estimation purposes, cases where coefficient estimates are more important than prediction, the elastic net has better qualities when it comes to grouped variables than LASSO. Grouped variables are situations where some variables are highly correlated, which is increasingly important in *large p, small N* problems. In the extreme case, elastic net succeeds in yielding equal coefficient estimates for highly correlated data, while LASSO does not - LASSO does not even have a unique solution [Zou and Hastie, 2005]. This property makes the elastic net less relevant for prediction, although still being similar to both LASSO and ridge.

4.2 Bagging

The name bagging refers to bootstrap aggregation, and the technique itself is, therefore, as the name suggests, a method to aggregate models based on bootstraps to improve the prediction itself. The method is suggested by Leo Breiman in Breiman [1996] and works by averaging the prediction over a collection of bootstrap samples in order to reduce model uncertainty. The estimate is found by fitting the model to each bootstrap sample $\boldsymbol{Z}^{*b}$, $b = 1, 2, ..., B$, where $\boldsymbol{Z}$ contains the training data consisting of both input and outcome variables from the training set $\mathcal{T}$ to have B different fitted models $\hat{f}^{*b}(\boldsymbol{X})$, $b = 1, 2, ..., B$. We then average over the models to have the bagging estimate

$$\hat{f}^{\text{bag}}(\boldsymbol{X}) = \frac{1}{N}\sum_{i=1}^{N}\hat{f}^{*b}(x_i) = \frac{1}{N}\frac{1}{B}\sum_{i=1}^{N}\sum_{b=1}^{B}\hat{f}^{*b}(x_i). \tag{4.6}$$

Following Breiman [1996], we can also show the procedure should work if we assume that each sample $\boldsymbol{Z}^{*b}$ is drawn from the "true" joint probability distribution $\mathcal{P}$. Now, with that assumption and using some simpler notation, when $B \to \infty$ then we have

$$\hat{f}^{\text{bag}}(x) = \mathbb{E}\left[\hat{f}(X)|\mathcal{T}\right], \tag{4.7}$$

and we have the expected squared error of a fitted model of the training set as

$$\mathbb{E}\left[Y - \hat{f}(X)|\mathcal{T}\right]^2 = y^2 - 2y\mathbb{E}\left[\hat{f}(X)|\mathcal{T}\right] + \mathbb{E}\left[\hat{f}(X)^2|\mathcal{T}\right].$$

Using Equation (4.7) and Jensen's inequality we have that

$$\begin{aligned}\mathbb{E}\left[Y - \hat{f}(X)|\mathcal{T}\right]^2 &\geq y^2 - 2y\mathbb{E}\left[\hat{f}(X)|\mathcal{T}\right] + \mathbb{E}\left[\hat{f}(X)|\mathcal{T}\right]^2 \\ \mathbb{E}\left[Y - \hat{f}(X)|\mathcal{T}\right]^2 &\geq \left[y - \hat{f}^{\text{bag}}(x)\right]^2. \end{aligned} \tag{4.8}$$

Integrating both sides of equation 4.8 over $\mathcal{P}$, we get that the mean-squared error of $\hat{f}^{\text{bag}}(\boldsymbol{X})$ is lower than the mean-squared error averaged over $\mathcal{T}$ of $\hat{f}(\boldsymbol{X})$. How much lower depends on how unequal the Jensen's inequality turns out to be. Here the effect of model uncertainty is clear. If $\hat{f}(\boldsymbol{X})$ does not change too much with replicate $\mathcal{T}$, the two sides will be nearly equal, and aggregation will not help.

The trade-off from using a bagging estimate compared to an overall estimate is that $\boldsymbol{Z}^{*b}$ is not drawn from the underlying probability distribution, $\mathcal{P}$, from which $\mathcal{T}$ is drawn, but rather

$\mathcal{P}_{\mathcal{T}}$[16], a distribution that concentrates mass $1/N$ at each set of observed input-outcome variable relations $(y_i, x_i) \in \mathcal{T}$. So if the procedure is unstable, then we can improve prediction through bagging, but if the procedure is stable then $\hat{f}^{\text{bag}}(\boldsymbol{X})$ will not be as accurate as $\hat{f}(\boldsymbol{X})$ as the latter is drawn from $\mathcal{P}$. Furthermore, when we bag a model any simple structure in the model is lost, which is clearly a drawback for interpretation [Friedman et al., 2009].

4.3 Bumping

Bumping is a related bootstrap-based method. However, instead of using bootstrapping for model averaging, it uses it as a stochastic model search mechanism. Tibshirani and Knight [1999], who introduced the method, cite Breiman [1996] for his development of the bagging procedure, but wants to keep the interpretability of the model (where possible[17]), while using the bootstrapping procedure to get different model specifications and to be able to choose from them. This will preserve the structure of the estimator while still inducing stability. This type of stochastic optimisation procedure helps us from ending up in a bad local minimum, and according to Tibshirani and Knight [1999], most adaptive procedures only find local minima. In their convention, the model based on the full training set is included in the search, so it is not possible for the model to find a worse local minimum than the one obtained from the training set.

4.4 Stacking

Stacked generalisation is a method of combining models to give a better predictive accuracy than any single model, just as in bagging. However, instead of building upon the same model type and varying the coefficients in said model, we can stack them according to estimated optimal weights adjusted for the complexity within the individual models. Furthermore, it does not necessarily build upon bootstrapping of the training set, but instead it uses cross-validation. In fact, Wolpert [1992] believes that it can be seen as a more sophisticated version of cross-validation since it combines individual models using cross-validation rather than pick among them.

It can also be seen to be an extension of other model combinations techniques, which assumes the training set is the population data set and combine models to give minimal in-sample error. The extension is on the part of the cross-validatory procedure of obtaining minimal generalisation error.

The stacking estimate of the weights in a linear model using LOOCV is

[16] $\mathcal{P}_{\mathcal{T}}$ is called a bootstrap approximation to $\mathcal{P}$.

[17] Tibshirani and Knight [1999] name neural network models as an exception as their structure does not give easily interpretable results in either case.

$$\hat{w}^{\text{stack}} = \underset{w}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \left[y_i - \sum_{m=1}^{M} w_m \hat{f}_m^{-i}(x_i) \right]^2,$$

where $\hat{f}_m^{-i}(x_i)$ is the mth fitted model on the training set with the ith observation removed. The final prediction is $\sum_{m=1}^{M} \hat{w}_m^{\text{stack}} \hat{f}_m(\boldsymbol{X})$. Therefore, by using the cross-validated predictions $\hat{f}_m^{-i}(\boldsymbol{X})$, stacking avoids giving unfairly high weight to models with higher complexity [Friedman et al., 2009].

4.5 Bayesian methods

We have already seen how Bayesian statistics can be used regularise models under certain prior distributions of the parameter estimates. Likewise, we could also show that bootstrapping is another way to sample estimates from the posterior distribution under uninformative priors. We even motivated the methodology section by a Bayesian example and used BIC as a model selection criterion. In this section, some space is devoted to introducing Bayesian statistics as a method of sampling from the posterior predictive distribution as well as for model combinations.

4.5.1 Bayesian in prediction

At first, let us specify a sampling model $\mathbb{P}(\boldsymbol{Z}|\theta)$ for our data $\boldsymbol{Z}$ given our parameters θ and a prior distribution for the parameters $\mathbb{P}(\theta)$ reflecting our knowledge about θ before we see the data. We can then compute the posterior distribution

$$\mathbb{P}(\theta|\boldsymbol{Z}) = \frac{\mathbb{P}(\boldsymbol{Z}|\theta)\mathbb{P}(\theta)}{\int \mathbb{P}(\boldsymbol{Z}|\theta)\mathbb{P}(\theta)d\theta}. \tag{4.9}$$

For most Bayesian predictions, it is this knowledgeable approach to our prior distribution that provides the real benefit of the use of Bayesian methods[18], but some of the supervised learning techniques are too complex, and so we cannot possibly claim with a straight face that our priors are selected because of the need to capture our prior belief about the problem [Neal, 2012]. However, just as in the cases of regularisation we saw earlier, different priors have been applied in the complex models with good results. For example, MacKay [1992] gives the weights and biases of a neural networks Gaussian prior distributions and lets the variance of the Gaussian prior be a hyperparameter[19], which allows the model to adapt whatever degree of smoothness indicated

[18] Besides the fact that maximum likelihood methods find the mode of the parameters and uses these for prediction, which does not necessarily coincide with mean of the posterior distribution of a new observation as in Bayesian prediction.

[19] Which in the machine learning terminology means a parameter whose value is set before the learning process begins, but in the Bayesian terminology means a parameter of the prior distribution.

by the data.

Now, the posterior distribution of the parameters provides the basis for predicting the value of a future observation z^{new} via the predictive distribution

$$\mathbb{P}(z^{\text{new}}|\boldsymbol{Z}) = \int \mathbb{P}(z^{\text{new}}|\theta)\mathbb{P}(\theta|\boldsymbol{Z})d\theta, \tag{4.10}$$

where we recognise $\mathbb{P}(z^{\text{new}}|\theta)$ as being the so-called *likelihood* that we also know as the objective to maximise in frequentist maximum likelihood procedures. The usual maximum likelihood estimate of the data density would then be $\mathbb{P}(z^{\text{new}}|\hat{\theta}^{\text{ML}})$, and thus does not account for the uncertainty in estimating θ.

This method is theoretically convincing, but for a long period of time the integral in Equation (4.9) was intractable for many types of models, as noted by e.g. Albert and Chib [1993] and Zellner and Rossi [1984]. The explosion in computer power available has now made these sorts of integrals relatively easy to compute via computationally intensive simulation methods [Jackman, 2009].

Usually, we restate the posterior distribution in Equation (4.9) as

$$\mathbb{P}(\theta|\boldsymbol{Z}) \propto \mathbb{P}(\boldsymbol{Z}|\theta)\mathbb{P}(\theta) \tag{4.11}$$

where the constant of proportionality is

$$\left[\int \mathbb{P}(\boldsymbol{Z}|\theta)\mathbb{P}(\theta)d\theta\right]^{-1}.$$

The constant of proportionality has the function of ensuring that the posterior density is a proper probability distribution, i.e. that it integrates to one. As Equation (4.11) is analytically intractable, we need to make use of Monte Carlo simulations to sample from the distribution. In this case, there are heaps of different methods now developed in the Bayesian toolbox, which we can make use of. However, some of these, such as the inverse CDF sampling method, are mostly relevant context of simpler modelling. In the next section, some Markov Chain Monte Carlo (MCMC) methods for sampling from the posterior distribution will be introduced. Both Neal [2012] and Chipman et al. [2010] use a sub-element of MCMC methods known as Gibbs sampling in their papers being an introduction to Bayesian Neural Networks and the development of Bayesian Additive Regression Trees (BART) respectively.

4.5.2 MCMC methods

Sampling from the posterior density using MCMC combines the Monte Carlo principle with ideas from Markov Chain theory; these ideas allow us to have dependency between our consecutive proposals of the posterior parameter vector, but still have an iterative history converging to a unique stationary distribution. The reason that dependency is so paramount to its success is that we can stay longer in the areas of the proposal distribution that contributes to the distribution of interest, and thus we can limit the computational time before converging to the unique stationary distribution. This is not generally true, as we for easier problems can make better guesses for the proposal distribution in first instance and exploit the independence between proposals to produce the unique stationary distribution more efficiently[20].

First, brief remarks on some conditions of a Markov chain will be made, and these will allow us to use this as the posterior density. After that, both the Metropolis-Hastings and the Gibbs algorithm will be suggested as algorithms that fulfil these conditions.

First of all, we need the Markov Chain to be ergodic, which will allow us to use the appropriately constructed chain as a distribution according to the relative frequency of which the chain visits sites of the parameter space [Jackman, 2009]. To have ergodicity of the Markov Chain, we need it to be irreducible, positive recurrent and aperiodic.

For the Markov Chain to be irreducible, we need it to be able to go everywhere in the parameter space that it ought to go to from any state that it might be in now with positive probability. This leads us to the definition

Definition 3. (Irreducible Markov Chain)

For some measure ϕ, a Markov chain $\{\theta^{(t)}\}$ on a state space Θ with transition kernel $K(\theta, \mathcal{A})$ is said to be ϕ-irreducible if $\forall\ \mathcal{A} \in \mathcal{B}(\Theta)$ with $\phi(\mathcal{A}) > 0$, $\exists\ n$ such that $K^n(\theta, \mathcal{A}) > 0\ \forall\ \theta \in \Theta$. If this condition holds with $n = 1\ \forall\ \mathcal{A} \in \mathcal{B}(\Theta)$ with $\phi(\mathcal{A}) > 0$ then the Markov chain is said to be strongly irreducible,

where the transition kernel is defined as the conditional probability that at step t, the Markov chain will 'jump' from $\theta^{(t-1)}$ to the set $\mathcal{A}$ [Jackman, 2009]. Irreducibility is sufficient to ensure the existence of a stationary[21] distribution for a Markov Chain, but the additional assumption of positive recurrence ensures the uniqueness of the stationary distribution. A Markov Chain is positive recurrent if all states are positive recurrent, and a state i is recurrent if the chain will

[20] Meaning that the rate at which the Monte Carlo error of the series converges to zero is not as fast as the rate we would get with an independence sampler. Jackman [2009] show how dependency increases the variance of the chain, which then increases the required length of the series before we are comfortable that we have explored the distribution.

[21] Where stationarity, in this sense, means that the distribution will persist once it is established.

return to state i with probability 1 within finite time. This ensures that the chain has the same limiting properties for *every* starting value.

Next, the notion of periodicity is most easily seen in discrete for Markov Chain on discrete state spaces, where a chain is periodic if it can end up in areas of the parameter space, wherefrom it cannot return to other areas of the parameter space. The period thus addresses the need for other communicating states before being able to revisit the state it is in. Also, the period of any irreducible chain can be no smaller than the period of any of its states. We need the chain the be aperiodic, which is defined as an irreducible Markov Chain with period 1. Hence, we need to be able to revisit our present state in the chain without any interim states.

Now, with irreducibility, positive recurrence and aperiodicity, we have ensured that a stationary distribution exists and that it is unique. However, if we can show that the Markov Chain is reversible then we have also ensured that a stationary distribution exists, and when we have reversibility the other necessary conditions to establish that the chain is ergodic follows easily in most cases related to the algorithms we will look at [Jackman, 2009]. This is a nice sufficient condition as the Metropolis-Hastings algorithm, with the Gibbs sampler as a special case, can be shown to be reversible. A Markov chain is said to be reversible if it possesses the detailed balance, which is defined as

Definition 4. (Detailed Balance Condition)

Consider a Markov chain $\{\theta^{(t)}\}$ *with state space* Θ*, transition kernel* $K(\cdot,\cdot)$ *and stationary distribution* $\mathbb{P}(\theta|\boldsymbol{Z})$*. If*

$$\mathbb{P}(\theta^{(t)})K(\theta^{(t+1)},\theta^{(t)}) = \mathbb{P}(\theta^{(t+1)})K(\theta^{(t)},\theta^{(t+1)})$$

$\forall\ \theta \in \Theta$ *then the Markov Chain is said to possess detailed balance*

So that if we can come up with transition kernels $K(\cdot,\cdot)$ that are reversible, satisfying the symmetry inherent in the detailed balance condition, then the resulting Markov chains will converge to the stationary distribution, $\mathbb{P}(\theta|\boldsymbol{Z})$[22].

The definition of a reversible Markov chain leads us directly to an algorithm that satisfies the condition, namely the Metropolis-Hastings algorithm. The original algorithm from Metropolis et al. [1953] was created using a symmetric proposal distribution, but this was later altered in Hastings [1970] to include non-symmetric proposal distributions. Following Jackman [2009],

[22]In short, as we, for discrete time with continuous time related proof and for all states i and j, have that $\sum_j \mathbb{P}(\theta^{(j)})K(\theta^{(i)},\theta^{(j)}) = \sum_j \mathbb{P}(\theta^{(i)})K(\theta^{(j)},\theta^{(i)}) = \mathbb{P}(\theta^{(i)})\sum_j K(\theta^{(j)},\theta^{(i)}) = \mathbb{P}(\theta^{(i)})$, and so this is a stationary distribution.

we say that the Metropolis-Hastings algorithm defines a set of 'jumping rules' that generate a Markov chain on the support of $\mathbb{P}(\theta|\boldsymbol{Z})$, Θ. At the start of iteration t we have $\theta^{(t-1)}$, and we make the transition to $\theta^{(t)}$ if we accept the proposal from the 'proposal' or 'jumping' distribution $J_t(\theta^*, \theta^{(t-1)})$, where θ^* is a sample from the distribution. The algorithm then defines an acceptance probability[23]

$$\alpha = \min\left(\frac{\mathbb{P}(\theta^*|\boldsymbol{Z})J_t(\theta^*|\theta^{(t-1)})}{\mathbb{P}(\theta^{(t-1)}|\boldsymbol{Z})J_t(\theta^{(t-1)}|\theta^*)}, 1\right),$$

where we notice that the posterior distribution of interest is both in the numerator and the denominator, so that the constant of proportionality cancels out. To explore the posterior probability distribution nicely, notice that $J_t(\theta^*, \theta^{(t-1)})$ must resemble $\mathbb{P}(\theta^*|\boldsymbol{Z})$ somewhat, as this is more computationally efficient since we will have higher acceptance ratios and not have to create seemingly unnecessary proposals in this case. In fact, if we choose $J_t(\theta^*|\theta^{(t-1)})$ as the target distribution then $\alpha = 1$. However, if we have high acceptance ratios due to having chosen $J_t(\theta^*|\theta^{(t-1)})$ to wander around the neighbourhood of the previously accepted value $\theta^{(t-1)}$, then we again have a computationally inefficient exploration of $\mathbb{P}(\theta|\boldsymbol{Z})$ as discussed earlier regarding the benefits of independence between proposals. The argument supporting that the acceptance ratio in the Metropolis-Hastings algorithm generates a symmetric function that satisfies the reversibility condition starts with the transition kernel. The transition kernel for the Metropolis-Hastings algorithm is given by the acceptance probability times the jumping distribution

$$K(\theta^{(t)}, \theta^{(t+1)}) = J_t(\theta^{(t)}|\theta^{(t+1)})\alpha,$$

and so

$$\begin{aligned}
\mathbb{P}(\theta^{(t)}|\boldsymbol{Z})K(\theta^{(t)}, \theta^{(t+1)}) &= \mathbb{P}(\theta^{(t)}|\boldsymbol{Z})J_t(\theta^{(t)}|\theta^{(t+1)})\alpha \\
&= \mathbb{P}(\theta^{(t)}|\boldsymbol{Z})J_t(\theta^{(t)}|\theta^{(t+1)}) \cdot \min\left(\frac{\mathbb{P}(\theta^{(t+1)}|\boldsymbol{Z})J_t(\theta^{(t+1)}|\theta^{(t)})}{\mathbb{P}(\theta^{(t)}|\boldsymbol{Z})J_t(\theta^{(t)}|\theta^{(t+1)})}, 1\right) \\
&= \min\left(\mathbb{P}(\theta^{(t+1)}|\boldsymbol{Z})J_t(\theta^{(t+1)}|\theta^{(t)}), \mathbb{P}(\theta^{(t)}|\boldsymbol{Z})J_t(\theta^{(t)}|\theta^{(t+1)})\right),
\end{aligned}$$

which is a symmetric function in $\theta^{(t)}$ and $\theta^{(t+1)}$. This symmetry grants us reversibility of the resulting Markov chain as given in Definition 4.

There are some traditional general choices for the proposal distribution, and these are presen-

[23] In the algorithm we compute α and sample $U \sim \text{Unif}(0, 1)$ and accept the proposal θ^* if $\alpha > U$.

ted here; the symmetric proposal as in the original Metropolis article $J_t(\theta^*|\theta^{(t-1)}) = J_t(\theta^{(t-1)}|\theta^*)$ leading to the simple acceptance probability $\alpha = \min\left(\frac{\mathbb{P}(\theta^*|\boldsymbol{Z})}{\mathbb{P}(\theta^{(t-1)}|\boldsymbol{Z})}, 1\right)$; the independent proposal $J_t(\theta^*|\theta^{(t-1)}) = J_t(\theta^*)$ likewise leading to the simple acceptance probability; the random walk $J_t(\theta^*|\theta^{(t-1)}) = J_t(\theta^* - \theta^{(t-1)})$; and the Gibbs sampler.

4.5.3 The Gibbs sampler

The Gibbs sampler, introduced in Geman and Geman [1984], is often used when θ is high-dimensional and sampling from the posterior density $\mathbb{P}(\theta|\boldsymbol{Z})$ is too hard for any other sampling method. The Gibbs sampler breaks sampling from $\mathbb{P}(\theta|\boldsymbol{Z})$ down to a series of inter-related, easier, lower-dimensional sampling problems. So, the parameter vector θ is divided into D sub-vectors proposedly according to correlation and distributional similarity. Then, the Gibbs sampler iterates through every jth element, with $j = 1, ..., D$, of θ and in each step evaluates the jth component of θ conditional on the rest of the elements in θ. The key is that by sampling from this series of D, lower-dimensional conditional densities, we can generate a Markov Chain on Θ that has the joint posterior distribution of θ as its unique stationary distribution [Jackman, 2009]. Thus, we iteratively sample from

$$\theta_j^t \sim \mathbb{P}(\theta_j|\theta_{-j}^{t-1}; \boldsymbol{Z}),\ \theta_{-j}^{t-1} = (\theta_1^t, ..., \theta_{j-1}^t, \theta_{j+1}^{t-1}, ..., \theta_D^{t-1}), \tag{4.12}$$

so that we have included updates of the parameter vectors in the conditions, and thereby simplify the problem. In the sense of the Metropolis-Hastings algorithm we have that each component of the proposal vector is updated sequentially and implicitly then the jumping distribution are simply the conditional densities $\mathbb{P}(\theta_j|\theta_{-j}^{t-1}; \boldsymbol{Z})$, and so the acceptance probability would always be 1, $\alpha = 1$ [Jackman, 2009]. Furthermore, some of the conditional probabilities might not be possible to sample from directly, and then we could choose to use the Metropolis-Hastings algorithm in conjunction with the Gibbs sampler.

4.5.4 Bayesian model averaging

Like the Bayesian information criterion for model selection, we could also use posterior model probabilities for averaging models. Recall that the BIC uses the relative posterior model probabilities to choose between models, Bayesian model averaging simply uses the results of the models relative to their posterior probability [Hoeting et al., 1999]. Instead of having parameter uncertainty directly in Equation (4.10), we will instead sum over a model space with K different models $\mathcal{M}_K$, where each model $M_k \in \mathcal{M}_K$, $k = 1, ..., K$ may be of different type and complexity

$$\mathbb{P}(z^{\text{new}}|\boldsymbol{Z}) = \sum_{k=1}^{K} \mathbb{P}(z^{\text{new}}|M_k) \cdot \mathbb{P}(M_k|\boldsymbol{Z}), \tag{4.13}$$

where the posterior probability for model M_k is given by

$$\mathbb{P}(M_k|\boldsymbol{Z}) = \frac{\mathbb{P}(\boldsymbol{Z}|M_k)\mathbb{P}(M_k)}{\sum_{l=1}^{K} \mathbb{P}(\boldsymbol{Z}|M_l)\mathbb{P}(M_l)}.$$

Hence, we are not relieved from the task of choosing a prior for the model probabilities, although we can choose to remain agnostic about that issue and choose uninformative priors. Another choice is with regards to both keeping the summation in Equation (4.13) practically feasible and to have a parsimonious predictive model. One possibility, which underlies Occam's window method, excludes models that fare far worse than the model that predicts data the best according to some threshold [Hoeting et al., 1999]. Another possibility, appealing to sparsity principle and Occam's razor, is to exclude complex models which receive less support from the data than their simpler counterparts. [Hoeting et al., 1999] also provide other of such model exclusion rules for the data analyst to use if necessary, which can be found in their paper.

Now to one of the most potent ideas within learning - boosting. Variants of its traditional uses are considered some of the best "off-the-shelf" procedures for data mining [Friedman et al., 2009]. Sometimes, though, it fails to have high predictive ability when the relationships between the input variables are too complex, as the input models, which we will see in the next section, are required to be rather simple.

4.6 Boosting

The first boosting algorithms, including AdaBoost, was created by Schapire and Freund, who have published a book about machine learning predictive classification based on the boosting procedure and algorithm; Schapire and Freund [2012]. However, as the book primarily focuses on classification issues, we will not be using the book extensively, but only to see the idea of boosting.

Roughly, the idea of boosting is to take a *weak learning* algorithm - an algorithm that gives a classifier that is slightly better than random - and transform it into a *strong* classifier, which does much better than random. Boosting procedures does this by taking a collection of weak classifiers, and then reweighting their contributions to form a classifier with much better accuracy than any individual classifier. So, in general, boosting is an approach for improving generalisation of a learning method based on the application of a single method to many appropriately modified versions of the training data. Many types of methods have been used in combination with

boosting, these include stumps (regression trees with only one split), small trees (regression trees with several splits), small linear models or simple nearest-neighbour methods [Cherkassky and Mulier, 2007]. The most frequently applied method of AdaBoost is trees, where the simplest form is the boosting of stumps, such that it builds an ensemble by splitting the training data at one point in one input variable and then training every new instance iteratively by emphasising the training data mismodelled in the previous instance. The use of smaller trees instead of stumps allows each model to model multiplicative relations between small sets of variables while still being a fairly weak identifier.

As previously discussed, AdaBoost is created with the objective of making a set of weak learners into a strong learner, but after the fact it is shown to be equivalent to a forward stagewise[24] additive model with exponential loss function and the basis function expansions as individual classifiers. The general forward stagewise additive model algorithm is described in the next section.

4.6.1 Forward stagewise additive modelling

In general, the (additive) basis function expansions have the form

$$f(\boldsymbol{X}) = \sum_{m=1}^{M} \beta_m b(\boldsymbol{X}; \gamma_m),$$

where β_m, $m = 1, 2, ..., M$ are the expansion coefficients, and $b(\boldsymbol{X}; \gamma_m)$ are usually simple functions of the multivariate argument $\boldsymbol{X}$, characterized by a set of parameters γ. For trees γ would parameterise the split variables and split points at the internal nodes, and the predictions at the terminal nodes [Friedman et al., 2009]. The models, then, are fitted by minimising a loss function averaged over the training data

$$\min_{\{\beta_m, \gamma_m\}_{m=1}^{M}} L\left(y, \sum_{m=1}^{M} \beta_m b(\boldsymbol{X}; \gamma_m)\right). \tag{4.14}$$

This typically requires computationally intensive numerical optimisation techniques. But as an alternative, forward stagewise additive modelling approximates the solution to Equation (4.14) by sequentially adding new basis functions to the expansion without correcting the previously set coefficients. Such that we do not take into account that a dis-optimal step in this iteration might lead to better solution in another iteration an eventually a bitter fit to the training data. Because of that, it can be seen as a greedy algorithm. Following Friedman et al. [2009], the algorithm is

[24] Note that the stagewise strategy does not adjust previously entered terms when new ones are added, and that this distinguishes it from stepwise approaches [Friedman, 2001].

given by

Algorithm 1. *Forward Stagewise Additive Modelling*

1. Initialise $f_0(\boldsymbol{X}) = 0$.
2. For $m = 1, 2, ...M$:
   (a) Compute
   $$(\beta_m, \gamma_m) = \arg\min_{\beta,\gamma} L(y, f_{m-1}(\boldsymbol{X}) + \beta b(\boldsymbol{X}; \gamma)). \tag{4.15}$$
   (b) Set $f_m(\boldsymbol{X}) = f_{m-1}(\boldsymbol{X}) + \beta_m b(\boldsymbol{X}; \gamma_m))$.

The optimisation in Equation (4.15) is typically a grid search, where we at each step find the solution that maximally reduces the current residual. This is a very computationally inefficient method of finding the optimum since we do not take into account information pointing to the direction in which the optimum might be but merely checks all coefficient values in a preset range. A popular method of optimising Equation (4.15) when the loss function is differentiable is by gradient descent, which eventually leads to the gradient boosting method.

4.6.2 Gradient boosting

The general negative gradient is given by

$$-g_m(\boldsymbol{X}) = -\left[\frac{\partial L(y, f(\boldsymbol{X})}{\partial f(\boldsymbol{X})}\right]_{f(\boldsymbol{X})=f_{m-1}(\boldsymbol{X})},$$

and the steepest descent method then chooses the update standing instead of $\beta b(\boldsymbol{X}; \gamma)$ in Equation (4.15) as $\boldsymbol{h}_m = -\rho_m g_m(\boldsymbol{X})$ where ρ_m is a scalar and the solution to

$$\rho_m = \arg\min_{\rho} L(f_{m-1}(\boldsymbol{X}) - \rho g_m(\boldsymbol{X})),$$

which is then found by line search. So instead of grid searching two parameters, we now only have to line search one parameter [Friedman, 2001]. The reader eloquent in econometrics might ask why optimisation schemes free from line search is not used instead. Here Dauphin et al. [2014] argue that in high-dimensional problems the number of saddle points grows exponentially in the number of parameters and that methods such as Newton-Raphson or Gauss-Newton[25] are prone

[25] Which, though, solves the problem of intractable second-order differentiation where this is a problem. Although exploiting this information is more efficient where possible.

to get stuck in these saddle points. The final update $f_m(\boldsymbol{X}) = f_{m-1}(\boldsymbol{X}) + \beta_m b(\boldsymbol{X}; \gamma_m)$ is then also replaced with $f_m(\boldsymbol{X}) = f_{m-1}(\boldsymbol{X}) - \rho_m g_m(\boldsymbol{X})$.

Each boosting iteration usually reduces the training sample error, which also means that each iteration fits the model classes to the training data [Friedman et al., 2009]. As found in Subsection 3.2 this will increase model variance and make predictions worse. However, there are other ways to control overfitting than merely choosing the number of boosting iterations. First, one can scale the contribution of each iteration with $0 < \nu < 1$ in $f_m(\boldsymbol{X}) = f_{m-1}(\boldsymbol{X}) - \nu\rho_m g_m(\boldsymbol{X})$, such that a lower value of ν result in a larger training error, $\overline{\text{err}}$. However, lower values ($\nu < 0.1$) are also found to yield good generalisation abilities, although requiring a larger amount of boosting iterations and, so, are more computationally demanding.

One could also include stochastic gradient boosting, which creates a new sub-sample consisting of a part, η, of the training data (without replacement) to use in each iteration, which increases robustness against overfitting [Friedman, 2002]. Friedman et al. [2009] find that shrinking performs better than stochastic gradient boosting as the number of iterations increase, although both perform better than no regularisation, and that a combination of the two methods performs good as well.

This completes the section on model uncertainty and prediction variance and how to deal with it. Next, we will develop different specific supervised learning algorithms and models, that we will be using for our predictions.

5 Supervised Learning Algorithms

In addition to these already voluminous methodology sections, we will set up two very popular machine learning techniques, namely regression tree models and neural networks. These techniques will be carried out in practice in Section 7 together with the already seen methods of OLS, ridge, LASSO and elastic net seen in Section 4. These will follow a brief section on the K-nearest neighbours regression technique used to create a variable in Section 6.

5.1 K-nearest neighbours regression

K-nearest neighbours (KNN) regression is a distance-based non-parametric regression method, where only a subset of the observations is used for each prediction. If we first define $N_k(\boldsymbol{X})$ to be the K-neighbourhood of $\boldsymbol{X}$, e.g. the set with the lowest distance according to a chosen distance measure, which is Euclidean for the sake of our use. Then the algorithm simply predicts based on the average of the K nearest neighbours,

$$\hat{f}(\boldsymbol{X}) = \frac{1}{K} \sum_{i \in N_k(\boldsymbol{X})} y_i.$$

5.2 Tree-based methods

Tree-based methods have been very successful since they, in the most uncomplicated cases, are very intuitive and easy to interpret. Moreover, when ensembles are built from them, they are easy to manage, are good at finding interacting input variables and show great prediction ability. However, the case of tree-based methods is not only for use as an off-the-shelf predictor; it has also won many, in fact most, Kaggle[26] competitions. Chen and Guestrin [2016] find that among the 29 challenge winning solutions published on Kaggle's blog during 2015, 17 solutions used XGboost, a variant of the tree-based models described partly later.

The first individual model described is CART (an acronym for Classification And Regression Trees), a very popular method for tree-based regression by Leo Breiman explained in Breiman [2017].

5.2.1 Classification and regression trees

In the simplest case, we restrict ourselves to recursive binary partitions. We split the space by selecting a variable and making a binary split within it, which separates the model outcome into

[26]Kaggle calls itself "your home for data science" and has many open data sets to train algorithms, and yourself, on. They also host competitions, which allow firms to set up prizes for solving their algorithmic problems.

two regions. We then model the outcome by the mean of y in each region. To make the most of the model framework, we choose the variable and split-point so to achieve the best fit - although this definition can vary, e.g. for greedy algorithms the best fit is determined by what explains the most variance within any given instance [Friedman et al., 2009].

However, we could also make a larger tree than this one stump, and we would do this by choosing a new split variable and a new split point in each region, and this process can be continued practically until the algorithms perfectly identify each observation within the sample, but usually some stopping rule is applied.

Then, intuitively, tree-based methods stratification of the feature space can be represented by a flowchart, where the algorithm sequentially answers binary questions in the input data that maximises the variance explained in the outcome variable. Following Friedman et al. [2009], we say that our data consist of p input variables and an outcome variable for each of the N observations: that is, (y_i, x_i) for $i = 1, 2, ..., N$, with $x_i = (x_{i1}, x_{i2}, ..., x_{ip})$. Then we need our algorithm to select splitting variables, select split points and what shape the tree should have automatically. Suppose we have stratified the data into M regions $R_1, R_2, ..., R_M$, then we model the response as a constant c_m in each region

$$f(\boldsymbol{X}) = \sum_{m=1}^{M} c_m I(\boldsymbol{X} \in R_m).$$

Where we, just as in the K-nearest neighbour case, see that c_m is just the average of the y_i variables within the region when we have squared error loss

$$\hat{c}_m = \frac{1}{N_m(\boldsymbol{X})} \sum_{x_i \in R_m} y_i,$$

where $N_m(\boldsymbol{X})$ denotes the number of observations in the region R_m. Generally, now, finding the best partition given a tree depth to minimise the loss is computationally infeasible, as we would have to take into account an insurmountable number of combinations of splitting variables and split points, hence we proceed with a greedy algorithm. Starting with all the data, we consider splitting variable j and split point s and define the pair of half-planes

$$R_1(j, s) = \{\boldsymbol{X} | x_j \leq s\} \text{ and } R_2(j, s) = \{\boldsymbol{X} | x_j > s\}.$$

Then we seek the splitting variable j and split point s that solve

$$\min_{j,s} \left[\sum_{x_i \in R_1(j,s)} (y_i - \hat{c}_1)^2 + \sum_{x_i \in R_2(j,s)} (y_i - \hat{c}_2)^2 \right].$$

Now, for each splitting variable, the split point s can be chosen very quickly and hence scanning through all of the inputs determination of the best pair (j, s) is feasible.

As Chen and Guestrin [2016] note, this is a reasonably sound strategy when the number of splits is comparatively low, but when constructing huge ensembles of models as those in the next subsections, we should allow ourselves to make approximations to these, what they call, exact greedy algorithms. Hence, they use an extension of the general idea of reducing the number of possible split points in each variable; for example, by only considering a subset of the possible split points such as the quantiles or percentiles; or by acknowledging one-hot encoded data and realising that these effectively only have one possible split point.

As noted in Friedman et al. [2009], there are a lot of possibilities for controlling the complexity of the trees, and without going too much into the details, they find that these are relevant to consider: a priori specifying tree depth, only split tree nodes if the decrease in loss due to split exceeds some threshold, and pruning already grown trees by deleting some nodes according to already seen regularisation methods.

5.2.2 Bagged regression trees and random forests

Bagging as a technique of creating an ensemble of models has already been introduced in Subsection 4.2. Moreover, as noted in [Friedman et al., 2009] bagging seems to work especially well for high-variance, low-bias procedures, such as trees described above. Furthermore, the first application of bagging in the original article, Breiman [1996], is using trees. However, since the strength of bagged trees is to reduce variance by using an ensemble of trees, it is a slight disadvantage that the trees are generated by the same procedure and with bootstrapping samples that are fairly similar. This means that the trees are similar, and therefore there are less to win by averaging out the noise.

The idea in random forests, then, is to de-correlate the individual trees by growing them differently, which, then, gives better chances of reducing variance. In the paper Breiman [2001] builds upon his previous work on bagged trees and constructs a method of de-correlating them and naming the method, and the paper, random forests, continuing the botanic analogy. The random forest model achieves this effect through a random selection of the input variables in each split. As noted earlier, the additional extension to the standard regression trees also include new complexity control parameters, and for random forests, we can now choose the number of

input variables to select from for each split. This is, of course, to be chosen by error estimates, but we can also be guided by general guidelines such as: the more correlation between the input variables, the higher the amount of sub-sampling of input variables.

5.2.3 Gradient boosting machines and XGboost

This section introduces the gradient boosting machine in the case of trees from Friedman [2001] and builds upon the general framework described in the boosting section under Subsection 4.6.2. Then we note that the 'eXtreme Gradient boosting' (XGboost) model of Chen and Guestrin [2016] is an efficient extension of the gradient boosting machine, which also allows for more tuning parameters, such as sub-sampling of the input variables as in the random forest algorithm and early stopping. The gradient tree boosting algorithm, or gradient boosting machine, is given underneath, where the effect of a scaling of the contribution of each iteration and sub-sampling each target data have been included, as in Friedman [2002], as well as sub-sampling the input variables.

Algorithm 2. *Gradient Boosting Machine with additions*

1. Initialise $f_0(\boldsymbol{X}) = \underset{\gamma}{\operatorname{argmin}}\, L(y, \gamma)$.
2. For $m = 1, 2, ...M$:
   (a) Compute
   $$r_m = -\left[\frac{\partial L(y, f(\boldsymbol{X}))}{\partial f(\boldsymbol{X})}\right]_{f=f_{m-1}}$$
   (b) Fit a regression tree of depth 2^{J_m} to a random subset (without replacement), η, with only a random sample of input variables, b, of the targets r_m giving terminal regions R_{jm}, $j = 1, 2, ...J_m$.
   (c) For $j = 1, 2, ..., J_m$ compute
   $$c_{jm} = \arg\min_{c} \sum_{\boldsymbol{X} \in R_{jm}} L(y, f_{m-1}(\boldsymbol{X}) + c)$$
   (d) Set $f_m(\boldsymbol{X}) = f_{m-1}(\boldsymbol{X}) + \nu \sum_{j=1}^{J_m} c_{jm} I(\boldsymbol{X} \in R_{jm})$.
3. Output $\hat{f}(\boldsymbol{X}) = f_M(\boldsymbol{X})$.

Here $I(\cdot)$ is the indicator function indicating the membership of $\boldsymbol{X}$ within a given set R_{jm}. As can be seen from Algorithm 2, there are a number of different ways to regularise the model. In

addition to this, the implementation, as discussed earlier, by Chen and Guestrin [2016] introduce even more ways of regularising the model, and thus this is not an exhaustive list of parameters, although it arguably includes the most essential regularisation parameters. For example, this model mixes the attributes of the standard gradient boosting machine and the random forest, and so it builds de-correlated trees using boosting to create the ensemble. The coefficient b is the fraction of input variables sampled in each tree. If this parameter is 1, then the trees will tend to be more correlated than otherwise, and the benefits gained from creating an ensemble of them will be less. Likewise, we can control the tree depth, the shrinkage parameter ν, and the fraction of the data to subset in each tree, η, which will also help us control the complexity and prevent overfitting.

5.2.4 Bayesian additive regression trees

Another important extension to the CART related methods is the Bayesian Additive Regression Trees (BART) model introduced in Chipman et al. [2010]. As a Bayesian model, they make use of the techniques of regularisation and prediction by sampling from the posterior as seen in Subsection 4.5. In BART, each tree is constrained by a regularisation prior to being a weak learner, hence being somewhat shallow, and fitting and inference are accomplished via an iterative Bayesian backfitting MCMC algorithm[27] that generates samples from a posterior. A further nicety of this Bayesian type of modelling is that it enables full posterior inference including point and interval estimates of the unknown regression function as well as the marginal effects of potential predictors as found in Chipman et al. [2010]. BART distinguishes itself from a previous version of Bayesian tree models of the same authors introduced in Chipman et al. [1998] by not focusing on Bayesian model averaging as in Subsection 4.5.4, but rather making a sum of trees but with each tree regularised to keep their effect small, and so make them weak learners. The BART model can be expressed as

$$Y = \left(\sum_{j=1}^{J} g(\boldsymbol{X}; T_j, M_j) \right) + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \sigma^2),$$

where each binary regression tree is T_j, M_j is the set of parameters $\hat{c}_{mj}$ in each terminal region and $g(\cdot)$ is the function which assigns all $\hat{c}_{mj} \in M_j$ to $\boldsymbol{X}$. To complete the model specification, Chipman et al. [2010] impose a prior on all parameters, namely $(T_1, M_1), ..., (T_J, M_J)$ and σ. They do this by reducing the prior formulation problem to the specification of a few interpretable

[27] Bayesian backfitting is the Gibbs sampling procedure applied to additive models according to the inventors in Hastie et al. [2000].

hyperparameters (in the Bayesian sense), which govern priors on T_j, M_j and σ. An extension of the model, with corresponding `R` implementation, is proposed by Kapelner and Bleich [2013], and they use almost the same hyperparameter and prior specification as described in Chipman et al. [2010]. It is this extension, named `bartMachine`, that is implemented in this thesis.

The prior on T_j affects the location of the nodes within the tree. The depth of the tree is defined as the distance from the root, and thus the root itself has depth 0. Nodes at depth d are non-terminal with prior probability $\alpha(1+d)^{-\beta}$ where $\alpha \in (0,1)$ and $\beta \in [0,\infty]$. This component of the prior on T_j has the ability to enforce shallow tree structures, and thereby limiting the flexibility within a tree. In Chipman et al. [2010] they implement and recommend building the model with $\alpha = 0.95$ and $\beta = 2$ for models with more than one tree, which yields prior probabilities of the node at depth d being non-terminal with $d = 1, 2, 3, 4, \geq 5$ to 0.05, 0.55, 0.28, 0.09 and 0.03, respectively. In the application in this thesis, this is considered a bit too shallow, and so $\beta = 1.5$ is considered instead since the value of β will penalise deeper trees less. The prior for c_{mj} is allowed to be dependent on T_j, and the recommended prior is the *conjugate*[28] normal distribution $\mathbb{P}(c_{mj}|T_j) = \mathcal{N}(\mu_c, \sigma_c^2)$. To steer the choice of μ_c and σ_c^2, Chipman et al. [2010] note that $\mathbb{E}(Y|\boldsymbol{X})$ is the sum of the J c_{mj}'s under the sum-of-trees model, and since the c_{mj}'s are apriori iid, the induced prior on $\mathbb{E}(Y|\boldsymbol{X})$ is $\mathcal{N}(m\mu_c, m\sigma_c^2)$. Kapelner and Bleich [2013] pick $m\mu_c$to be the range centre $(y_{\min} + y_{\max})/2$. As this can be affected by outliers, they recommend using the log-transform, if the problem is concerning. The variance hyperparameter σ_c^2, then, is empirically chosen so that the range centre plus or minus k[29] variances cover 95% of the outcome values in the training set. Thus, the value of σ_c^2 is chosen such that $m\mu_c - k\sqrt{m}\sigma_c = y_{\min}$ and $m\mu_c + k\sqrt{m}\sigma_c = y_{\max}$. Hence, the larger the value of k the smaller the value of σ_c^2, which results in more model regularisation. For $\mathbb{P}(\sigma)$, the prior on error variance, the choice also falls upon a conjugate prior. The choice for Chipman et al. [2010] and Kapelner and Bleich [2013] is the same, although differing in semantics. Thus, following Kapelner and Bleich [2013] lead us to choosing $\mathbb{P}(\sigma) = \Gamma^{-1}(\nu/2, \nu\lambda/2)$, where λ is determined from the data so that there is a $q \in (0,1)$ apriori chance that the BART model will improve upon the RMSE from an OLS with $q = 0.9$ by default [Kapelner and Bleich, 2013]. This prior limits the probability mass placed on small values of σ^2 to prevent overfitting. Furthermore, the higher the value of q, the larger the values of the sampled σ^2's, resulting in more model regularisation [Kapelner and Bleich, 2013].

[28] Conjugacy in priors is very important to Bayesian modelling since it allows for much simpler posterior distribution. We will not, though, go more in-depth with the matter in this thesis.

[29] With $k = 2$ in this implementation.

5.3 Neural networks

The much-hyped neural network model, typically represented topologically by nodes connected in different layers, draws its references not to botany, as in tree-based models, but instead to the human brain. According to Friedman et al. [2009], the idea of neural networks was to extract linear combinations of inputs and then model the target as a non-linear function of these features.

In the model, derived features z_m are created from linear combinations of the inputs, and then the target y is modelled as a function of linear combinations the z_m,

$$\begin{aligned} z_m &= \sigma(\alpha_{0m} + \alpha_m^T \boldsymbol{X}), \quad m = 1, ..., M, \\ f(\boldsymbol{X}) &= g(\beta_0 + \beta^T \boldsymbol{Z}), \end{aligned}$$

where $\boldsymbol{Z} = (z_1, z_2, ..., z_M)$. Also, in the feature equation α_{0m} is termed *bias* and the vector α_m is termed *weights.*

Using a single-layer network as in this paper, the activation function $\sigma(\cdot)$ is typically chosen to be a sigmoid in the form of a logistic function or a hyperbolic tangent (tanh), but other popular choices include Gaussian radial basis function and the rectified linear unit (reLU),

$$\sigma(\cdot) = \begin{cases} 1/(1 + e^{-(\alpha_{0m} + \alpha_m^T \boldsymbol{X})}) & \text{Logistic function} \\ \frac{e^{(\alpha_{0m} + \alpha_m^T \boldsymbol{X})} - e^{-(\alpha_{0m} + \alpha_m^T \boldsymbol{X})}}{e^{(\alpha_{0m} + \alpha_m^T \boldsymbol{X})} + e^{-(\alpha_{0m} + \alpha_m^T \boldsymbol{X})}} & \text{Hyperbolic tangent} \\ e^{\frac{-||\boldsymbol{X} - c_m||_2^2}{2\sigma^2}} & \text{Gaussian radial basis function} \\ \max(\alpha_{0m} + \alpha_m^T \boldsymbol{X}), 0) & \text{Rectified linear unit.} \end{cases}$$

The output function, $g(\cdot)$, will define what kind of outcome we want, and since we have a linear output, we simply want it to be the identification function, i.e. $g(\beta_0 + \beta^T \boldsymbol{Z}) = \beta_0 + \beta^T \boldsymbol{Z}$. The derived features z_m are called *hidden units* because the values z_m are not directly observed, and the vector of them, $\boldsymbol{Z}$, is called a *hidden layer.* In general, there can be more than one hidden layer which then constitutes a deep neural network - previously a model that has been difficult to fit, but which has become subject to intense attention thanks to the work of in particular Geoffrey Hinton and the idea of greedy layerwise pre-training [Sutskever, 2013]. Note that if the activation function is linear, and we, therefore, have zero hidden layers, the network becomes a standard linear model in the case of a linear output layer. The network is then trained by minimising a loss, which by a squared loss function becomes

$$\begin{aligned} R(\theta) &= \sum_{i=1}^{N} R_i(\theta) \\ &= \sum_{i=1}^{N} (y_i - f(x_i))^2 \\ &= \sum_{i=1}^{N} (y_i - g(\beta_0 + \beta^T z_i))^2. \end{aligned}$$

The generic approach to minimisation is by gradient descent, as the case was for the gradient boosting models, which is this case is called *back-propagation* [Friedman et al., 2009], a name it has probably been given due to the layered fashion of optimisation using the familiar chain-rule of differentiation. Typically, we do not want the global minimiser of $R(\theta)$ since this is likely to be an overfit solution. Hence, some regularisation is needed which can be implemented in a number of ways, say through early stopping or directly through a penalty term, such as the weight decay[30].

The derivatives of the model as given in [Friedman et al., 2009] are

$$\begin{aligned} \frac{\partial R_i(\theta)}{\partial \beta_m} &= -2(y_i - f(x_i))\sigma(\alpha_{0m} + \alpha_m^T x_i) \\ \frac{\partial R_i(\theta)}{\partial \alpha_m} &= -2(y_i - f(x_i))\beta\sigma'(\alpha_{0m} + \alpha_m^T x_i)x_i. \end{aligned}$$

Given these derivatives, a gradient descent update at the $(t+1)$st iteration has the form

$$\begin{aligned} \beta_m^{(t+1)} &= \beta_m^{(t)} - \gamma_r \sum_{i=1}^{N} \frac{\partial R_i(\theta)}{\partial \beta_m^{(r)}}, \\ \alpha_m^{(t+1)} &= \alpha_m^{(t)} - \gamma_r \sum_{i=1}^{N} \frac{\partial R_i(\theta)}{\partial \alpha_m^{(r)}}, \end{aligned}$$

where the inserted parameter γ_r is the *learning* rate, a parameter that controls the size of the steps and therefore works similarly to ν in Algorithm 2. Now we rewrite the derivatives as

[30] Implementing the loss as $R(\theta) + \lambda J(\theta)$ with regularisation function as in section 4.1

$$\frac{\partial R_i(\theta)}{\partial \beta_m} = \delta_i \sigma(\alpha_{0m} + \alpha_m^T x_i)$$
$$\frac{\partial R_i(\theta)}{\partial \alpha_m} = s_{mi} x_i$$

where δ_i and s_{mi} are "errors" from the current model iteration at the output and hidden layer units respectively. From their definitions, they satisfy

$$s_{mi} = \sigma'(\alpha_{0m} + \alpha_m^T x_i)\beta_m \delta_i, \quad m = 1, ..., M, \tag{5.1}$$

which is known as the back-propagation equations. In the *forward pass*, the weights are fixed, and the predicted values are computed. In the *backward pass*, the errors δ_i are computed, and then back-propagated via Equation (5.1) to give the errors s_{mi}. Both sets of errors are then used to compute the gradients for the updates in the next forward pass. Back-propagation can, though, be very slow to compute, but as the Hessian matrix can be large and difficult to work with, other approximate methods will have to suffice. One problem with optimisation is that the error function $R(\theta)$ is non-convex, possessing many local minima. As a result, the final solution is quite dependent on the initial weights. Friedman et al. [2009] suggest one must at least try a number of random starting configurations (within the range $[-0.7, 0.7]$ with scaled data), and then choosing the solution with the lowest (penalised) error. This procedure sort of mimics bumping of Subsection 4.3, but averaging procedures are not recommended due to the non-linearity of the model. Instead, bagging could be a feasible solution within one pre-specified set of initial weights [Friedman et al., 2009].

5.3.1 Bayesian neural networks

Neal [2012] gives an extensive walk-through of Bayesian methods applied to the neural network model. Since it is complicated to incorporate prior knowledge, Neal [2012] looks at several possible classes of prior distributions for network parameters that reach sensible limits as the size of the networks grow, such as Gaussian diffusion priors, which has been used to win a challenge called NIPS 2003 [Friedman et al., 2009]. Even though regular MCMC methods could probably be successfully implemented, Neal [2012] also demonstrate the hybrid Monte Carlo.

Neal [2012] finds that the use of hyperparameters (in the Bayesian sense) controlling the priors for weights is roughly analogous to the role of a weight decay constant in conventional neural networks. However, instead, with Bayesian training, values for these hyperparameters (or, more precisely, their distribution) can be found without the need for a tuning process.

To be concrete, Neal [2012] finds that one option to the hyperparameter associated with weights and biases is a Gaussian distribution with zero mean and standard deviation σ, yielding

$$\mathbb{P}(\alpha_{0m}, \alpha_{1m}, ..., \alpha_{pm}) = \frac{1}{(2\pi\sigma^2)^{-k/2}} e^{-\sum_{j=0}^{p} \frac{\alpha_{jm}^2}{2\sigma^2}}.$$

The prior for the hyperparameter itself is expressed in terms of the "precision", $\tau = \sigma^{-2}$, which is given a prior distribution of the Gamma form with mean ω:

$$\mathbb{P}(\tau) = \frac{(u/2\omega)^{u/2}}{\Gamma(u/2)} \tau^{u/2-1} e^{-\tau u/2\omega},$$

where the value of u is positive and controls how broad the prior for τ is. These are the same priors as those in the software implemented in this thesis, and they are a collaboration between Neal and David McKay, the author inspiring the software, in Neal [2012]. The hierarchical prior structure implemented they name Automatic Relevance Determination (ARD) prior, which creates restrictions to overfitting.

Furthermore, in the implementation of the software used in this thesis, the `brnn` package of `R`, they also use a Gauss-Newton approximation to the Hessian for faster optimisation from Foresee and Hagan [1997], and initial weights given by Nguyen and Widrow algorithm [Rodriguez and Gianola, 2016].

6 Data

The following section will describe the data applied in the predictive analysis. It will comprise of a description of the data and its preprocessing, a description of the input variables in their groups; house characteristics, local amenities, and macro and municipal level variables and a discussion of their relevance. And, lastly, a description of the outcome variable; transaction prices of Danish single-family houses from 2005 to 2016. Observations in the year 2016 are used as a hold-out sample so that the results can be compared to those of SKAT in that year. Thus, this chapter will enable the reader to visualise the data and reproduce the results.

6.1 Data description

The raw data stems from several sources which have been carefully chosen and combined to give the best possible idea of what your house would be worth. The data is not entirely exhaustive and still has a lot of room for improvement - which will become apparent throughout this section. The main part of the data is gathered in cooperation with the land surveying company LIFA A/S through "Den Offentlige Informationsserver" (OIS), which includes the register databases "Bygnings- og Boligregistret" (BBR), "Statens Salgs- og Vurderingsregister" (SVUR) and "Det Fælleskommunale Ejendomsstamregister" (ESR). OIS uses a unique identifier for each municipality, which then has a unique identifier for each house in the municipality so that these can be combined to a unique identifier across all databases. These registers contain several individual databases each covering unique information.

First, BBR has a data set containing specific characteristics on more than 2.5 million buildings in Denmark - both private and public. This means that it holds information on each and all single-family houses and buildings belonging to them such as carports and greenhouses, but also university buildings, upper secondary school buildings, and daycare facilities. This is important since a part of the information the data set holds is the geographical location in the form of longitude and latitude coordinates in the global datum system World Geodetic System (WGS84), and so can be used to provide important local amenities information for each household. In addition to this, the data set provides information on building characteristics such as year of construction, year of reconstruction, total building area, roofing, exterior walling, and heating.

Second, ESR[31] provides information on the taxation of the house among other information such as ownership and cadastral information.

Third, SVUR contains information on historical sale values, dates, types of sale and SKAT's

[31] Which according to *www.kombit.dk/ESRudfasning* is getting phased out as part of the government's digitalisation strategy.

appraisals, which, though, are updated to 2016 values, which is what makes the year 2016 relevant as a hold-out sample.

Furthermore, in order to exploit the locational data to the fullest, public information is gathered on *www.kortforsyningen.dk*, a part of Denmark's open data strategy, where information on the coastline, forests, lakes, train stations and much more are available in shapefile and other Geographic Information System (GIS) formats.

In order to account for municipality level variables, publicly accessible data from Statistics Denmark are gathered, which contains all kinds of information on Danish citizens on an aggregated level[32]. There are several other small sources of information. Some are utilised in the data, such as energy labels, and some are discarded due to limited data quality, such as district plans for, for example, the maximum height of a building.

The structure of the data is characterized as pooled cross-sectional data, implying both cross-sectional and time-series features. However, this thesis will not exploit the information inherent in a panel data set, that each house with repeated sales reveals idiosyncratic information about that house, which may not be revealed in the data set otherwise. This is because one of the primary data sets containing building information is updated whenever changes to the house information occur. When this data is linked with the sales data, then some of the house information is spurious, and we have to delete that sale. But some houses do not have changes between sales, and this information can potentially be exploited for further research.

6.2 Data preprocessing

The raw data needed extensive preparation and cleaning, and the reader will be spared the details, but the broad lines of the operations will be given beneath to give an understanding of the data.

6.2.1 House characteristics

First of all, all buildings in BBR are characterized by a unique observation, and so in the respect of our needs, it is in long format. As the objective is to appraise single-family houses, all buildings are deleted from the data set if they do not belong to or is a single-family house. Furthermore, buildings belonging to a given single-family house are then augmented to it. As many of the variables are categorical on a nominal scale, these are encoded to the building as dummy variables before augmentation. This has made an enormous complexity in the data set since information on a possible garage is included in wide format, so that we know if the house includes e.g. an

[32]It would have been better if these variables were on ZIP-code level rather than municipality level, but this data is not publicly available.

external garage, and then what building materials are used for the exterior walling and roofing of that garage. Furthermore, information on heating and type of water access is also provided in this manner - so-called one-hot encoding. Energy labels are given as categorical values with a large number of categories since the scale on which they are given has changed during different time-periods. This data is, though, on an ordinal scale, and this information is exploited by encoding it as such using the conversion table given in Pedersen [2016], which can be seen in Appendix B.1 in Figure B.1 only with numerical values[33]. Since the data on energy labels is fairly incomplete, the data has been imputed by a standard linear OLS model using an intercept and log house age and log years since last renovation as independent variables motivated by the correlations found in Næss-Schmidt et al. [2015].

6.2.2 Locational amenities

Firstly, in the BBR buildings data, there are labels referring to which type of building type it is, or which type of institution it belongs to. This information is used to create categories of different types of buildings such as daycare homes, so that the great circle distance to *nearest* of each type of building can be calculated. Secondly, coordinates[34] on the 50 largest cities in Denmark are retrieved from *www.latitudelongitude.org/dk* and used to calculate distances to the 11 largest cities in Denmark, and then also to the nearest of the largest 50 cities in Denmark. Thirdly, public information on *www.kortforsyningen.dk* is gathered. This includes point layers of windmills, city centres and train stations, polygon shape layers of forests and lakes and line layers of the coast among more. This data is handled using `QGIS 2.18.18` and eventually saved in a format usable for `R`. For windmills[35], city centres and train stations, the information can be directly used to calculate distances to the nearest of these points. But for forests and lakes only bigger ones are chosen to be included, so that the distances to the nearest centroids of the largest 1830 forests and 469 lakes are calculated. For the coastline, the plugin `QChainage` is used to select points each 30 metres along the coast, before these are saved and the distances from each household to the nearest of these points are calculated. Lastly, neighbourhood square metre prices are included as information for each house. Three different methods of including this information is chosen. The first one is to select houses sold before the house itself, within the same year and within two kilometres of the house and weight their square metre prices by the

[33] Linearity in the effects of getting better energy labelling is not expected, and so for generalised linear model types, one-hot encoding on these labels is used.

[34] I suppose the coordinates are some centroid of the polygon shaping the city.

[35] Although it says so in the GeoDanmark data description, the windmills category does not only include windmills. In a deep-dive on the data, it was noticed that some of the windmills included in the data are actually pylons. But as both these structures are disliked due to their obstruction of the view, this information is kept anyway.

inverse of the geographic distance as in McCluskey et al. [2013]. The second method is similar to the first one, but only using the standard arithmetic average instead of using weights. The third method is to employ K-nearest neighbour averaging as described in Subsection 5.1 to each municipality and for houses sold within a given year, where K was chosen as maximally 10, but could also be lower if the number of sales within that municipality within that year was lower than 10.

Furthermore, relevant information is found on *www.statistikbanken.dk* and appended to the data set on a municipality and yearly level. Among these variables are the Consumer Price Index (CPI) of Denmark, which is used to deflate all monetary termed values, so that they are presented in real terms.

Distances are calculated using the Haversine formula for determining great circle distances between two points on a sphere given their longitudes and latitudes, which is fairly simple and excludes ellipsoidal effects. It is published in e.g. Sinnott [1984], and the distance can be rewritten as

$$d = 2r\sin^{-1}\left(\sqrt{\sin^2\left(\frac{\phi_2-\phi_1}{2}\right)+\cos(\phi_1)\cos\left(\phi_2\right)\sin^2\left(\frac{\psi_2-\psi_1}{2}\right)}\right)$$

where d is the distance between two points with longitude and latitude (ψ, ϕ) and r is the radius of the Earth[36].

6.2.3 Data filtering

As the aim is to uncover the prices of single-family houses, which is the largest real-estate stock in Denmark with about 1.2 million of the total 2.2 million houses [Rigsrevisionen, 2013], the first step in the filtering process is to remove other types of properties, such as condominiums, terraced houses, etc. The second step is to exclude irregular transactions, and this is possible since the data in SVUR includes a categorical variable that defines the type of sale. This variable includes "regular sale", "foreclosure sale", "family handover" and "other forms of sale". Both family handover, foreclosure sales and other forms of sales will not reflect the market price of the house as other parameters influence the price. Furthermore, buildings with asbestos are deleted, due to the limited knowledge of the severity.

Also, as an AVM is intended for cost-effective appraisals of average properties, it is appropriate to use mechanical and automated criteria to remove unusual observations [Schulz et al., 2014]. Two methods are chosen to this end, with the first one being to exclude houses sold for less than 100,000.- kr. and more than 25,000,000.- kr. as advised by FD [2014], who suggest they do it

[36]The radius used in this thesis is 6378.137 kilometres.

to counter typing errors and exclude atypical houses from their statistics. Then, observations on houses where the ratio of the square metre price of the house to that of the neighbourhood is in the extreme percentiles are deleted. Chosen percentiles are shown in the table below.

1%	2%	3%	4%	5%	25%	50%	75%	95%	96%	97%	98%	99%
0.225	0.298	0.345	0.383	0.415	0.731	0.963	1.273	2.055	2.185	2.364	2.666	3.344

Table 6.1: Percentiles of the ratio of house price per square metre

In some instances, there are more single-family houses per unique identifier, which, according to LIFA A/S, can be buildings on a rented plot - these are also deleted.

Independently from the reasoning behind the first filtering, observations in the test year 2016 are deleted if SKAT's valuations are lower than the arbitrary threshold of 1000.- kr. This amounts to deleting 468 observations.

6.3 Description of the input variables

In this section, necessary descriptive statistics and explanations on the input variables will be provided, so that the reader understands this part of the data. Once again, this will be split into explaining house characteristics and locational amenities.

6.3.1 House characteristics

First, we consider the average, the median, the 5th percentile and the 95th percentile values for a range of inputs connected to housing characteristics. In parenthesis are the percentages of how many houses have the specification considered, and the statistics of variables related to these specifications are conditioned on having the specification. Furthermore, it should be noted that the energy labels given in the table below are already imputed as described earlier, and so 36.16% of the energy labels are imputed. This is, of course, a problem for the model. Also, there are `R` packages that support modelling with missing variables, such as `XGboost`, or methods within certain modelling that support smarter imputation, such as the expectation-maximisation algorithm within Bayesian modelling[37]. However, since the `caret` package, that does not support incomplete data, mainly have been used, imputation of missing energy labels is chosen. Therefore, the summary statistic of the imputed energy labels is what is presented below.

[37] Also other model techniques, although more inherent with Bayesian techniques.

Name of input variable	Mean	Median	5$^{\text{th}}$ percentile	95$^{\text{th}}$ percentile
Number of storages	1.02	1	1	1
Building footprint area	127.4	123	68	205
House floor area	129	124	68	209
Taxes	11950	8359	3145	34871
Age	64.24	54	14	136
Years since renovation† (17%)	27.3	26	11	126
Energy label	6.561	7	5	9
Porch area (11.84%)	17.89	16	4	36
Cellar and/or attic area (39.36%)	53.36	50	16	98
Non-dwelling area (14.96%)	30.18	25	10	62
Business area (0.98%)	32.7	24	8	90
Attached garage area (3.46%)	37.82	35	16	69
Attached carport area (3.62%)	33.89	33	18	53
Sunroom/winter garden area (10.75%)	20.43	20	10	33
Attached annex type building (4.58%)	22.1	14	6	69.25
External garage area (21.75%)	33.11	29	15	65
External carport area (29.19%)	29.19	27	14	51
Annex area (18.14%)	33.88	18	6	122
Greenhouse area (0.13%)	10.81	10	5	22
N	179952			

Table 6.2: Descriptive statistics of house characteristics

Note: The summary statistics of area type is only from the subset of houses, that has that type of building.
†If the building is not renovated, it is set to be the same age as the building itself. The statistics presented here, though, is only for those that have been renovated.

As seen from the summary statistics above, there are a number of different characteristics of a house, which could be important for determining its price.

Most obviously the general condition of a house affects its price. Therefore, all the information on factors related to the overall conditions of a house is exploited; materials, general renovations as well as the building years and energy labels (found to have strong relationships in Næss-Schmidt et al. [2015]). These factors may also (and probably) have a separate effect on the price of a house, but as a machine learning application with the goal of prediction, we are not so interested in identification issues and isolating causal effects. Næss-Schmidt et al. [2015] seek to understand energy labels, and how they affect house prices in Denmark. They had access to the number of errors and omissions in their houses (*tilstandsrapporter*) indicating the average condition of houses, and they find that the number of errors and omissions follow the energy labels quite

closely. As noted earlier, the table should be read such that e.g. 18.14% of single-family houses have an annex, and, of those houses, the median area of the annex is 18 square metres.

Furthermore, the data includes information on the materials used in building the houses, and these are all included as dummy variables.

Exterior wall	**Roofing**	**Heating**
Brick (90.470%)	Fibre cement (45.198%)	District heating (45.528%)
Lightweight concrete (4.267%)	Tile (25.669%)	Central heating: 2 units (44.019%)
Wood (2.281%)	Cement (17.310%)	Central heating (42.983%)
Half-timbering (1.515%)	Roofing felt (3.501%)	Electric heater (5.412%)
Concrete (0.420%)	Built-up (2.505%)	Heat pump (4.104%)
Missing (0.014%)	Metal (2.024%)	Heater (0.864%)
Other (1.033%)	Thatched (1.554%)	None (0.041%)
	Missing (1.350%)	Other (∼0%)
	PVC (0.082%)	
	Glass (∼0%)	
	Other (0.800%)	

Supplemental heating	**Water**
Fireplace (2.155%)	Public tap water (97.970%)
Other (0.731%)	No or well water (0.007%)
Solar panels (0.612%)	Other (2.037%)
Heat pump (0.547%)	
Heater (0.428%)	

Table 6.3: Statistics of house characteristics

Note: If needed, the "Other" category will be used as base-category.
$N = 179952$.

Here we see that most Danish single-family houses sold between 2005 and 2016 have brick exterior walls, fibre cement or tile roofing. Some of the houses have more than one type of heating installed, and 45.5% have district heating. In addition to the primary heating, some houses also include other types of heating, e.g. 2.16% of the houses include a fireplace. In BBR, there are several types of water installations included, which is summed up into three categories. It should be noted that "Public tap water" not only includes public tap water but also smaller water boards only connected to a small set of houses. In the "Other" category, there are also houses with their own water connection and even smaller water boards.

With all these kinds of information, it is easily imagined that one could create types of houses from these characteristics that are especially attractive. We could use these variables to know

whether or not we have a lovely old, but renovated, half-timbered house located centrally in a big town (or on the countryside). We could also try to detect patrician villas as bigger houses in bricks and with tile roofing in certain, more affluent, areas built from the 1860s to the 1930s. These relationships are very complex, and so the data seen here would be best exploited by a model more adept at complex relationships.

In Appendix B.2 Table B.1 the characteristics of the external buildings, such as the garage, similar to Table 6.3 are also given. By utilising this data, we can hopefully get a better prediction of how much a given annex to a given house is worth or what type it is. Using the area of that external building and what type of material is used to build it could be good predictors of whether or not it is insulated, and then for garages that they include space not only for cars. This could be important, since, by Danish law, if a wanted addition of an external building exceeds 50 square metres[38], the builder must seek building permission if not otherwise specified within the local plan. This says something about the magnitude of, for example, a garage greater than 50 square metres. In this data, 10.6% of garages are greater than this amount, so some of these are probably built for additional purposes.

When investigating the minimum and maximum values of the variables in Table 6.2, it is found that further cleansing of outliers and typos can be considered, especially in the training set in the purpose of not fitting too tightly to these extremes, and therefore be more able to predict the average house. One method with which this can be done is using the Mahalanobis distance as considered in Schulz et al. [2014], which will find the most extreme properties on selected input variables weighted by a covariance matrix.

6.3.2 Locational amenities

A common phrase in real estate circles is ”location, location, location” referring to the extreme importance of a property's position, which also gave name to TV shows in Denmark and other countries for the same reasons. Several of the preceding articles reviewed in Section 2 emphasise the effect of locational amenities on the value of houses. Motivated partly by these papers and partly by what seems possible with open data, the data set is constructed. For example, as in e.g. Ottensmann et al. [2008], municipality level variables, such as the median family wealth are included. Moreover, school spending within the municipality are included since Hayes and Taylor [1996] have shown the significant effect of school quality[39]. Furthermore, Ottensmann et al. [2008] look into several ways in which cities affect the prices of houses, and other articles aiming for

[38]Found in *www.bolius.dk/love-og-regler-for-udhuse-og-skure-17566* the 30th of April 2018

[39]They have test scores and so they use this as a proxy for school quality, whereas municipality school spending is used in this thesis.

prediction, such as McCluskey et al. [2014, 2013], also include some specifications on distances to cities, and so several variables related to the distance to different cities are included. In the summary statistic in the Appendix B.3 Table B.2, we can see the rest of the input variables and some information on how they are distributed. For the interested reader, the chosen GeoDanmark items' geographical distribution can also be found in the Appendix B.4.

As problems with spatial autocorrelation, the phenomenon that houses in the same neighbourhoods have similar square metre prices and are mostly of same style, year and quality, have been investigated in articles such as Basu and Thibodeau [1998], information inherent in the neighbourhood square metre prices is exploited by introducing three different specifications for that. The figure beneath plots the average unweighted square metre prices for each municipality on a map of Denmark in 2016, which gives a great visualisation of the diversity square metre prices across Denmark.

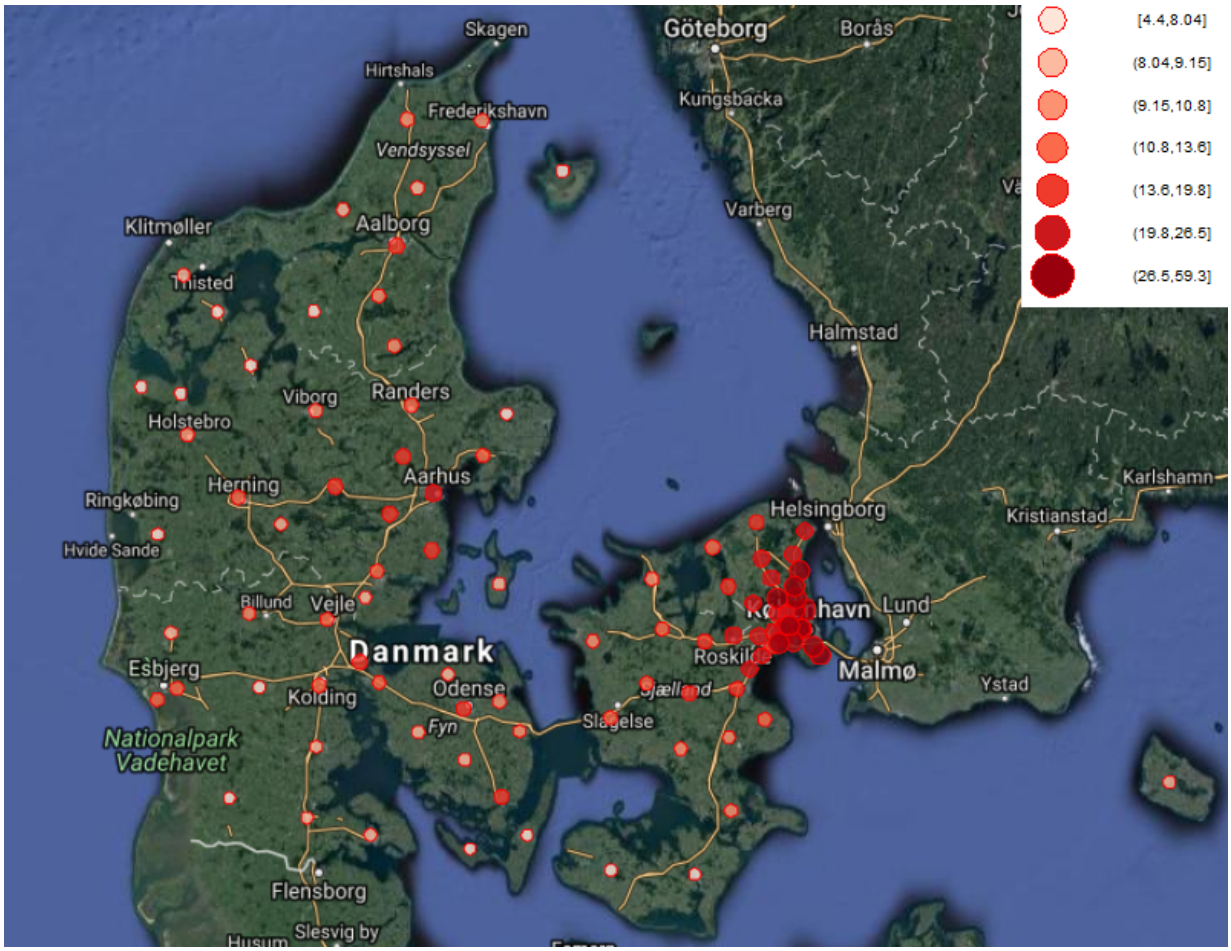

Figure 6.1: Average square metre price in 2016

Note: The labels in the top right corner are presented in thousands of Danish kroner in real terms.

Now, from this figure, we can also see importance from unobserved factors such as distance to the freeway, which can also partly be captured by these general variables. We can also begin to grasp the necessity of treating spatial autocorrelation directly in the prediction, especially for linear regressions. Imagine, for example, that we had included a linear effect of the distance to greater cities, then houses sold on Bornholm would easily be under-priced, as people who bid for houses on Bornholm probably, heterogeneously, have a lesser preference towards bigger cities, and value nature (and sun) more. From this graph, we can also see the geographical dispersion of prices as discussed in Hviid [2017].

6.4 House prices

As explained in Section 2, prices of single-family houses are relevant in many instances, and even in the case of taxation use, it is relevant as argued by Rigsrevisionen [2013]. The total recorded number of sales during the years of 2005 to 2016 is 179952. The summary statistics of these sales are given in the table below

	Mean	Median	5$^{\text{th}}$ percentile	95$^{\text{th}}$ percentile
House Price	1998000	1600000	461347	4737625
Log House Price	14.26	14.29	13.04	15.37

Table 6.4: Descriptive statistics for house prices in Danish kroner

The whole distribution of sales over the years is given in Section B.5 Figure B.8. Interestingly, there is a fair amount of positive skew in the distribution, it has a fat right tail and is zero-truncated (of course). For many types of machine learning algorithms, this does not pose a significant problem. For example, a K-nearest neighbour regression would average sales values over K similar houses, or tree structures would average sales within certain branches of that tree. But take a linear model and there are several problems with these characteristics; the model would linearly extrapolate on the effects it finds so that we could experience negative prices, and outliers would have a disproportionate weight on the final results. So the linear OLS comparison model is made (and the other linear models), log house price is used as the outcome variable, which has a nicer distribution for modelling as we can also see in Section B.5 Figure B.8.

Furthermore, the distribution over the years matches that of the test year, 2016, perfectly well, as seen when comparing with Figure C.10 in Section 7. Moreover, the distribution of the number of sales does not suggest any issues with the representability of the year 2016 or any extraordinary event in this year. The house price index is plotted to show how the number of sales correlates with the index. We can also see on the figure below that the general house price index fluctuates a bit less than the number of sales, probably because people are more reluctant to sell at a loss even though the market is in distress - so they hold on to the house a bit more.

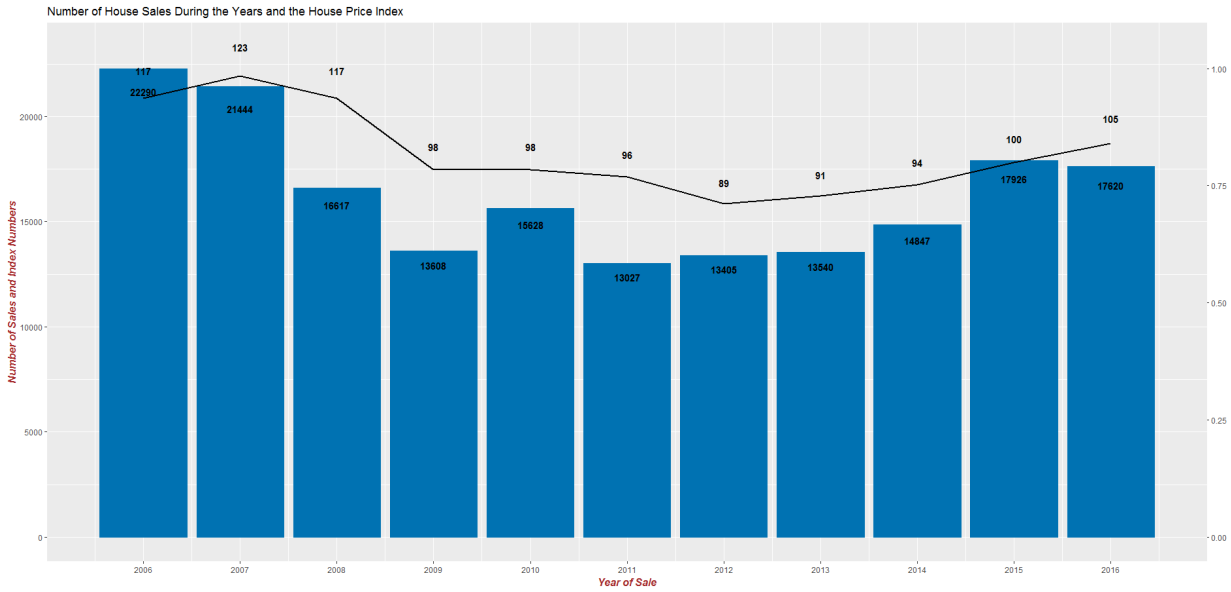

Figure 6.2: Number of house sales and house price index plotted against sale year

Note: The line shows the House Price Index, and the bars show the distribution of the number of sales over the years.

7 Results

This chapter presents the results of the prediction models. The results are evaluated on the basis of Root Mean Squared Error (RMSE), which is what they have been trained to minimise, but they are also evaluated on Mean Absolute Percentage Error (MAPE) and the fraction of predictions within the span of $[-20\%, 20\%]$ around the realised sale price as used in ICE [2016] (subsequently referred to as "the percentage within the span"). The implementation of the models is using the `caret` package of Kuhn et al. [2008] in `R`, which includes a wide range of different models and allows for the tuning of parameters using several error estimates including cross-validation as applied in this thesis. Furthermore, it is able to handle multiple computer cores fitting models at the same time using the `doParallel` package for the Windows OS.

When reading the results section, one should have several delimitations in mind. First, when building up the data for different model types, more effort into translating each variable to something that will make the model better can be given. As an example, consider the distance to Copenhagen which enters in numerical form in the OLS regression, although it certainly does not have a linear effect on the outcome across the values. Preferably, several dummy variables with thresholds on the distance would have been made. Instead, only mechanical computations to tailor to the different model types are used; for linear models, this is a standardization of the non-binary input variables and a log-transformation of the outcome variable; for neural networks the non-binary input variables is demeaned and scaled to be in $[-1, 1]$ while binary input variables are set to be in $\{-1, 1\}$ such that they geometrically go through origo; for tree-based models, no changes to the data from the state that they were in. Second, although the predictive model of ICE is linear with input variable coefficients estimated by OLS, the comparison to the OLS model in this thesis is not total since theirs is directly set up to handle spatial autocorrelation within a linear model framework in a smart manner.

In this section, a description of the tuning strategy for the hyperparameters of the models is given. This is followed by a presentation of the results, and an analysis of the reasons for the comparative differences in the predictive ability of the models in the data.

7.1 Tuning strategy

Because of the large number of models, observations and computational intensity in the tuning process, it becomes necessary to have a metaheuristic strategy for the tuning process. Of course, when the objective is partly to evaluate the models against each other, one should choose one simple and fair strategy. Birattari and Kacprzyk [2009] find that when several algorithms are compared, all of them should make use of the available domain-specific knowledge and equal

computational effort should be invested in all the pilot studies. Similarly, in the testing phase, all the algorithms should be compared on an equal time basis.

Therefore, the tuning strategy is to make a pilot scheme by randomly drawing 5% of the observations, which are used to tune the parameters. This has the separate effect lowering the number of parameters in the tuning process, which is undesirable, as the dummy encoded variables have so little variance in these cases; some of them will have zero 1 observations such as built-up roofing on an annex. In some cases, it has been allowed to change tuning parameters, if there has been a reason to believe that this will improve performance significantly when scaling up the data. As seen later, this is why the XGboost model has a tree depth of 9 instead of the tree depth of 5 suggested by the tuning process as depicted in Figure C.7. Preferably, the parameters are tuned in the full sample, but, due to the limited amount of computer time and power, this is neglected. In the pilot scheme, a somewhat large number of hyperparameter combination searches is allowed in the fast algorithms, but a lower number of combinations is preferred in computationally intensive methods such as BART. Hyperparameter combinations for grid search are chosen based on rules-of-thumb from e.g. Friedman et al. [2009]. The tuning of all models can be seen in Appendix C.

7.2 Model performances

Table 7.1 on page 74 together with Figure 7.1 on page 75 present the main results of this thesis using the metrics RMSE, which is defined as $\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{f}(x_i)\right)^2}$, MAPE, which is defined as $\frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{f}(x_i)}{y_i}\right|$ and the percentage within the span. RMSE is very sensitive to the really expensive houses, and MAPE is more sensitive to the cheaper houses with very volatile prices, so we have to have those things in mind when evaluating the models. The results are compared with SKAT's appraisals, which form some sort of baseline. However, the measure of the performance relative to SKAT is fairly uncertain since SKAT's appraisals on average are 9.5% higher than the realised prices that enter into the analysis. On the other hand, the new appraisals are purely mechanical, and there are not - as opposed to SKAT's appraisals - manual corrections. The difference in the distributions of realised prices and SKAT's prices can be seen in Figure B.8. The four main results are presented below.

First, from the top of Table 7.1 we find OLS, ridge, LASSO and the elastic net. These are modelled on 151 input variables, but the tuning is only on a subset of these variables which

comes down to 129 variables after clearing variables with low variance[40]. In this data set, the models do not find any reason to regularise the OLS fit very much, and so all the variables are deemed somewhat relevant - or at least the exclusion of unimportant variables is not justified by the penalisation of other variables. As mentioned in Subsection 3.2, the mechanical inclusion of all variables squared and all pairwise interaction terms, for those for which it is possible, to allow for non-linear effects is impossible due to the sheer size of this input matrix. These types of models can definitely be improved upon if one is willing to invest time in setting up data specifically for this purpose. One could, for example, include relevant interaction terms manually. For example, if an interaction between square metre prices in the neighbourhood and house floor area is included the RMSE decreases by almost 10000.

Nonetheless, these types of models fare poorly with this data set. They only make a minor improvement to SKAT's model, and they are outperformed by many others, although doing better than CART, which probably fits too closely to the training set data and the Bayesian neural network which has only been trained on 6.67% of the data and has not been tuned. With the data as is, these models cannot model any complex variable relationships but only linear effects from specifications from a baseline, and so there are no positive second-order effects from having a big house if it is near Copenhagen compared to if it is in Western Jutland, which is unrealistic. Furthermore, three different measures of square metre prices that are somewhat similar are included, which makes the coefficient estimates more uncertain and therefore are worse for predictions. That said, though, the input matrix still remains invertible, and so there are differences in the measures.

Second, CART is the worst performing algorithm of all with a discomforting 16% less predictions within the span than SKAT. So that although this model is very adept in modelling complex variable relationships and is easily interpreted, it is not directly applicable. As seen from Figure C.4 the lowest cross-validation errors are from a max tree depth of 33 and up implying a high level of complexity in the data, but also that the model would easily overfit. Interestingly, though, is it that bagging thirty of such CART models produces a model that is much superior to the first one. Bagged CART has almost 4% more predictions within the span than SKAT and almost 20% less RMSE than CART itself. This suggests that there is a high amount of instability in the model development, and hence there is a significant amount of variance reduction due to the dif-

[40]With variance threshold found according to the Bernoulli distribution, where the variance is $p(1-p)$, where p is the probability of success, and where p is determined from the binomial distribution with the objective of having 5 successes in $8117 \cdot 0.8 \approx 6494$ draws in that variable with probability larger than 99.99%. This is done to ensure that the model does not break down due to low variance in a variable within the tuning process. This means that the required p is set to 0.03 and variables with less variance than $0.03 \cdot (1 - 0.03) = 0.0291$ is deleted.

ferent choices of splitting variables and split points the bagged models make when presented with bootstrapped data sets. The story for the random forest model is similar; it achieves substantial prediction variance reduction by building an ensemble of trees that are de-correlated through a sub-sampling of the input variables at each splitting. Although nothing can be said for sure as to why the bagged CART and random forest models have such similar performances, one reason could be that some of the most important variables are the measures of square metre prices, and that there is a high chance for the random forest to include at least one of these measures in each split which lessens the de-correlation, thus making the models more similar to each other. As seen from Figure C.9, they are more correlated to the each other than any other model. The Bayesian version within the tree-based models, BART, also predicts reasonably well. Due to its high computational demand, it even has to do with less data than the other models - it is only trained on a random sample of 50% of the training data.

Third, XGboost is a fast algorithm with many tuning parameters, and so it allowed doing a broad range of tuning fast as seen in Figure C.7. This is partly the reason for the success of this model. Moreover, it also inherits properties of both the random forest model as well as something similar to the mechanism in bagged CART, but also some properties of the very successful strategy of building ensembles via boosting. It is by far the best performing algorithm in terms of predictive ability with a MAPE of approximately 22% compared to approximately 23.5% for SKAT and almost 10% more predictions within the span than SKAT as seen from Figure 7.1.

Fourth, the darling of the previous attempts to push linear models off the statistical house appraisals throne, the neural network performs fairly well - and a lot better than OLS. It requires more care and nurturing with regards to the input variables, and also more than has been applied in this thesis. There is a large variety of these types of models, and so there are a considerable amount of scope for the implementation of more flexible, and regularised, types of neural networks. Furthermore, as seen in Figure C.8, the tuning parameters are not stable, but the computational intensity of the fitting process makes it more difficult to find good tuning values.

Interestingly, though, is it that even though it has considerably lower RMSE than SKAT, it also has a considerably higher MAPE. This suggests that the neural network is better than SKAT at predicting expensive houses, but has problems ”figuring” out how to price cheaper houses. Due partly to this observation and the somewhat low correlation between the XGboost model and the neural network model compared to their accuracy seen in Table 7.1 and Figure C.9, it would be interesting to see if the XGboost model's predictive ability could be enhanced by stacking it smartly with the neural network.

The other neural network model, Bayesian regularisation neural network, fares poorly. In a comparison in Friedman et al. [2009], it is found to be the best performing model in their data set, so it is profoundly disappointing that it fares so badly in this case. However, the combination of Bayesian methods and the neural network method might still be fruitful; it is designed to require only a few choices of model specification by the researcher since the regularisation is guided by only a few hyperparameters, which can be guided by data within the fitting process. However, this process is very computationally demanding, and so the algorithm is slow and can only be allowed 6.67% of the training data, which is a drag on its predictive ability.

Model name	Model performance	Tuning parameters	R package
OLS		773662 (25.35%)	`elasticnet` 1.1
Ridge	772658 (25.33%)	$\lambda = 0.00172$	`elasticnet` 1.1
LASSO	773662 (25.35%)	$\lambda = 0$	`glmnet` $2.0 - 16$
Elastic Net	772658 (25.33%)	$\lambda = 0.00172$ & $\alpha = 1$	`glmnet` $2.0 - 16$
CART	858751 (31.96%)	tree depth = 33	`rpart` $4.1 - 13$
Bagged CART†	690021 (23.26%)	tree depth = 33	`caret` $6.0 - 79$
Random Forest†	694366 (23.18%)	subsample = 68.8%	`randomForest` $4.6 - 14$
BART‡	692602 (23.34%)	# trees = 200	`bartMachine` 1.2.4.2
XGboost	643168 (21.99%)	$M = 200$, tree depth = 9, $\eta = 0.6$, $b = 0.8$ & $\nu = 0.04$	`xgboost` 0.6.4.1
Neural Network	709237 (25.20%)	$\gamma_r = 0.05$ & # hidden features = 60	`nnet` $7.3 - 12$
Bayesian Neural Network‡	834450 (28.66%)	# hidden features = 40	`brnn` 0.6
SKAT		789726 (23.51%)	
N	17620		

Table 7.1: RMSE and MAPE model performance

Note: The table displays root mean squared errors (RMSE) and in parenthesis is mean absolute percentage errors (MAPE). In the second to most right column is the best tuned hyperparameter values as determined by cross-validation with K=5. Note also that the number of input variables, p, and the number of observations may differ between models.
†Ensemble of 30 trees.
‡BART and brnn are computationally demanding, and so only use 50% and 6.67% of the data in the prediction, respectively.

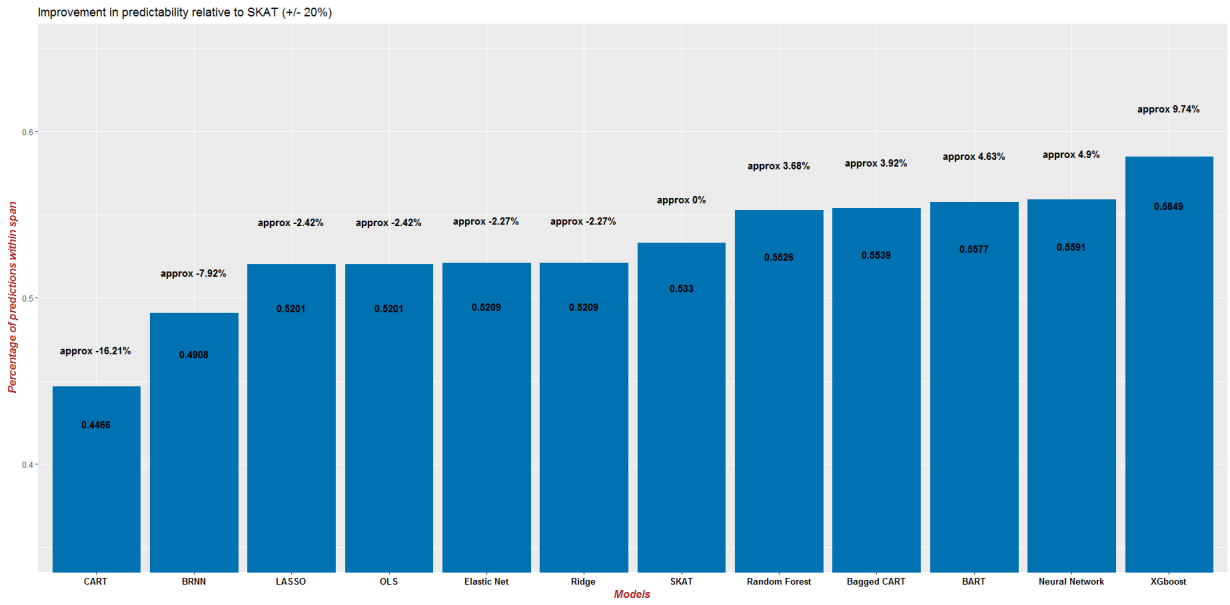

Figure 7.1: Improvement in the percentage of appraisals within $(+/-20\%)$ span of realised prices relative to SKAT in the hold-out sample of 2016

Note: The measure of the performance relative to SKAT is fairly uncertain since SKAT's appraisals on average is 9.5% higher than the prices, that enter into the analysis. On the other hand the new appraisals are purely mechanical, and there are not - as opposed to SKAT's appraisals - manual corrections.

7.3 Ability to predict house prices

Since Rigsrevisionen [2013] finds it necessary to have a viable frame of reference when valuing houses for taxation purposes, it becomes important to look at which models have the highest accuracy - no matter the method with which they extract their results. To this purpose, it is interesting that most of the machine learning algorithms in this thesis outperform both SKAT's former appraisals, that might already have been manually corrected by a professional appraiser, and the OLS model. Furthermore, the inability of the generalised linear models in capturing important complex effects makes it more interesting to look at neural networks and tree-based models, which are more adept at handling non-linear relationships. A downside to these types of methods is that they will come to different conclusions on the house price each time the models are run with different starting points or when different sub-samples of data are drawn. This is likely to make it less trustworthy to the general user, whether it is someone buying a house or the government valuing houses for taxation purposes. However, as the techniques gain popularity within both the general public and within policymaking as predicted by Athey [2017], machine learning should be considered more heavily in these matters too, since these initial assessments are highly encouraging.

However, before the models can be implemented, there is still a lot of work, and a lot of different ideas to be carried out. As seen in Figure C.11 on page 113, the best model, the XGboost model, has trouble predicting the prices of inexpensive homes. This could be because energy labels serve us badly as a proxy for the condition of the house, or because of the higher selling period in inexpensive areas compared to expensive areas, which gives the buyer more leverage, all else equal, in the buying situation. Furthermore, the cadastral area is not included, and other factors in the deal such as a lawn tractor and furniture cannot be included, which might affect the value of the house substantially. This is supported by the fact that there is more dispersion around the median in rural areas, as determined by the Coefficient of Dispersion (CoD) often calculated in Ratio Studies [IAAO, 2013a], which can be seen in Figure C.12[41]. These are examples of the high noise in house price data. In general, several other factors spill into this. For example, we cannot be too certain on the accuracy of the measurements of our inputs, there are a lot of individual factors in each specific sale situation, and many houses are inherently unique or at least has some sort of unobserved heterogeneity.

However, that does not imply that we cannot do even better. In Figure 7.2 on page 78 the errors of the best model are plotted spatially. As we can see from this visualisation there is still considerable spatial autocorrelation in the error terms; to be specific notice how the green values tend to cluster together in groups. Furthermore, the red numbers, those houses that have been overvalued by the model, tend partly to lie close to trafficked roads or railways - common knowledge tells us that these are noisy and devalues the house. These points imply that there still is a lot of scope for improving the models via improving the data and tailoring it to the specific models. Furthermore, if these methods should be used in the governmental issue of predicting house prices for taxation, then more backtesting and further statistics are necessary; how will the new house valuations affect the taxes on a municipal and regional level? Is it possible for a variable that will structurally enhance the valuation of a house to have an adverse effect under certain circumstances? Is the sample of sold single-family houses representative for the whole population of Danish single-family houses, so that the generalisation has satisfactory external validity? And further reliability and robustness measures, such as those suggested in IAAO [2013a] and IAAO [2013b].

To the other end of evaluating the predictive ability of machine learning algorithms compared to the standard econometric linear model, these results are very uplifting. As machine learning algorithms are better at handling complex relationships and noisy and high-dimensional data in

[41]The COD is defined as: $\text{COD} = \frac{100}{R_m}\left[\sum_{i=1}^{N}\left|\frac{R_i - R_m}{N}\right|\right]$, where $R_i \, \forall \, i = 1, ..., N$ is the ratio of the prediction to the realised price and $R_m = \text{median}\,(R)$ with $R = \{R_1, ..., R_N\}$

predictive issues, they massively outperform their linear peer. This thesis, then, echoes Mullainathan and Spiess [2017] and Athey [2017] in applauding and welcoming the impact machine learning will have on econometrics - and hope that the acceptance of these methods will increase for policymakers in the future.

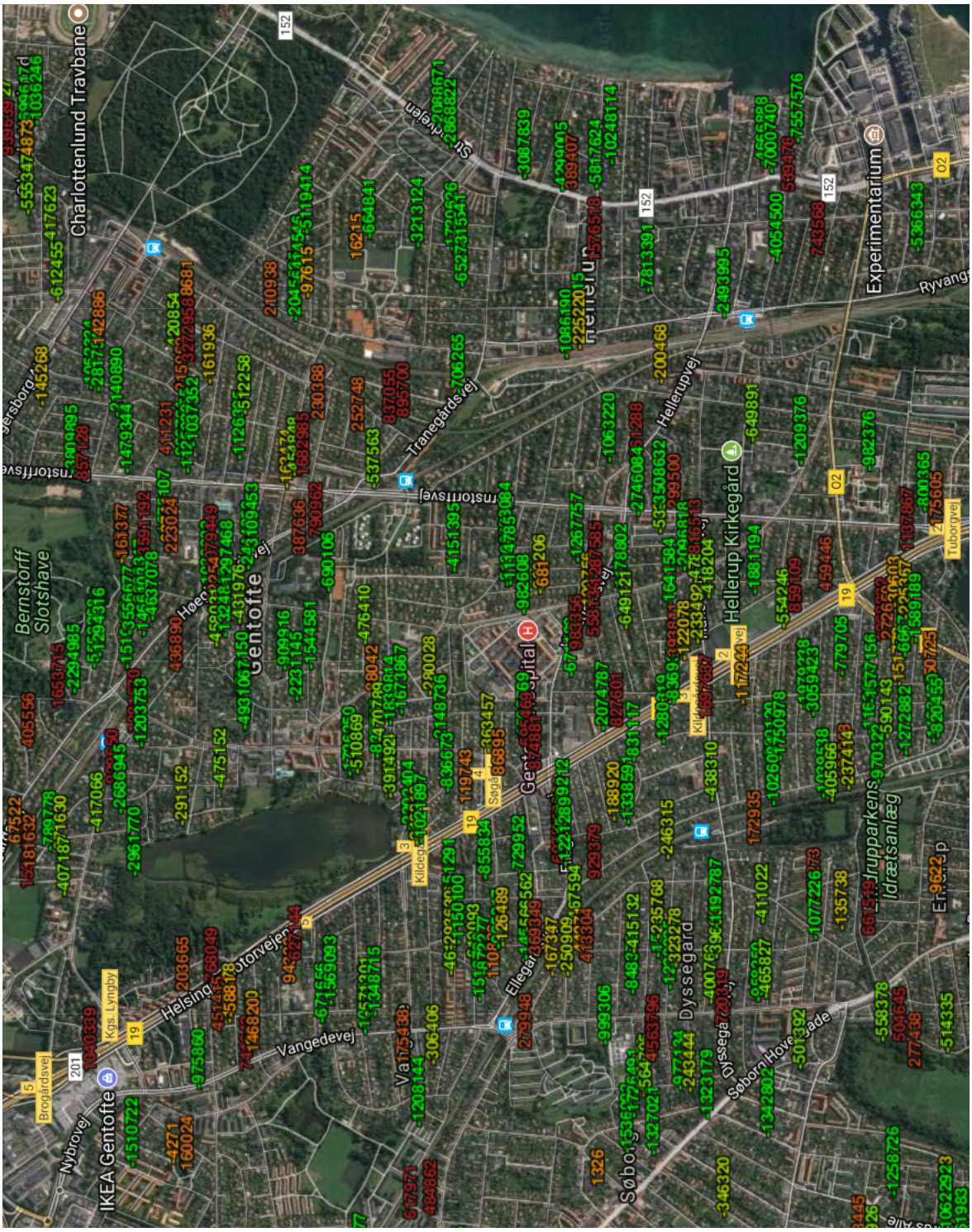

Figure 7.2: Errors of the best model visualised spatially in a northern part of the Copenhagen metropolitan area

Note: The green numbers show errors of the XGboost model where it has predicted a too low value and adversely the red numbers show where it has predicted a too high value.

8 Conclusion

In this thesis, an introduction to the use of machine learning algorithms for prediction is given, and these are applied to a very relevant issue. Several of central elements in the machine learning toolbox are looked at; the bias-variance trade-off; the estimation of various errors; model regularisation for reducing complexity; creating ensembles of models; and some individual algorithm types, respectively. A notable size of the thesis is devoted to the use of these methods within an application, that could be relevant for both private, corporate and governmental use, namely mass appraisal of houses.

The prediction of house prices is an exceptionally difficult task. The data is subject to a lot of noise because of the inherently unique situation each and every house sale is, and it is challenging to gather data of good quality. Furthermore, there is a huge amount of unobserved heterogeneity partly because each house is a composite of so many important factors, which may be prioritised differently for each individual house buyer. To add to the difficulty, these factors have complex relationships with each other and are in themselves probably not linear. In 2013 the Danish tax authority was deemed by the government not to have lived up to its responsibility to deliver precise, transparent and just valuations, and so the Danish governmental valuation of houses was temporarily suspended. An internal group within the Danish tax authority, ICE, was then created and assigned to the job of creating a better mass appraisal model. The current Danish laws effectively inhibit ICE from exploiting the advances in machine learning algorithms within prediction problems. In this thesis, though, it is argued that the use of machine learning algorithms to this issue should be reconsidered by the government since its predictive performance over its linear peer is substantial. Nonetheless, the use of good property appraisals goes beyond the scope of the government. For example, banks can use them when underwriting loan advances, home equity withdrawals and remortgaging or aspiring homeowners can use them to gain a good understanding of the housing market in different areas and possibly as a guideline for their offer.

It is argued that the problem of mass appraisal is inherently one of prediction, which makes it both interesting and concerning that most of the mass appraisal literature concerns itself with linear estimation or close cousins of it. Machine learning algorithms are explicitly concerned with prediction rather than consistent and unbiased estimation of parameters; this feature makes it more adept to these kinds of problems. Although it is recognised that there indeed are other issues involved in the use of appraisal models, prediction accuracy is the sole objective when evaluating the models.

From data gathered from several public Danish information sources, a range of different models is estimated - ranging from the linear model to generalised linear models to tree-based models and then to neural networks. These are compared to each other and to the suspended valuations of SKAT, which, though, are still being updated. It is found that the generalised linear models cannot exploit their abilities to regularise the linear model to the fullest since this involves creating an infeasibly large input matrix. Instead, methods that inherently model complex and non-linear variable relationships produce superior predictive abilities. Hitherto, the darling of academics who try to create an automated valuation model using machine learning models is the neural network. Even though the conclusion that neural networks are inferior to tree-based models in predicting house prices cannot be made as these can be greatly extended, it is found that the tree-based models produce the best results under the settings. In particular, the XGboost model, which has won many Kaggle competitions too, has the best predictive ability by a considerable margin. Its success is partly attributed to its fast algorithm, which allows assessing more tuning combinations within a limited time frame and partly attributed to its ability to detect and fit complex relationships using flexible regularisation. Its flexible regularisation comprises both elements known from random forests and bagged CART but also some of its nature from the boosting method.

Admittedly, the model would not be fit for direct use as of now, and thus suggestions on some issues to be dealt with going forward are made. These include both policy perspectives, issues regarding further model testing, and issues with the specific models one could continue to develop. Nonetheless, the conclusion of this thesis is twofold. First, the superiority of machine learning algorithms in contrast to the OLS-based prediction models is echoed. Second, while the techniques gain popularity within both the general public and within policy making, machine learning should be considered more heavily in Danish mass appraisal models since these initial assessments are highly encouraging.

References

Hirotugu Akaike. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723, 1974. 3.3.1

James H Albert and Siddhartha Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American statistical Association*, 88(422):669–679, 1993. 4.5.1

Evgeny A Antipov and Elena B Pokryshevskaya. Mass appraisal of residential apartments: An application of random forest for valuation and a cart-based approach for model diagnostics. *Expert Systems with Applications*, 39(2):1772–1778, 2012. 2.1

Susan Athey. The impact of machine learning on economics. In *Economics of Artificial Intelligence*. University of Chicago Press, 2017. 1.1, 7.3

Sabyasachi Basu and Thomas G Thibodeau. Analysis of spatial autocorrelation in house prices. *The Journal of Real Estate Finance and Economics*, 17(1):61–85, 1998. 2.1, 6.3.2

Richard E Bellman. *Adaptive control processes: a guided tour*. Princeton university press, 1961. 3.2

Mauro Birattari and Janusz Kacprzyk. *Tuning metaheuristics: a machine learning perspective*, volume 197. Springer, 2009. 7.1

Christopher M Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995. 3.4

Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996. 4.2, 4.2, 4.3, 5.2.2

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. 5.2.2

Leo Breiman. *Classification and regression trees*. Routledge, 2017. 5.2

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016. 5.2, 5.2.1, 5.2.3, 5.2.3

Vladimir Cherkassky and Filip M Mulier. *Learning from data: concepts, theory, and methods*. John Wiley & Sons, 2007. 1.2, 3, 3.1, 3.2, 3.4.1, 3.5, 3.5, 3.5, 4, 4.6

Hugh A Chipman, Edward I George, and Robert E McCulloch. Bayesian cart model search. *Journal of the American Statistical Association*, 93(443):935–948, 1998. 5.2.4

Hugh A Chipman, Edward I George, Robert E McCulloch, et al. Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298, 2010. 4.5.1, 5.2.4

Niels Arne Dam, Tina Saaby Hvolbøl, Peter Birch Sørensen, Erik Haller Pedersen, and Susanne Hougaard Thamsborg. Udviklingen på ejerboligmarkedet i de senere år - kan boligpriserne forklares? Technical report, Danmarks Nationalbank, 2011. 2.2, 5

Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in neural information processing systems*, pages 2933–2941, 2014. 4.6.2

Luc Devroye, László Györfi, and Gábor Lugosi. *A probabilistic theory of pattern recognition*, volume 31. Springer Science & Business Media, 2013. 3.4.1, 3.4.1, 3.5, 3.5

Persi Diaconis, Susan Holmes, and Richard Montgomery. Dynamical bias in the coin toss. *SIAM review*, 49(2):211–235, 2007. 3.1

Bradley Efron. Bootstrap methods: another look at the jackknife. *The annals of statistics*, 7(1): 1–26, 1979. 3.3.3

Bradley Efron and Robert Tibshirani. Improvements on cross-validation: the 632+ bootstrap method. *Journal of the American Statistical Association*, 92(438):548–560, 1997. A.3

Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004. 4.1.2

Liran Einav and Jonathan Levin. Economics in the age of big data. *Science*, 346(6210):1243089, 2014. 3

Graham Elliott and Allan Timmermann. *Economic Forecasting*. Princeton University Press, 2016. A.2

Theodoros Evgeniou, Massimiliano Pontil, and Tomaso Poggio. Regularization networks and support vector machines. *Advances in computational mathematics*, 13(1):1, 2000. 3.4.1

FD. Datagrundlaget for statistikken. Technical report, Finans Danmark, 2014. 6.2.3

F Dan Foresee and Martin T Hagan. Gauss-newton approximation to bayesian learning. In *Neural networks, 1997., international conference on*, volume 3, pages 1930–1935. IEEE, 1997. 5.3.1

LLdiko E Frank and Jerome H Friedman. A statistical view of some chemometrics regression tools. *Technometrics*, 35(2):109–135, 1993. 4.1

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 2. Springer series in statistics New York, 2009. 2.1, 3.2, 3.2, 3.2, 3.3, 3.3, 3.3, 3.3.1, 3.3.1, 3.3.2, 3.3.3, 3.3.4, 3.3.4, 3.4, 3.4.1, 3.5, 4.1, 4.1, 4.1.1, 4.1.2, 4.1.3, 4.2, 4.4, 4.5.4, 4.6.1, 4.6.1, 4.6.2, 5.2.1, 5.2.2, 5.3, 5.3, 5.3.1, 7.1, 7.2, A.3

Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001. 24, 4.6.2, 5.2.3

Jerome H Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38 (4):367–378, 2002. 4.6.2, 5.2.3

Wenjiang J Fu. Penalized regressions: the bridge versus the lasso. *Journal of computational and graphical statistics*, 7(3):397–416, 1998. 4.1

Seymour Geisser. The predictive sample reuse method with applications. *Journal of the American statistical Association*, 70(350):320–328, 1975. 3.3.4

Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Ieee transactions on pattern analysis and machine intelligence*, PAMI-6 (6):721–741, 1984. 4.5.3

Robert J Gloudemans and Dennis W Miller. Multiple regression analysis applied to residential properties. *Decision Sciences*, 7(2):294–304, 1976. 2.1

Gene H Golub, Michael Heath, and Grace Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979. 3.3.4

Allen C Goodman and Thomas G Thibodeau. Age-related heteroskedasticity in hedonic house price equations. *Journal of Housing Research*, pages 25–42, 1995. 2.1

Allen C Goodman and Thomas G Thibodeau. Housing market segmentation. *Journal of housing economics*, 7(2):121–143, 1998. 2.1

Daniel S Hamermesh. Six decades of top economics publishing: Who and how? *Journal of Economic Literature*, 51(1):162–72, 2013. 3

Trevor Hastie, Robert Tibshirani, et al. Bayesian backfitting (with comments and a rejoinder by the authors. *Statistical Science*, 15(3):196–223, 2000. 27

W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. 4.5.2

Kathy J Hayes and Lori L Taylor. Neighborhood school characteristics: what signals quality to homebuyers? *Economic Review-Federal Reserve Bank of Dallas*, page 2, 1996. 6.3.2

Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. 4.1.1

Jennifer A Hoeting, David Madigan, Adrian E Raftery, and Chris T Volinsky. Bayesian model averaging: a tutorial. *Statistical science*, pages 382–401, 1999. 4.5.4, 4.5.4

Simon Juul Hviid. A regional model of the danish housing market. Technical report, Danmarks Nationalbank, 2017. 2.2, 6.3.2

Simon Juul Hviid and Paul Lassenius Kramp. Aftale om boligskat stabiliserer boligpriser. Technical report, Danmarks Nationalbank, 2017. 2.2, 2.2

IAAO. Standard on ratio studies, 2013a. 7.3

IAAO. Standard on mass appraisal of real property, 2013b. 2.1, 7.3

ICE. Nye og mere retvisende ejendomsvurderinger. Technical report, Skatteministeriet, 2016. 1.1, 1.4, 2.1, 7

Erik Haller Pedersen & Jakob Isaksen. Recent housing market trends. Technical report, Danmarks Nationalbank, 2015. 2.2

Simon Jackman. *Bayesian analysis for the social sciences*, volume 846. John Wiley & Sons, 2009. 3.1, 4.5.1, 4.5.2, 4.5.2, 20, 4.5.2, 4.5.3, 4.5.3

Peter Engberg Jensen, Lise Gronø, Preben Lund Larsen, and Søren Leth-Petersen. Forbedring af ejendomsvurderingen. Technical report, Skatteministeriet, 2014. 2.1

Adam Kapelner and Justin Bleich. bartmachine: Machine learning with bayesian additive regression trees. *arXiv preprint arXiv:1312.2171*, 2013. 5.2.4

Ruud M Kathmann. Neural networks for the mass appraisal of real estate. *Computers, environment and urban systems*, 17(4):373–384, 1993. 2.1

Tom Kauko and Maurizio d'Amato. Introduction: Suitability issues in mass appraisal methodology. *Mass appraisal methods: An international perspective for property valuers*, pages 1–24, 2008. 2.1

Asbjørn Klein, Simon Juul Hviid, Tina Saaby Hvolbøl, Paul Lassenius Kramp, and Erik Haller Pedersen. Boligprisbobler og fordelene ved en stabiliserende boligbeskatning. Technical report, Danmarks Nationalbank, 2016. 2.2, 2.2

Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995. 3.3.4

Max Kuhn et al. Caret package. *Journal of statistical software*, 28(5):1–26, 2008. 7

Visit Limsombunchai. House price prediction: hedonic price model vs. artificial neural network. In *New Zealand Agricultural and Resource Economics Society Conference*, pages 25–26, 2004. 2.1

David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992. 4.5.1

Jonathan Mark and Michael A Goldberg. Multiple regression analysis and mass assessment: A review of the issues. *Appraisal Journal*, 56(1), 1988. 2.1

William J McCluskey, Dzurllkanian Zulkarnain Daud, and Norhaya Kamarudin. Boosted regression trees: An application for the mass appraisal of residential property in malaysia. *Journal of Financial Management of Property and Construction*, 19(2):152–167, 2014. 2.1, 6.3.2

WJ McCluskey, M McCord, PT Davis, M Haran, and D McIlhatton. Prediction accuracy in mass appraisal: a comparison of modern approaches. *Journal of Property Research*, 30(4):239–265, 2013. 2.1, 6.2.2, 6.3.2

Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953. 4.5.2

Sendhil Mullainathan and Jann Spiess. Machine learning: an applied econometric approach. *Journal of Economic Perspectives*, 31(2):87–106, 2017. 1.2, 2.1, 3.2, 7.3

Sigurd Næss-Schmidt, Christian Heebøl, and Niels Christian Fredslund. Do homes with better energy efficiency ratings have higher house prices? econometric approach. Technical report, Danish Energy Agency, 2015. 6.2.1, 6.3.1

Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012. 4.5.1, 4.5.1, 5.3.1

John R Ottensmann, Seth Payton, Joyce Man, et al. Urban location and housing prices within a hedonic model. *Journal of Regional Analysis and Policy*, 38(1):19–35, 2008. 2.1, 6.3.2

Seth Payton, Greg Lindsey, Jeff Wilson, John R Ottensmann, and Joyce Man. Valuing the benefits of the urban forest: a spatial hedonic approach. *Journal of Environmental Planning and Management*, 51(6):717–736, 2008. 2.1

Stig Uffe Pedersen. Bekendtgørelse om håndbog for energikonsulenter. Technical report, Energi-, Forsynings- og Klimaministeriet, 2016. 6.2.1

Steven Peterson and Albert Flanagan. Neural network hedonic pricing models in mass real estate appraisal. *Journal of Real Estate Research*, 31(2):147–164, 2009. 2.1

Rigsrevisionen. Beretning til statsrevisorerne om den offentlige ejendomsvurdering. Technical report, 2013. 1.1, 6.2.3, 6.4, 7.3

Paulino Perez Rodriguez and Daniel Gianola. *brnn: Bayesian Regularization for Feed-Forward Neural Networks*, 2016. URL `https://CRAN.R-project.org/package=brnn`. R package version 0.6. 5.3.1

Sherwin Rosen. Hedonic prices and implicit markets: product differentiation in pure competition. *Journal of political economy*, 82(1):34–55, 1974. 2.1

Robert E Schapire and Yoav Freund. *Boosting: Foundations and algorithms*. MIT press, 2012. 2.1, 4, 4.6

Rainer Schulz, Martin Wersing, and Axel Werwatz. Automated valuation modelling: a specification exercise. *Journal of Property Research*, 31(2):131–153, 2014. 2.1, 6.2.3, 6.3.1

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014. 3.4.1

Roger W Sinnott. Virtues of the haversine. *Sky Telesc.*, 68:159, 1984. 6.2.2

Mervyn Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the royal statistical society. Series B (Methodological)*, pages 111–147, 1974. 3.3.4

Ilya Sutskever. Training recurrent neural networks. *University of Toronto, Toronto, Ont., Canada*, 2013. 5.3

Danny PH Tay and David KH Ho. Artificial intelligence and the mass appraisal of residential apartments. *Journal of Property Valuation and Investment*, 10(2):525–540, 1992. 2.1

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996. 4.1.2, 4.1.2

Robert Tibshirani and Keith Knight. Model search by bootstrap 'bumping'. *Journal of Computational and Graphical Statistics*, 8(4):671–686, 1999. 4.3, 17

Vladimir N Vapnik and A Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971. 3.5

Vladimir N Vapnik and Alexey J Chervonenkis. Theory of pattern recognition. 1974. 3.5

Hal R Varian. Big data: New tricks for econometrics. *Journal of Economic Perspectives*, 28(2): 3–28, 2014. 3.1, 7

David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992. 4.4

Arnold Zellner and Peter E Rossi. Bayesian analysis of dichotomous quantal response models. *Journal of Econometrics*, 25(3):365–393, 1984. 4.5.1

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005. 4.1, 4.1.3, 4.1.3

A Mathematical Derivations

A.1 Optimism of the training error rate

Let $\hat{f}(x_i) = \hat{y}_i$, then

$$
\begin{aligned}
\omega &= \mathbb{E}_y\left[\text{op}\right] \\
&= \mathbb{E}_y\left[\text{Err}_{\text{in}} - \overline{\text{err}}\right] \\
&= \mathbb{E}_y\left[\text{Err}_{\text{in}}\right] - \mathbb{E}_y\left[\overline{\text{err}}\right] \\
&= \mathbb{E}_y\left[\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{Y^0}\left[L(Y_i^0, \hat{f}(x_i))|\mathcal{T}\right]\right] - \mathbb{E}_y\left[\frac{1}{N}\sum_{i=1}^{N}L(y_i, \hat{f}(x_i))\right] \\
&= \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_y\mathbb{E}_{Y^0}\left[(Y_i^0 - \hat{y}_i)^2|\mathcal{T}\right] - \mathbb{E}_y\left[(y_i - \hat{y}_i)^2\right] \\
&= \frac{1}{N}\sum_{i=1}^{N}\Big[\mathbb{E}_y\mathbb{E}_{Y^0}\left[(Y_i^0)^2|\mathcal{T}\right] + \mathbb{E}_y\mathbb{E}_{Y^0}\left[(\hat{y}_i)^2|\mathcal{T}\right] - 2\mathbb{E}_y\mathbb{E}_{Y^0}\left[Y_i^0\hat{y}_i|\mathcal{T}\right] \\
&- \mathbb{E}_y\left[y_i^2\right] - \mathbb{E}_y\left[(\hat{y}_i)^2\right] + 2\mathbb{E}_y\left[y_i\hat{y}_i\right]\Big] \\
&= \frac{1}{N}\sum_{i=1}^{N}\Big[\mathbb{E}_y\left[y_i^2\right] + \mathbb{E}_y\left[(\hat{y}_i)^2\right] - 2\mathbb{E}_y\left[y_i\right]\mathbb{E}_y\left[\hat{y}_i\right] \\
&- \mathbb{E}_y\left[y_i^2\right] - \mathbb{E}_y\left[(\hat{y}_i)^2\right] + 2\mathbb{E}_y\left[y_i\hat{y}_i\right]\Big] \\
&= \frac{2}{N}\sum_{i=1}^{N}\left[\mathbb{E}_y\left[y_i\hat{y}_i\right] - \mathbb{E}_y\left[y_i\right]\mathbb{E}_y\left[\hat{y}_i\right]\right] \\
&= \frac{2}{N}\sum_{i=1}^{N}\left[\mathbb{E}_y\left[y_i\hat{y}_i\right] - \mathbb{E}_y\left[y_i\right]\mathbb{E}_y\left[\hat{y}_i\right] - \mathbb{E}_y\left[y_i\right]\mathbb{E}_y\left[\hat{y}_i\right] + \mathbb{E}_y\left[y_i\right]\mathbb{E}_y\left[\hat{y}_i\right]\right] \\
&= \frac{2}{N}\sum_{i=1}^{N}\mathbb{E}_y\left[y_i\hat{y}_i + \mathbb{E}_y\left[y_i\right]\mathbb{E}_y\left[\hat{y}_i\right] - y_i\mathbb{E}_y\left[\hat{y}_i\right] + \mathbb{E}_y\left[y_i\right]\hat{y}_i\right] \\
&= \frac{2}{N}\sum_{i=1}^{N}\mathbb{E}_y\left[\left(\hat{y}_i - \mathbb{E}_y\left[\hat{y}_i\right]\right)\left(y_i - \mathbb{E}_y\left[y_i\right]\right)\right] \\
&= \frac{2}{N}\sum_{i=1}^{N}\text{Cov}\left(\hat{y}_i, y_i\right)
\end{aligned}
$$

A.2 The information criteria

In a case of K parametric models $\mathcal{M}_K$, where each model $M_k \in \mathcal{M}_K$, $k = 1, ..., K$ requires estimating d parameters $\theta \in \mathbb{R}^d$. The BIC, AIC and HQ choose the model to minimise the expressions of the form

$$\text{IC}(k) = -2N^{-1}\log[\mathbb{P}_{\hat{\theta}}(Y)] + h(n_k)g(N)$$

over all models, where $\mathbb{P}_{\hat{\theta}}(Y)$ is the likelihood of the data evaluated at the parameter estimates and $h(d)$ is the penalising expression of model complexity, which is increasing in d, and $g(N)$ is a function decreasing in sample size, N. AIC, BIC and HQ now looks like this:

$$\begin{aligned}
\text{BIC}: &\quad -2N^{-1}\log[\mathbb{P}_{\hat{\theta}}(Y)] + d\log(N)N^{-1} \\
\text{AIC}: &\quad -2N^{-1}\log[\mathbb{P}_{\hat{\theta}}(Y)] + 2dN^{-1} \\
\text{HQ}: &\quad -2N^{-1}\log[\mathbb{P}_{\hat{\theta}}(Y)] + 2d\log(\log(N))N^{-1}
\end{aligned}$$

[Elliott and Timmermann, 2016].

A.3 On bootstrapping

The derivation of the ".632 estimator" is complex, but intuitively it pulls the leave-one out bootstrap down toward the training error rate, and hence reduces its upward bias [Friedman et al., 2009]. The constant of the estimate is derived from the probability of a given observation i being in a given sample b, as the contribution to the bootstrap estimate will be zero otherwise:

$$\begin{aligned}
\Pr(\text{observation } i \in \text{bootstrap sample } b) &= 1 - \left(1 - \frac{1}{N}\right)^N \\
&\approx 1 - e^{-1} \\
&= 0.632.
\end{aligned}$$

The 0.632 estimator is defined as

$$\widehat{\text{Err}}^{(1)} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|C^{-i}|}\sum_{b \in C^{-i}} L(y_i, \hat{f}^{*b}(x_i)),$$

Where C^{-i} is the set of indices of the bootstrap samples b that do not contain observation i

and $|C^{-i}|$ is the number of such samples.

As the .632 estimator can break down in overfit situations other measures have been suggested. For more information see Efron and Tibshirani [1997] for the derivation and discussion of the benefits of the ".632+ estimator"

A.4 Bayes estimates

First, to see how the Bayesian estimates with given priors are equivalent to the minimisation with regularisation specification we rewrite the likelihood function.

We assume that $Y_1, Y_2, ..., Y_n$ are independent and $Y_i \sim \mathcal{N}(\beta^T x_i, \sigma^2)$ where $x_i \in \mathbb{R}^p$. Then

$$\begin{aligned}
\mathbb{P}(Y|\beta,\sigma) &= \prod_{i=1}^{N} Y_i \\
&= \prod_{i=1}^{N} \left(\mathcal{N}(x_i\beta, \sigma^2)\right) \\
&= \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(Y_i - x_i\beta)^2}{2\sigma^2}\right) \\
&= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(Y - \boldsymbol{X}\beta)^T(Y - \boldsymbol{X}\beta)}{2\sigma^2}\right)
\end{aligned}$$

Where we can rewrite $(Y - \boldsymbol{X}\beta)^T(Y - \boldsymbol{X}\beta)$ such that

$$\begin{aligned}
(Y - \boldsymbol{X}\beta)^T(Y - \boldsymbol{X}\beta) &= (Y^T - \beta^T\boldsymbol{X}^T)(Y - \boldsymbol{X}\beta) \\
&= Y^TY - Y^T\boldsymbol{X}\beta - \beta^T\boldsymbol{X}^TY + \beta^T\boldsymbol{X}^T\boldsymbol{X}\beta \\
&= \beta^T\boldsymbol{X}^T\boldsymbol{X}\beta - 2Y^T\boldsymbol{X}\beta + Y^TY
\end{aligned}$$

Where we add and subtract $(Y - \boldsymbol{X}\hat{\beta})^T(Y - \boldsymbol{X}\hat{\beta})$ where $\hat{\beta} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T y = \hat{\beta}_{\text{ML}} = \hat{\beta}_{\text{OLS}}$ such that

$$
\begin{aligned}
&(Y - \boldsymbol{X}\beta)^T(Y - \boldsymbol{X}\beta) \\
&= \beta^T\boldsymbol{X}^T\boldsymbol{X}\beta - 2\beta^T\boldsymbol{X}^TY + 2\hat{\beta}^T\boldsymbol{X}^TY - \hat{\beta}^T\boldsymbol{X}^T\boldsymbol{X}\hat{\beta} + (Y - \boldsymbol{X}\hat{\beta})^T(Y - \boldsymbol{X}\hat{\beta}) \\
&= \beta^T\boldsymbol{X}^T\boldsymbol{X}\beta + 2\left(\hat{\beta} - \beta\right)^T\boldsymbol{X}^T\boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^TY - \hat{\beta}^T\boldsymbol{X}^T\boldsymbol{X}\hat{\beta} + (Y - \boldsymbol{X}\hat{\beta})^T(Y - \boldsymbol{X}\hat{\beta}) \\
&= \beta^T\boldsymbol{X}^T\boldsymbol{X}\beta + 2\left(\hat{\beta} - \beta\right)^T\boldsymbol{X}^T\boldsymbol{X}\hat{\beta} - \hat{\beta}^T\boldsymbol{X}^T\boldsymbol{X}\hat{\beta} + (Y - \boldsymbol{X}\hat{\beta})^T(Y - \boldsymbol{X}\hat{\beta}) \\
&= \beta^T\boldsymbol{X}^T\boldsymbol{X}\beta + 2\hat{\beta}^T\boldsymbol{X}^T\boldsymbol{X}\hat{\beta} - 2\beta^T\boldsymbol{X}^T\boldsymbol{X}\hat{\beta} - \hat{\beta}^T\boldsymbol{X}^T\boldsymbol{X}\hat{\beta} + (Y - \boldsymbol{X}\hat{\beta})^T(Y - \boldsymbol{X}\hat{\beta}) \\
&= \left(\beta - \hat{\beta}\right)^T\boldsymbol{X}^T\boldsymbol{X}\left(\beta - \hat{\beta}\right) + (Y - \boldsymbol{X}\hat{\beta})^T(Y - \boldsymbol{X}\hat{\beta})
\end{aligned}
$$

and so

$$
\mathbb{P}(Y|\beta,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{\left(\beta - \hat{\beta}\right)^T\boldsymbol{X}^T\boldsymbol{X}\left(\beta - \hat{\beta}\right) + (Y - \boldsymbol{X}\hat{\beta})^T(Y - \boldsymbol{X}\hat{\beta})}{2\sigma^2}\right). \tag{A.1}
$$

Now, to see the equivalence with the ridge estimate, we assume that β_j has a prior distribution $\beta_j \sim N(0,\tau^2)$, $j = 1,...,p$, where $\beta_1,...,\beta_p$ are independent and $\lambda = \sigma^2/\tau^2$ and $\hat{\beta}(\lambda)$ is the ridge estimate of β. We start with the Bayesian posterior of β, and then utilise the form of the likelihood in equation (A.1).

$$
\begin{aligned}
\mathbb{P}(\beta|Y,\lambda,\sigma) &= \frac{\mathbb{P}(Y|\beta,\sigma,\lambda)\cdot\mathbb{P}(\beta|\lambda,\sigma)}{\int \mathbb{P}(Y|\beta,\sigma,\lambda)d\beta} \\
&\propto \mathbb{P}(Y|\beta,\sigma,\lambda)\cdot\mathbb{P}(\beta|\lambda,\sigma) \\
&\propto \exp\left(-\frac{1}{2\sigma^2}(\beta-\hat{\beta})^T\boldsymbol{X}^T\boldsymbol{X}(\beta-\hat{\beta})\right)\cdot\frac{1}{\sqrt{2\pi\tau^2}}\exp\left(-\frac{\beta^T\beta}{2\tau^2}\right) \\
&\propto \exp\left(-\frac{1}{2\sigma^2}(\beta-\hat{\beta})^T\boldsymbol{X}^T\boldsymbol{X}(\beta-\hat{\beta})-\frac{\beta^T\beta}{2\tau^2}\right) \\
&= \exp\left(-\frac{1}{2\sigma^2}(\beta-\hat{\beta})^T\boldsymbol{X}^T\boldsymbol{X}(\beta-\hat{\beta})-\frac{\beta^T\beta\lambda-\hat{\beta}^T\hat{\beta}\lambda+\hat{\beta}^T\hat{\beta}\lambda}{2\sigma^2}\right) \\
&= \exp\left(-\frac{1}{2\sigma^2}\left((\beta-\hat{\beta})^T\boldsymbol{X}^T\boldsymbol{X}(\beta-\hat{\beta})+\beta^T\beta\lambda-\hat{\beta}^T\hat{\beta}\lambda+\hat{\beta}^T\hat{\beta}\lambda\right)\right) \\
&= \exp\left(-\frac{1}{2\sigma^2}\left((\beta-\hat{\beta})^T\left(\boldsymbol{X}^T\boldsymbol{X}+\lambda\boldsymbol{I}_p\right)(\beta-\hat{\beta})-\hat{\beta}^T\hat{\beta}\lambda\right)\right) \\
&\propto \exp\left(-\frac{1}{2\sigma^2}\left((\beta-\hat{\beta})^T\left(\boldsymbol{X}^T\boldsymbol{X}+\lambda\boldsymbol{I}_p\right)(\beta-\hat{\beta})\right)\right).
\end{aligned}
$$

Which we can maximise to get the ridge estimator in equation (4.3).

When it comes to the LASSO estimate we instead choose the Laplacian prior $f(\beta_i) = 1/(2\tau^2)\exp\left(-|\beta_i|/\tau^2\right)$ and now we instead have that

$$
\begin{aligned}
\mathbb{P}(\beta|Y,\lambda,\sigma) &= \frac{\mathbb{P}(Y|\beta,\sigma,\lambda)\cdot\mathbb{P}(\beta|\lambda,\sigma)}{\int \mathbb{P}(Y|\beta,\sigma,\lambda)d\beta} \\
&\propto \mathbb{P}(Y|\beta,\sigma,\lambda)\cdot L(\beta|\lambda,\sigma) \\
&\propto \exp\left(-\frac{1}{2\sigma^2}(\beta-\hat{\beta})^T\boldsymbol{X}^T\boldsymbol{X}(\beta-\hat{\beta})\right)\cdot\frac{1}{2\tau^2}\exp\left(-\frac{||\beta||}{\tau^2}\right) \\
&\propto \exp\left(-\frac{1}{2\sigma^2}(\beta-\hat{\beta})^T\boldsymbol{X}^T\boldsymbol{X}(\beta-\hat{\beta})\right)\exp\left(-\frac{\lambda||\beta||}{\sigma^2}\right) \\
&= \exp\left(-\frac{1}{2\sigma^2}\left((\beta-\hat{\beta})^T\boldsymbol{X}^T\boldsymbol{X}(\beta-\hat{\beta})+2\lambda||\beta||\right)\right).
\end{aligned}
$$

Where the minimisation hereof is equivalent to that of equation (4.4) up to a rescaling of λ.

B Data visualisations

B.1 Energy labels

2006 Scale	2008V1 Scale	2008V2 Scale	2012 Scale	2013 Scale	Limits in kWh/sqm/year
-	-	-	-	A2020	20
-	-	-	A1	A2015	≤ 30,0 + 1000/A
A1,A2	A	A1,A2	A2	A2010	≤ 52,5 + 1650/A
B1	B	B	B	B	≤ 70,0 + 2200/A
B2,C1	C	C	C	C	≤ 110 + 3200/A
C2,D1	D	D	D	D	≤ 150 + 4200/A
D2,E1	E	E	E	E	≤ 190 + 5200/A
E2,F1	F	F	F	F	≤ 240 + 6500/A
F2,G1,G2	G	G	G	G	> 240 + 6500/A

Figure B.1: Conversion table for energy labelling

B.2 External buildings' characteristics

Carport		Garage	
Exterior wall	**Roofing**	**Exterior wall**	**Roofing**
Wood (52.18%)	Built-up (13.208%)	Brick (65.501%)	Fibre cement (36.891%)
Brick (14.37%)	Metal (11.511%)	Wood (12.271%)	Built-up (14.816%)
Lightweight concrete (0.901%)	Fibre cement (8.653%)	Lightweight concrete (6.684%)	Tile (12.356%)
Metal (0.477%)	Roofing felt (7.615%)	Metal (0.445%)	Roofing felt (9.859%)
Concrete (0.066%)	Cement (5.287%)	Concrete (0.416%)	Metal (4.967%)
Other (32.005%)	Tile (4.025%)	Other (14.681%)	Cement (4.882%)
	Other (49.701%)		Other (16.229%)

Annex	
Exterior wall	**Roofing**
Brick (39.088%)	Fibre cement (28.477%)
Wood (30.315%)	Roofing felt (12.203%)
Lightweight concrete (4.160%)	Metal (8.534%)
Metal (1.173%)	Tile (8.356%)
Concrete (0.236%)	Cement (6.043%)
Other (25.028%)	Built-up (4.386%)
	Other (32.000%)

Table B.1: Descriptive statistics of external buildings' characteristics

Note: The "Other" category includes missing variables in these statistics, and, if needed, this will be the base category.

B.3 Distances and other summary statistics

Name of input variable	Mean	Median	5TH percentile	95TH percentile
Distances to buildings				
Nursing home	4.58	2.24	0.26	16.09
Daycare centre	1.07	0.51	0.13	4.25
Health centre	9.15	4.53	0.44	29.74
Hospital	4.40	2.91	0.39	12.82
Primary school	6.08	3.72	0.42	21.98
High school	1.12	0.65	0.16	3.88
University	35.92	29.43	2.95	87.96
Other teaching institution	6.70	3.32	0.42	25.69
Transport building	1.39	0.76	0.14	4.79
Distances to cities				
Copenhagen	144.62	160.96	8.32	259.55
Aarhus	106.62	110.08	20.69	161.42
Odense	110.39	113.25	17.51	207.11
$\cdots$ other cities excluded				
Closest of 50 cities	14.29	10.07	1.01	38.00
Distances to GeoDanmark data				
City centre	9.96	8.16	0.81	26.42
Train station	9.77	3.56	0.51	27.46
Windmill	4.51	3.77	1.01	10.56
Lake	8.26	6.24	1.44	20.51
Wood	4.40	3.67	1.31	9.99
Coast	9.88	5.71	0.39	37.99
Square metre prices				
Neighbourhood m^2 price†	15964	12935	5402	37462
Neighbourhood m^2 price‡	15907	12831	5319	37691
Neighbourhood m^2 price°	15856	12737	5981	37391
Municipality level variables and others				
Spending (per citizen)	56856	56523	50895	65042
School spending (per pupil)	62203	62066	52900	71930
Cultural spending (per citizen)	1582	1508	951	2364
Debt (per citizen)	16670	15650	7633	28280
Family wealth 2016	2041712	1796446	1514223	3688995
Unemployment level	4.28	4.30	2	6.3
Percentage non-western	1.96	1.80	1.04	3.94
House Price Index from Statistics Denmark	101.91	97.70	87.26	120.57
N	179952			

Table B.2: Descriptive statistics of locational amenities

Note: The summary statistics of area type is only from the subset of houses that has the given type of building.

† Unweighted.

‡ Weighted by the inverse of the distance.

° KNN regression within municipality.

B.4 Plots of geographically distributed items related to input variables

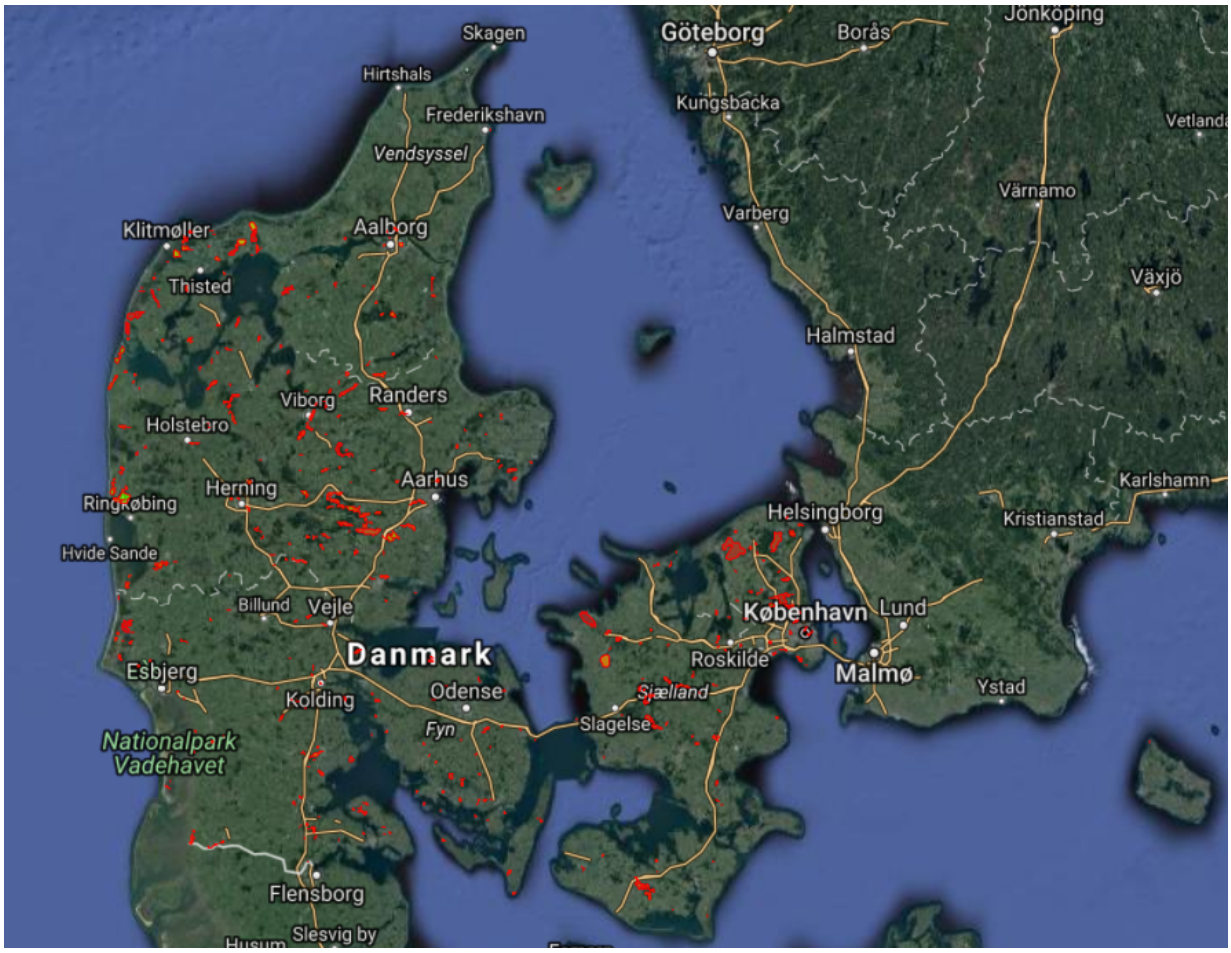

Figure B.2: The 469 lakes in the analysis

Note: Preferably more lakes would have been included in the analysis.

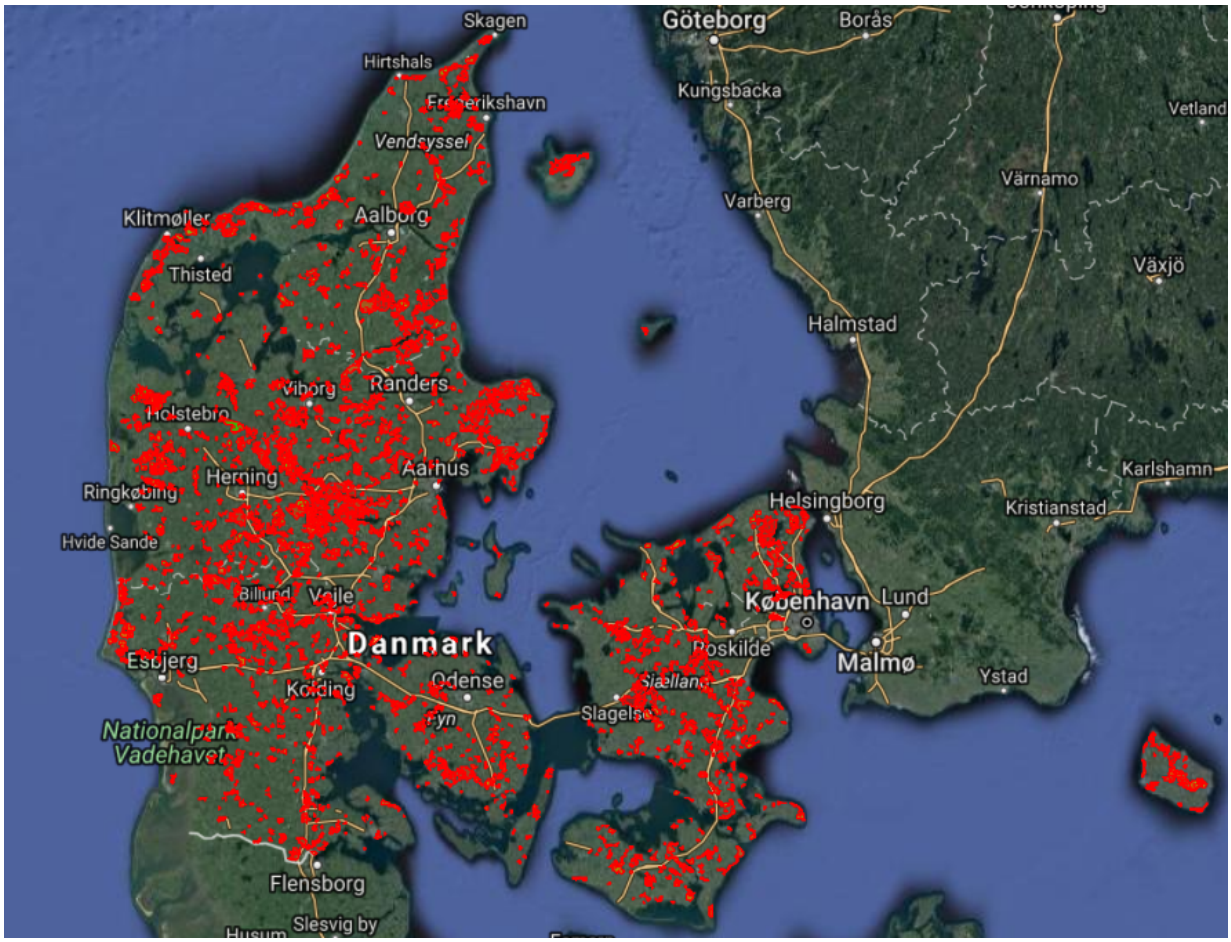

Figure B.3: The 1830 forests in the analysis

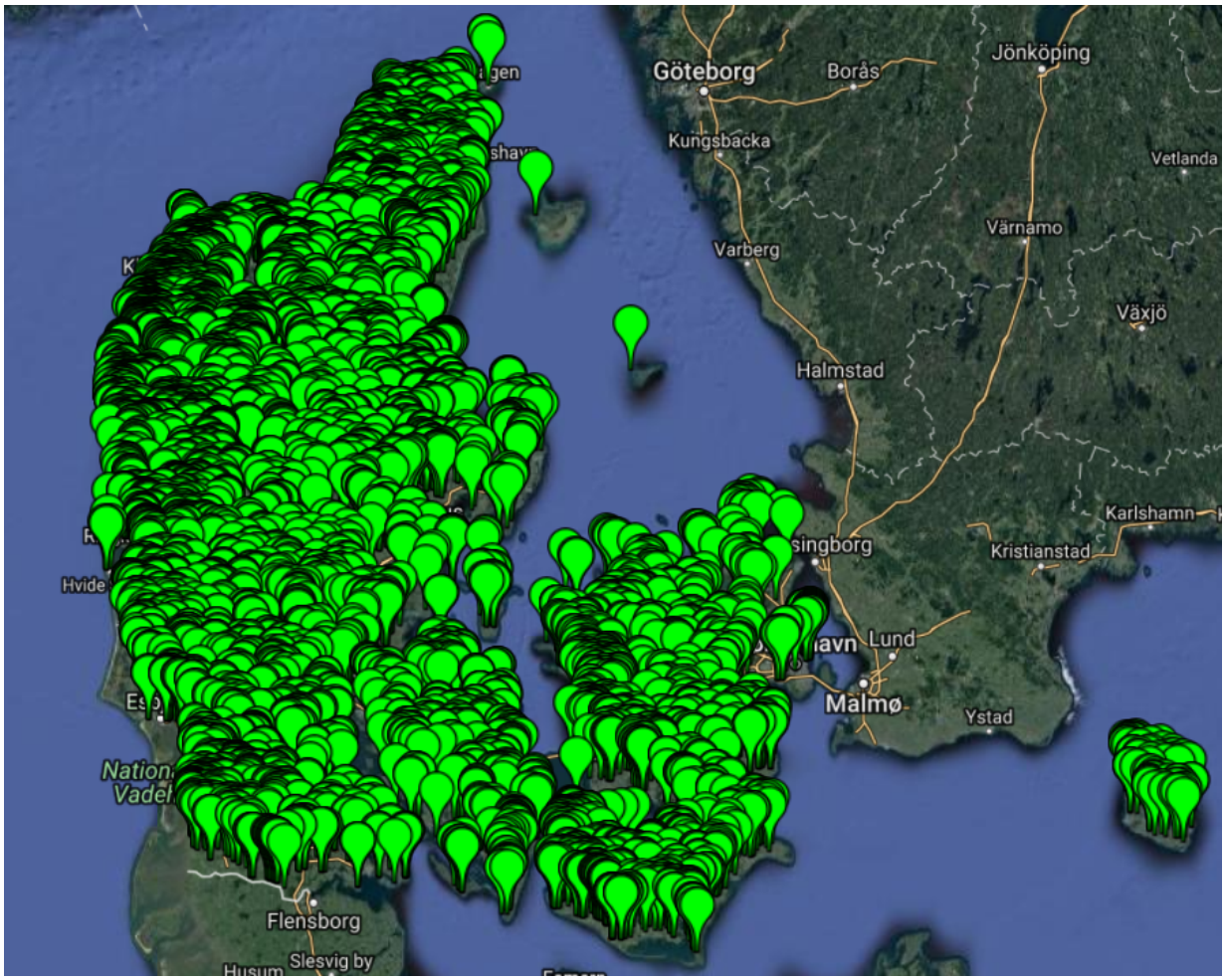

Figure B.4: All 6175 windmill type construction

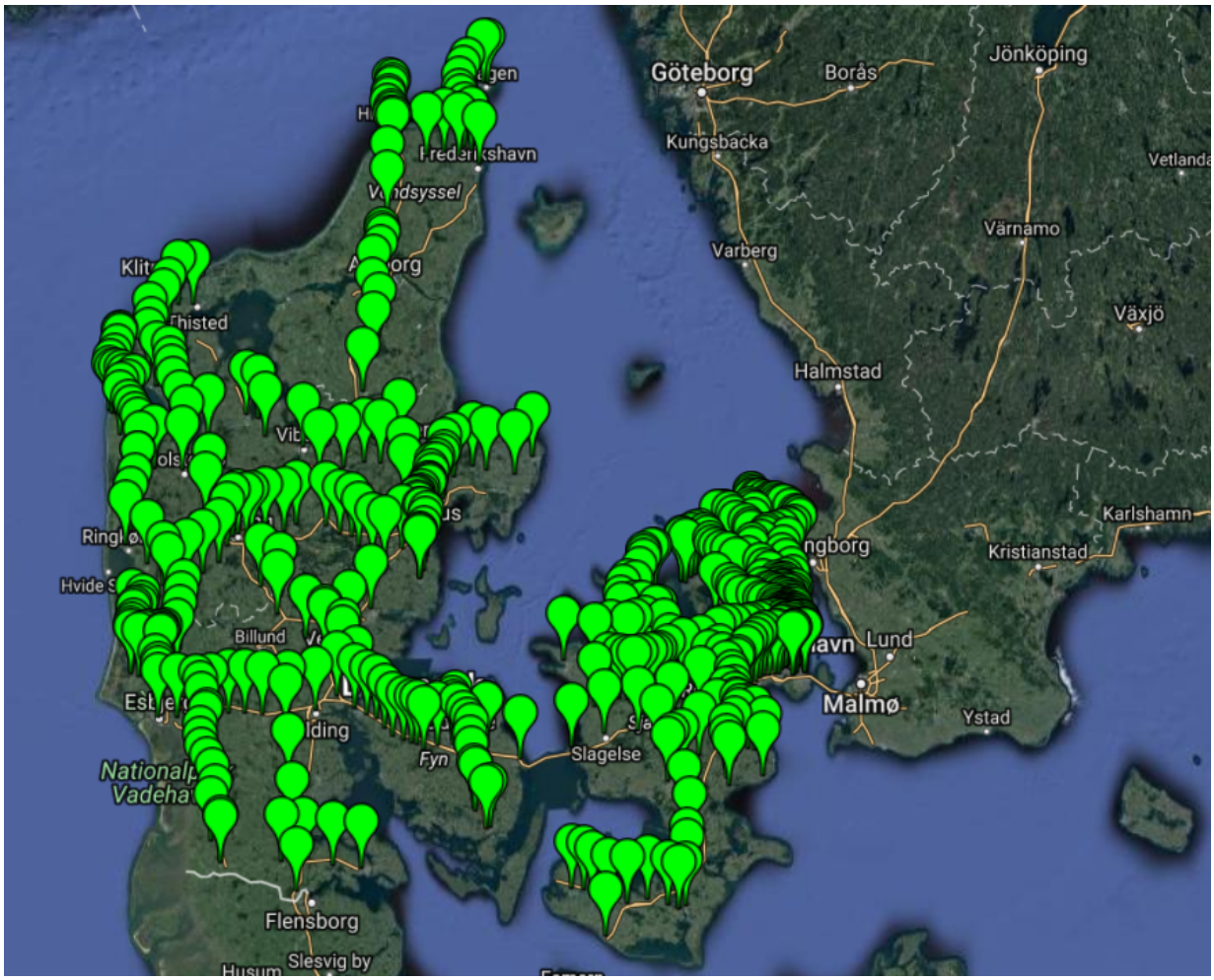

Figure B.5: All 503 train stations in Denmark

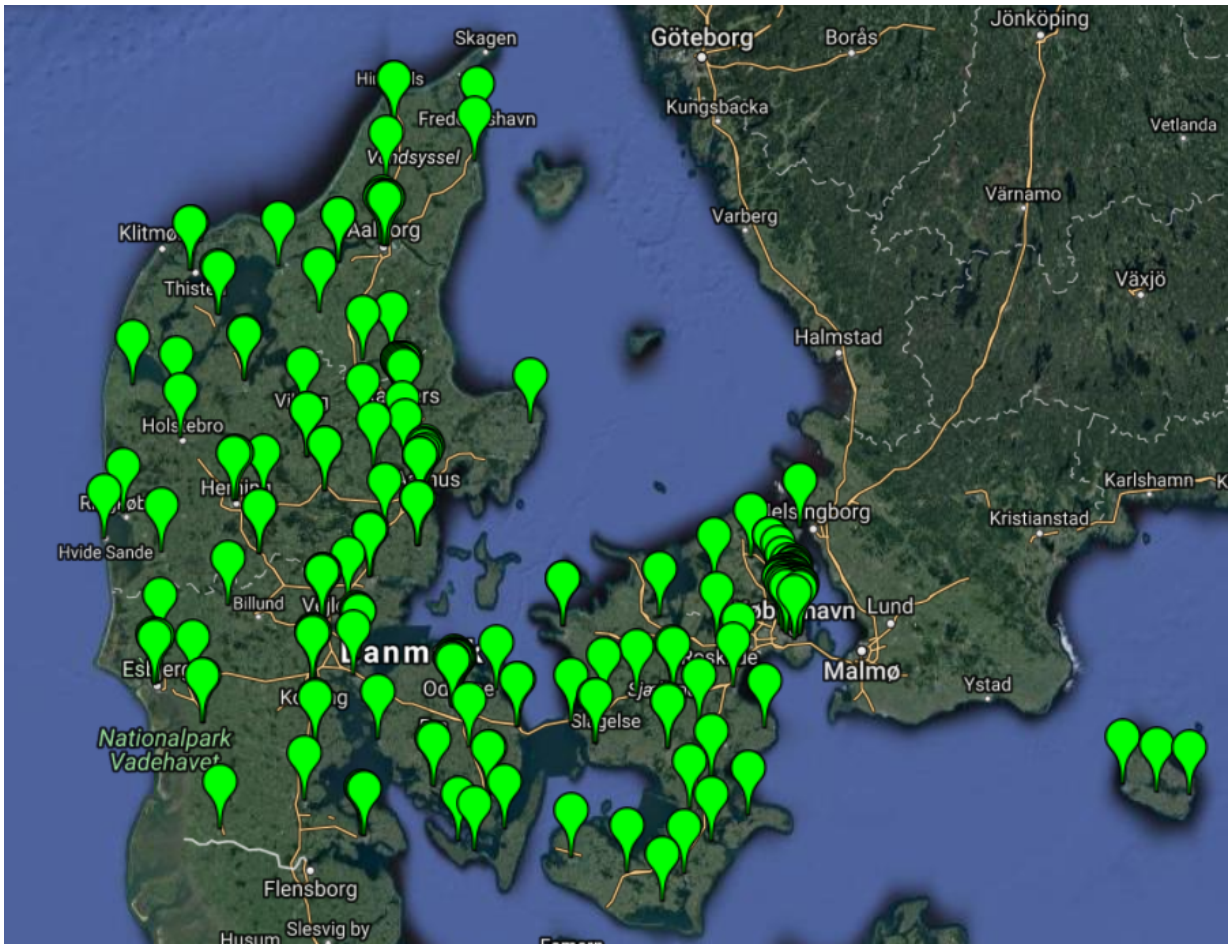

Figure B.6: All 227 city centres obtained from GeoDanmark

Figure B.7: The coastline in QGIS

B.5 Statistics of house prices

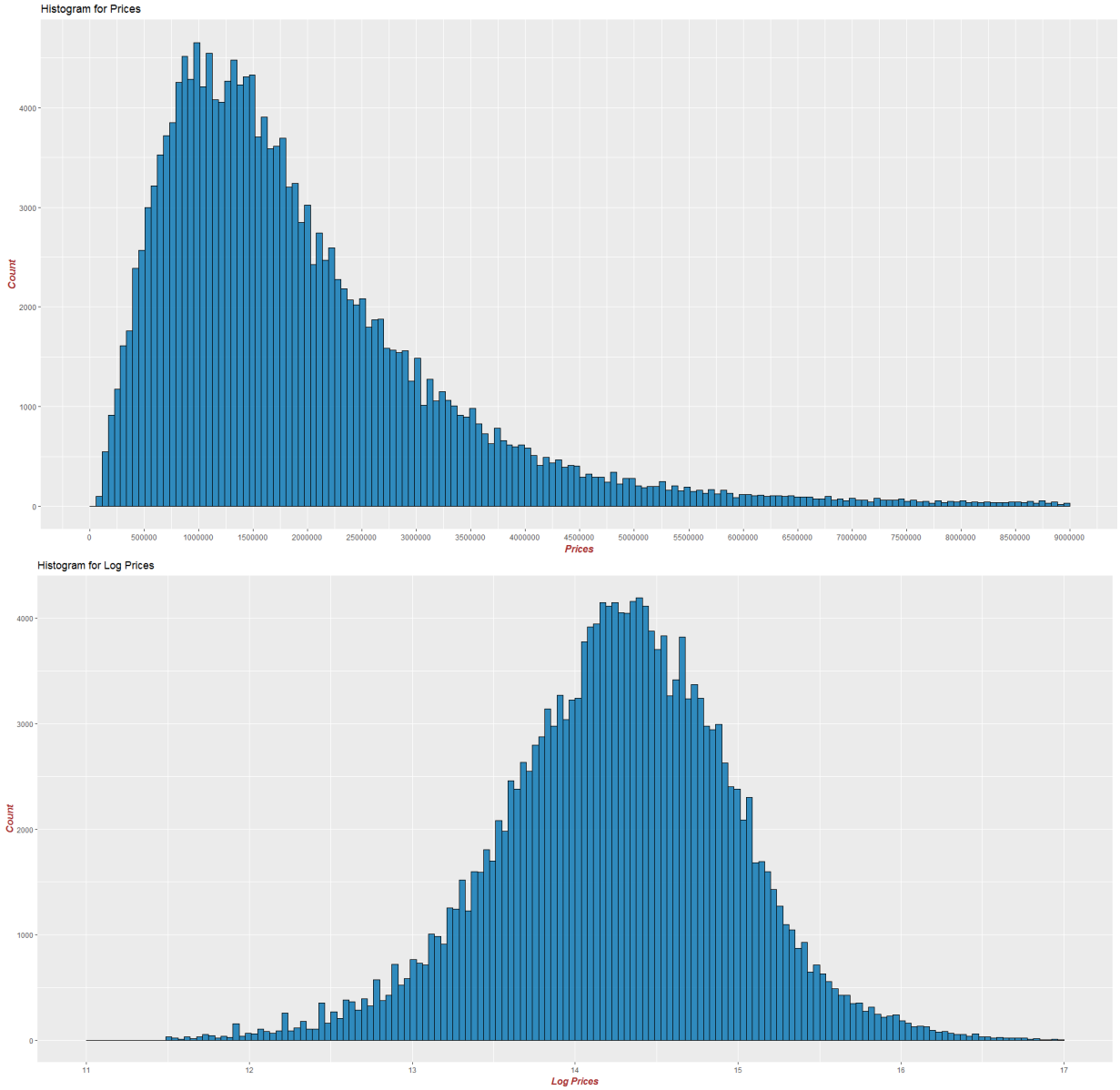

Figure B.8: Histogram of realised prices in Danish kroner

C Results and Tunings

In this appendix section, tuning figures using the `R` package `ggplot2` are presented.

C.1 Tunings

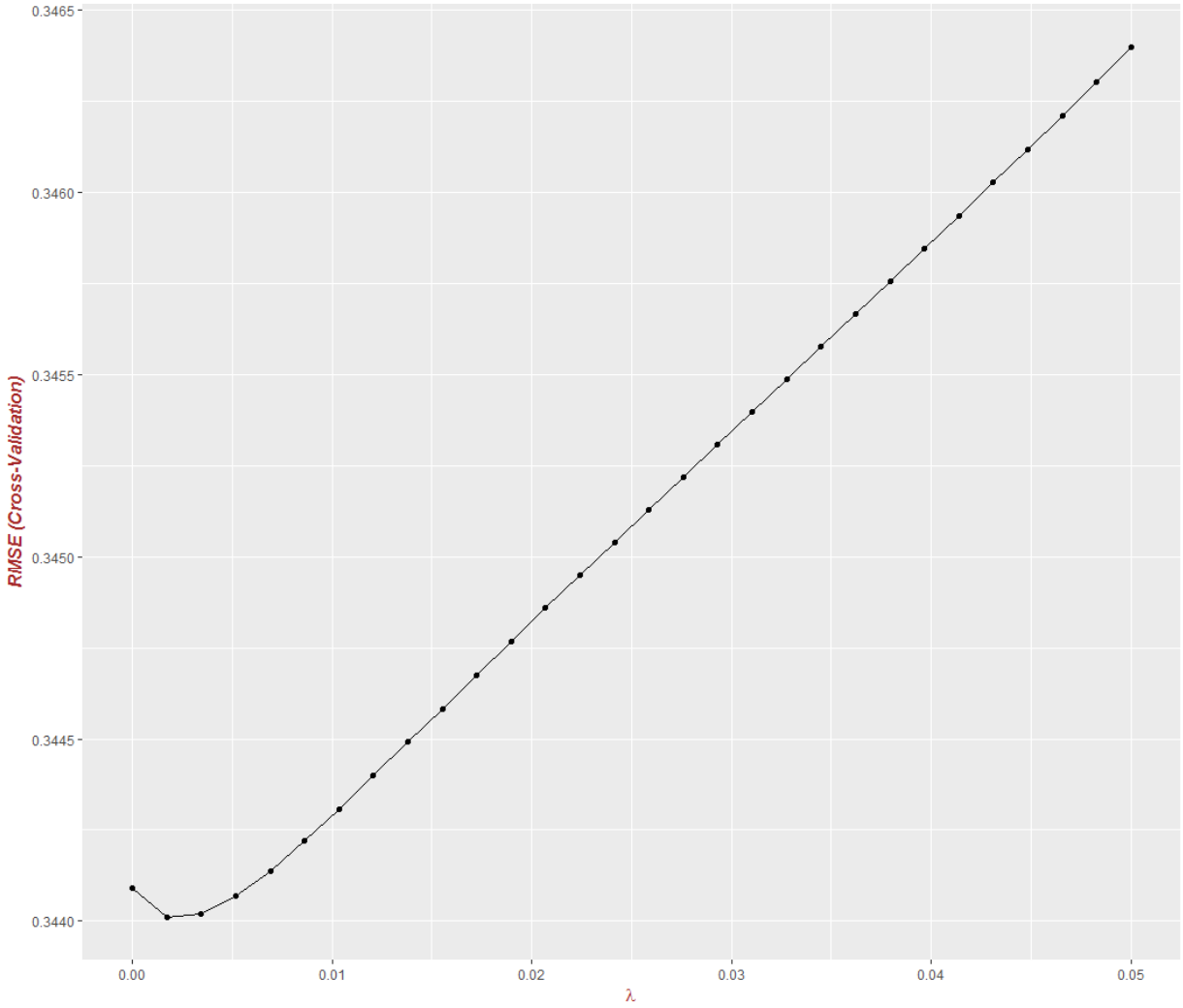

Figure C.1: Tuning of hyperparameters in ridge regression

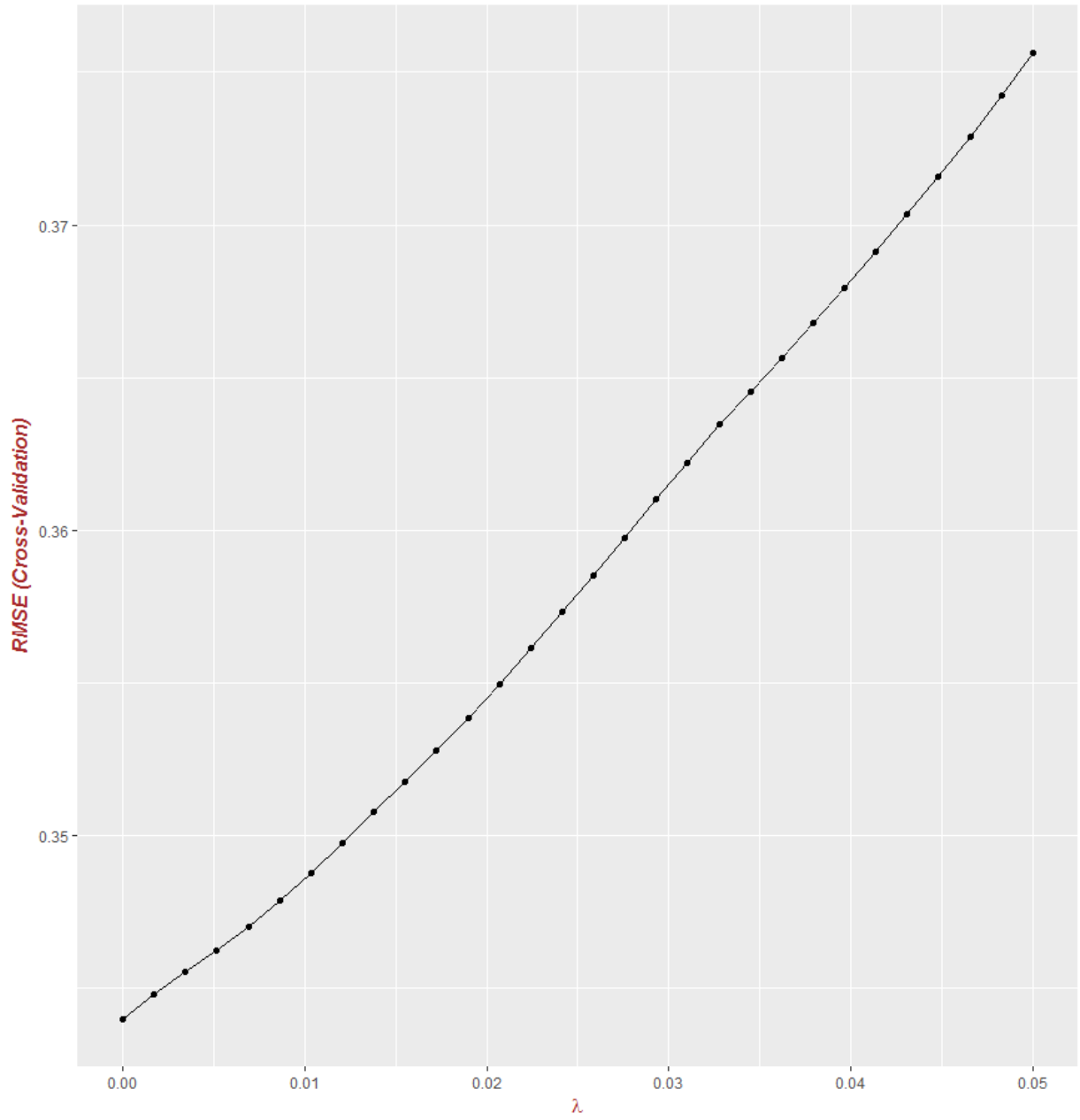

Figure C.2: Tuning of hyperparameters in LASSO regression

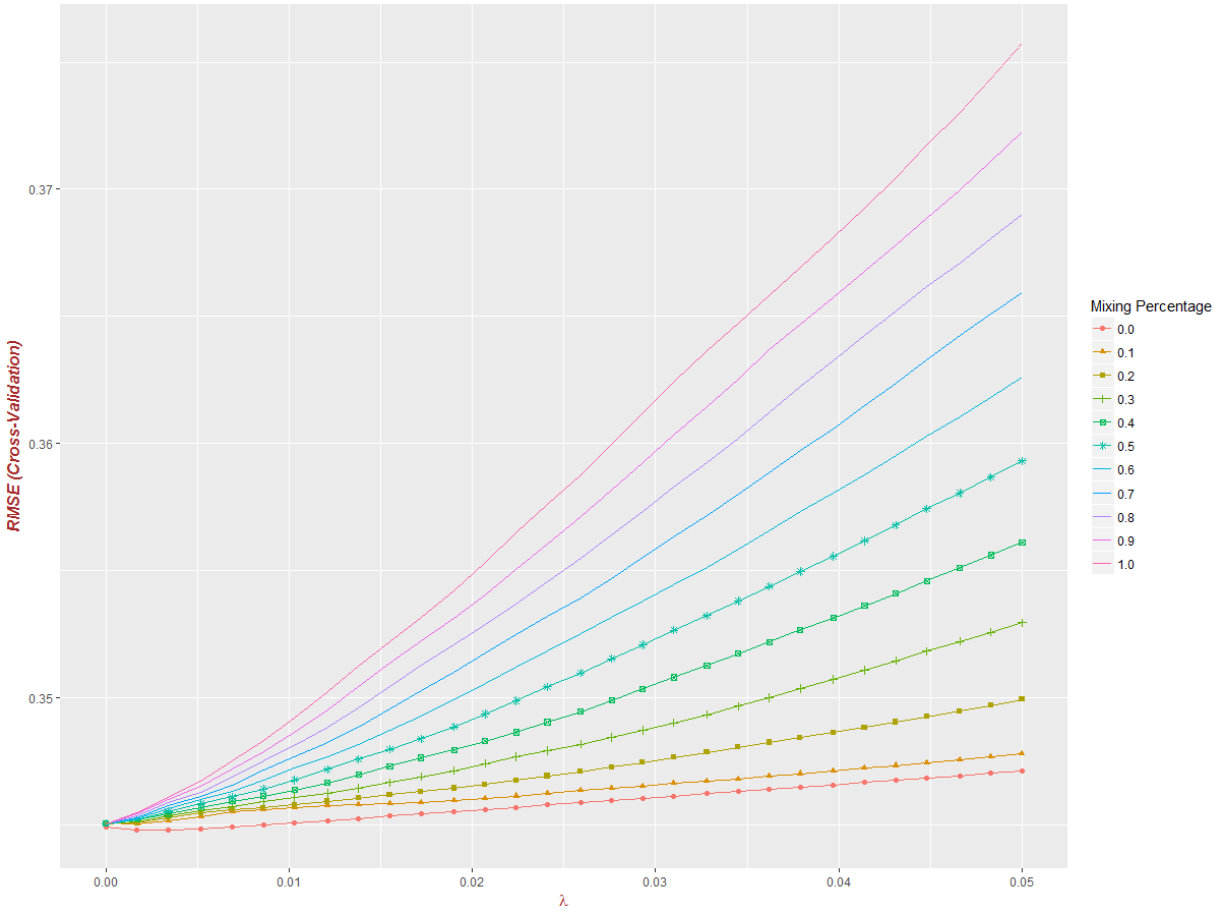

Figure C.3: Tuning of hyperparameters in elastic net regression

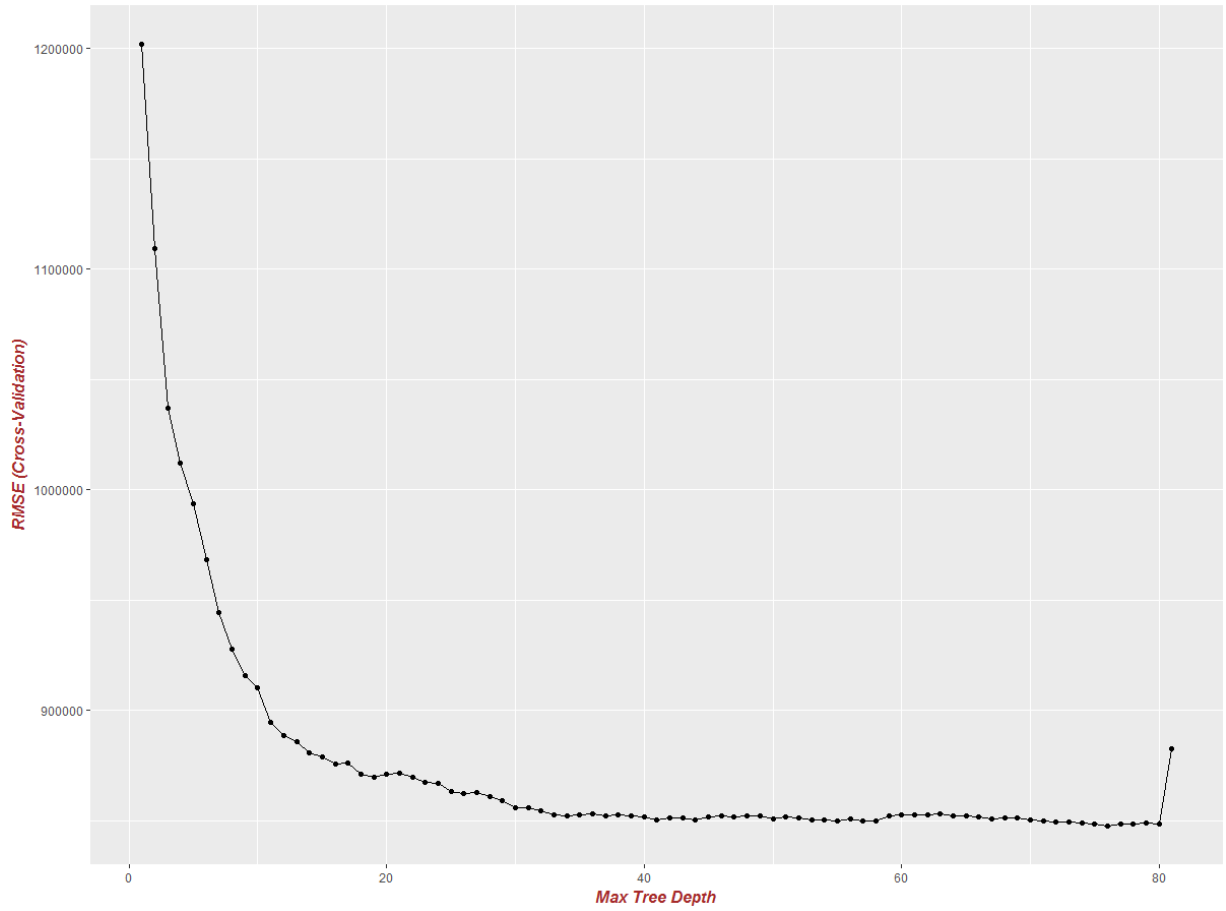

Figure C.4: Tuning of hyperparameters in CART regression

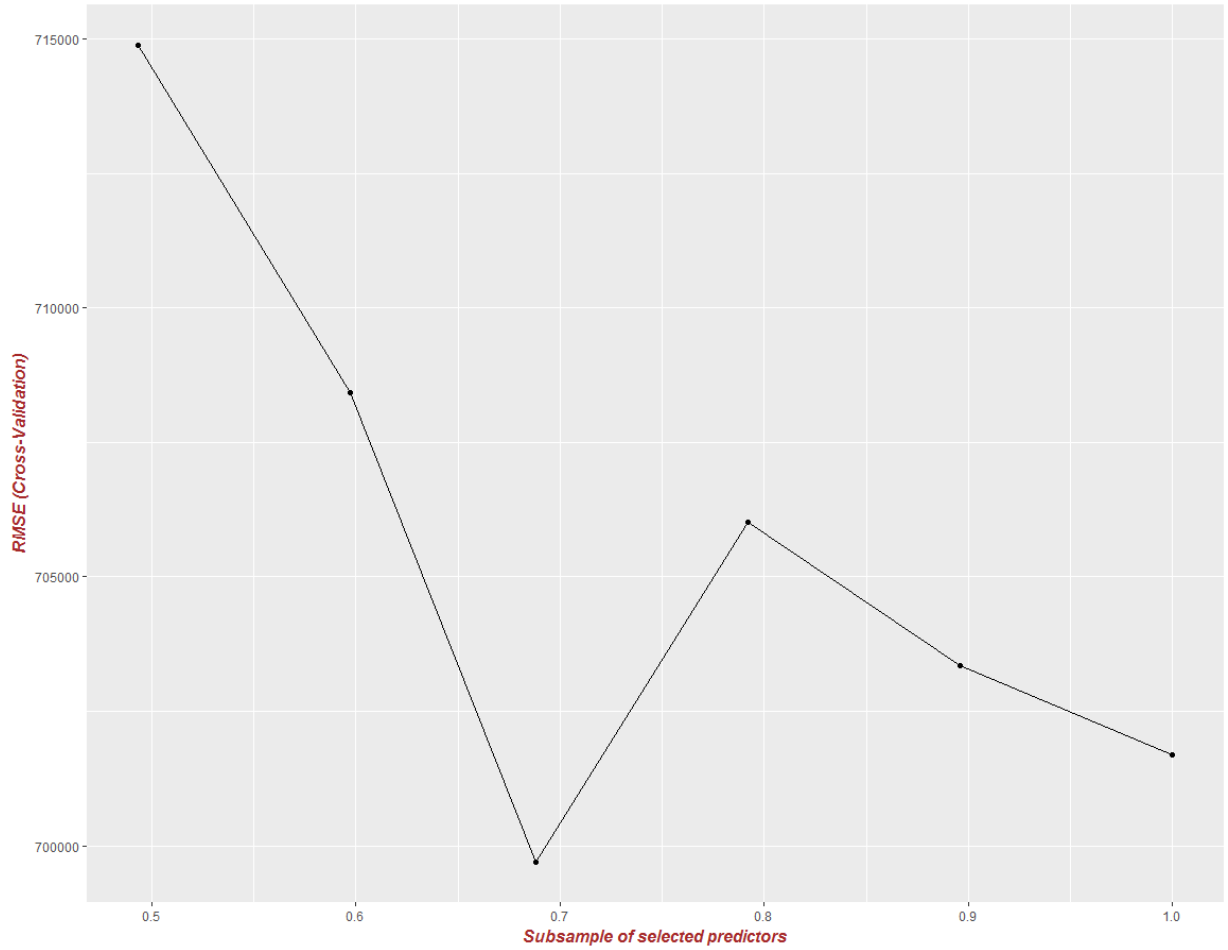

Figure C.5: Tuning of hyperparameters in random forest regression

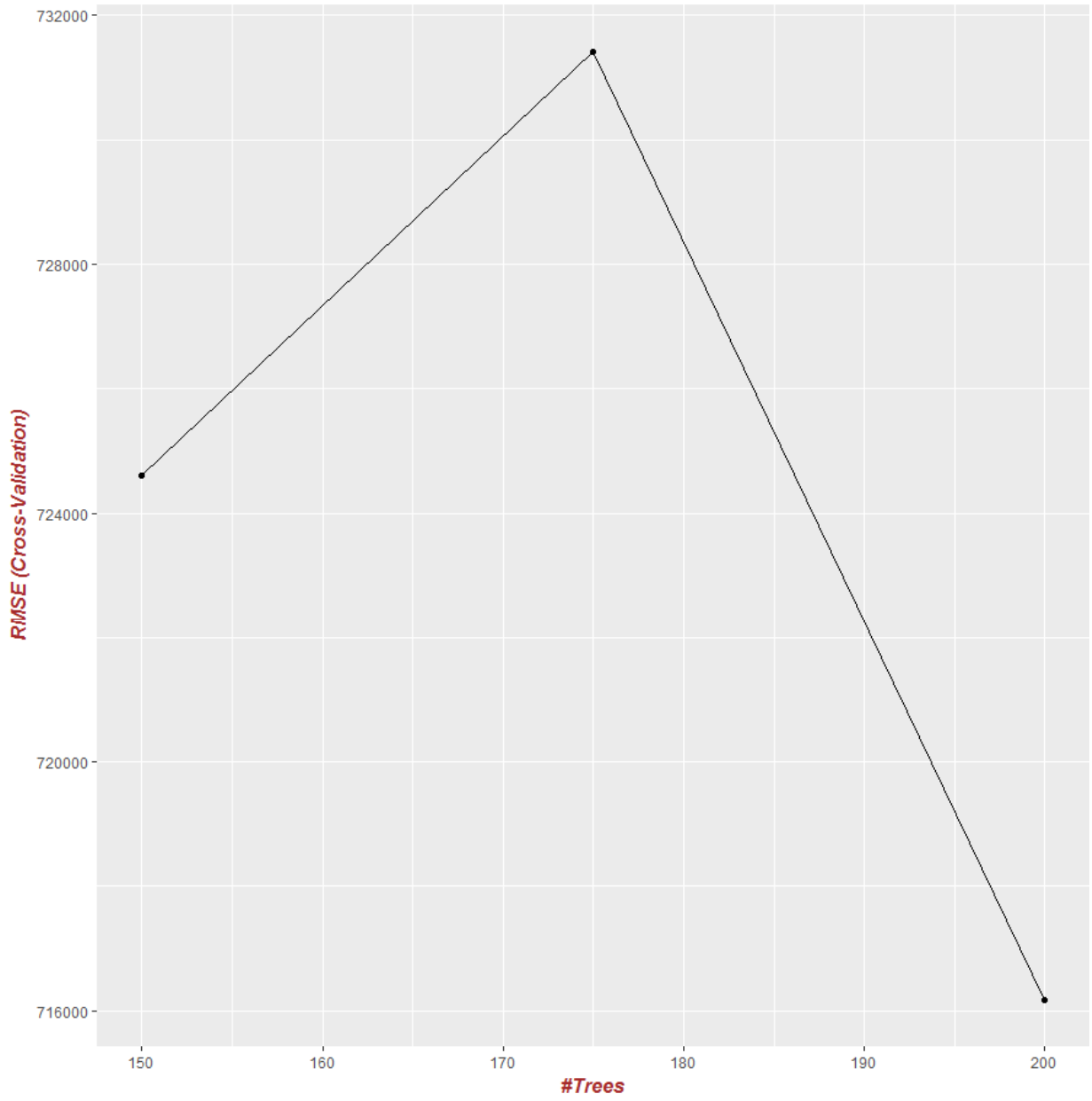

Figure C.6: Tuning of hyperparameters in BART regression

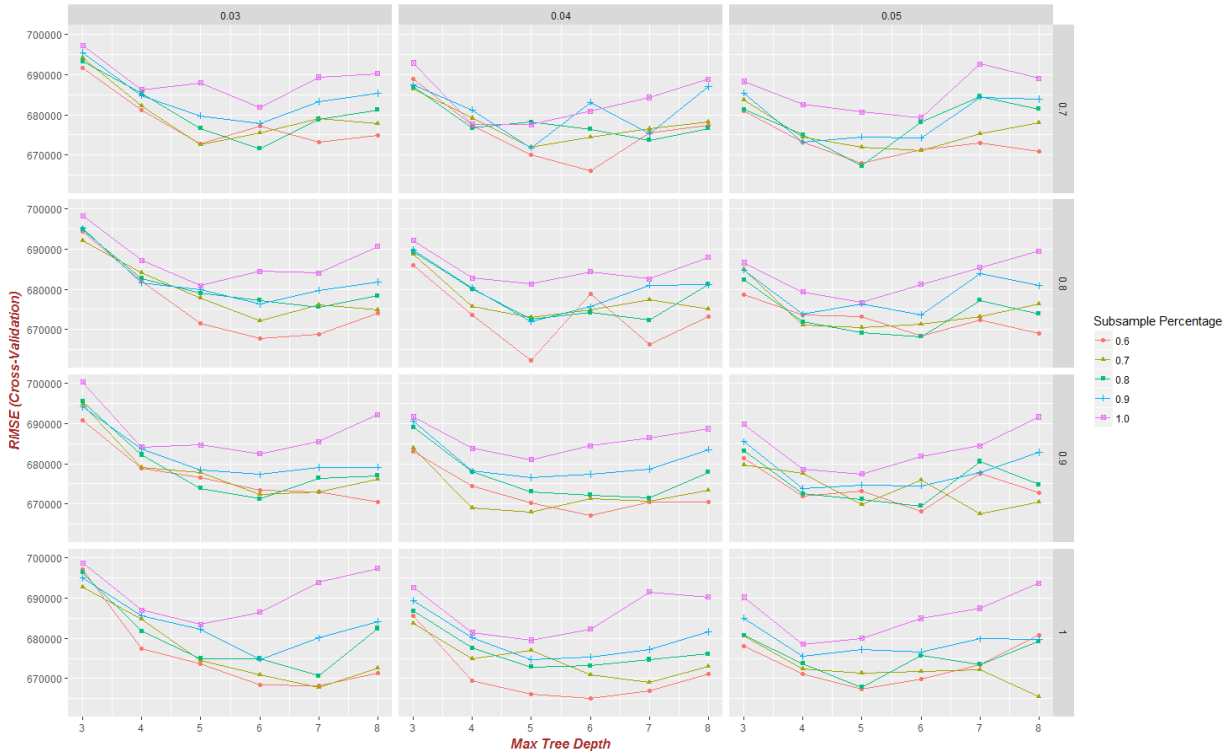

Figure C.7: Tuning of hyperparameters in XGboost regression

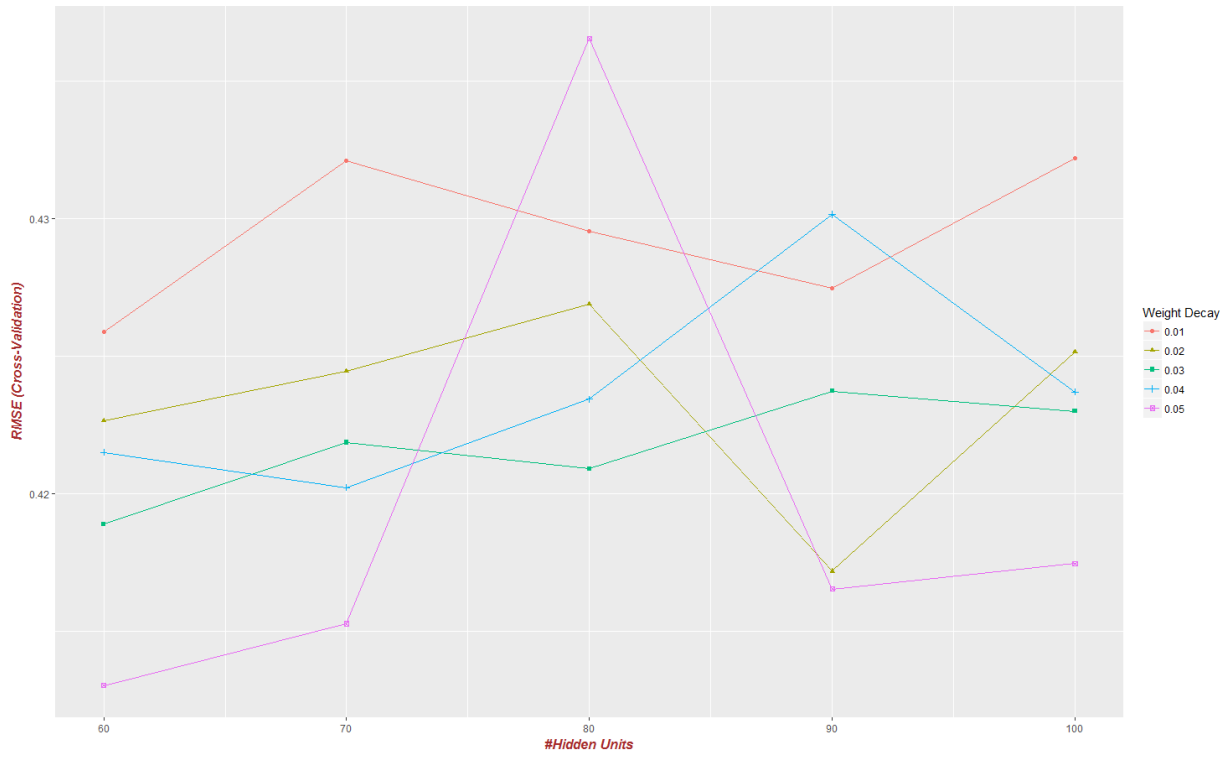

Figure C.8: Tuning of hyperparameters in neural network regression

C.2 Results and other graphics

	OLS	BagCART	BRNN	Neunet	RandomForest	XGboost	BART	Price	SKAT
OLS	1	0.953	0.843	0.951	0.948	0.96	0.948	0.909	0.915
BagCART	0.953	1	0.851	0.962	0.992	0.988	0.967	0.927	0.951
BRNN	0.843	0.851	1	0.857	0.85	0.857	0.846	0.815	0.826
Neunet	0.951	0.962	0.857	1	0.962	0.972	0.958	0.92	0.948
RandomForest	0.948	0.992	0.85	0.962	1	0.989	0.968	0.926	0.95
XGboost	0.96	0.988	0.857	0.972	0.989	1	0.979	0.937	0.959
BART	0.948	0.967	0.846	0.958	0.968	0.979	1	0.924	0.946
Price	0.909	0.927	0.815	0.92	0.926	0.937	0.924	1	0.914
SKAT	0.915	0.951	0.826	0.948	0.95	0.959	0.946	0.914	1

Figure C.9: Model correlations including realised sale price and SKAT's appraisals

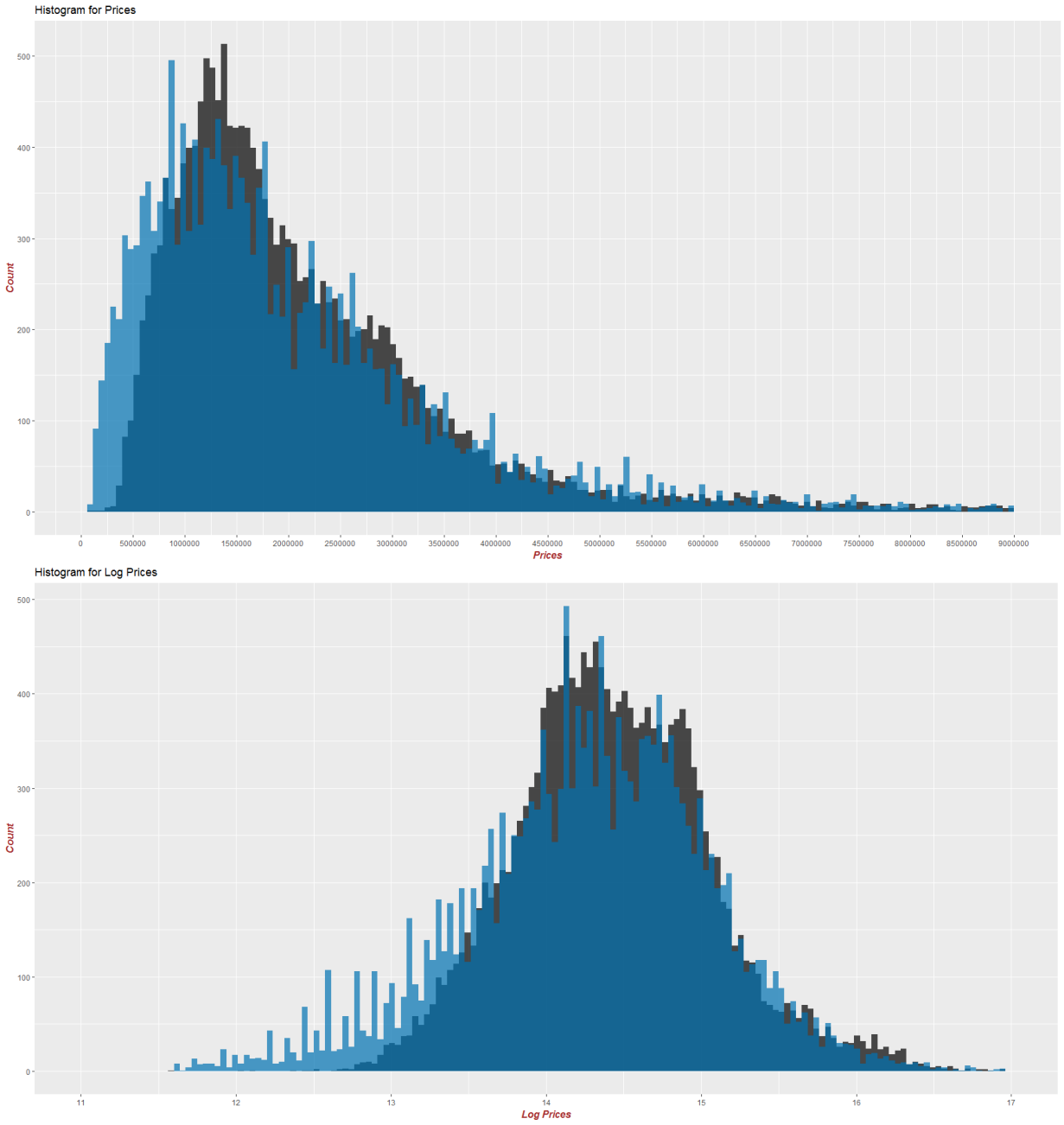

Figure C.10: Histogram of realised prices and SKAT's valuations in 2016

Note: The the blue fill is a histogram of SKAT's valuations in 2016, and the black fill is a histogram of the realised prices of the same houses in 2016.

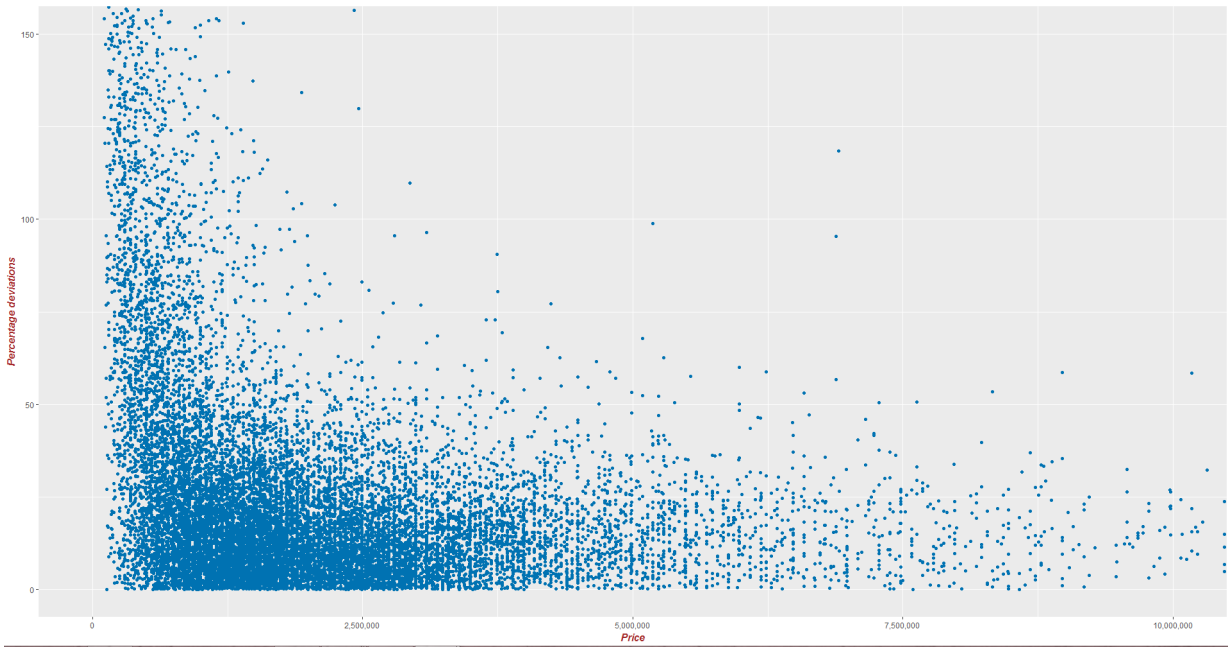

Figure C.11: Percentage deviations between predictions and realised prices

Note: This plot is only on a subset of the data to better visualise the point being made. It depicts absolute percentage deviations between the XGboost model and the observed prices.

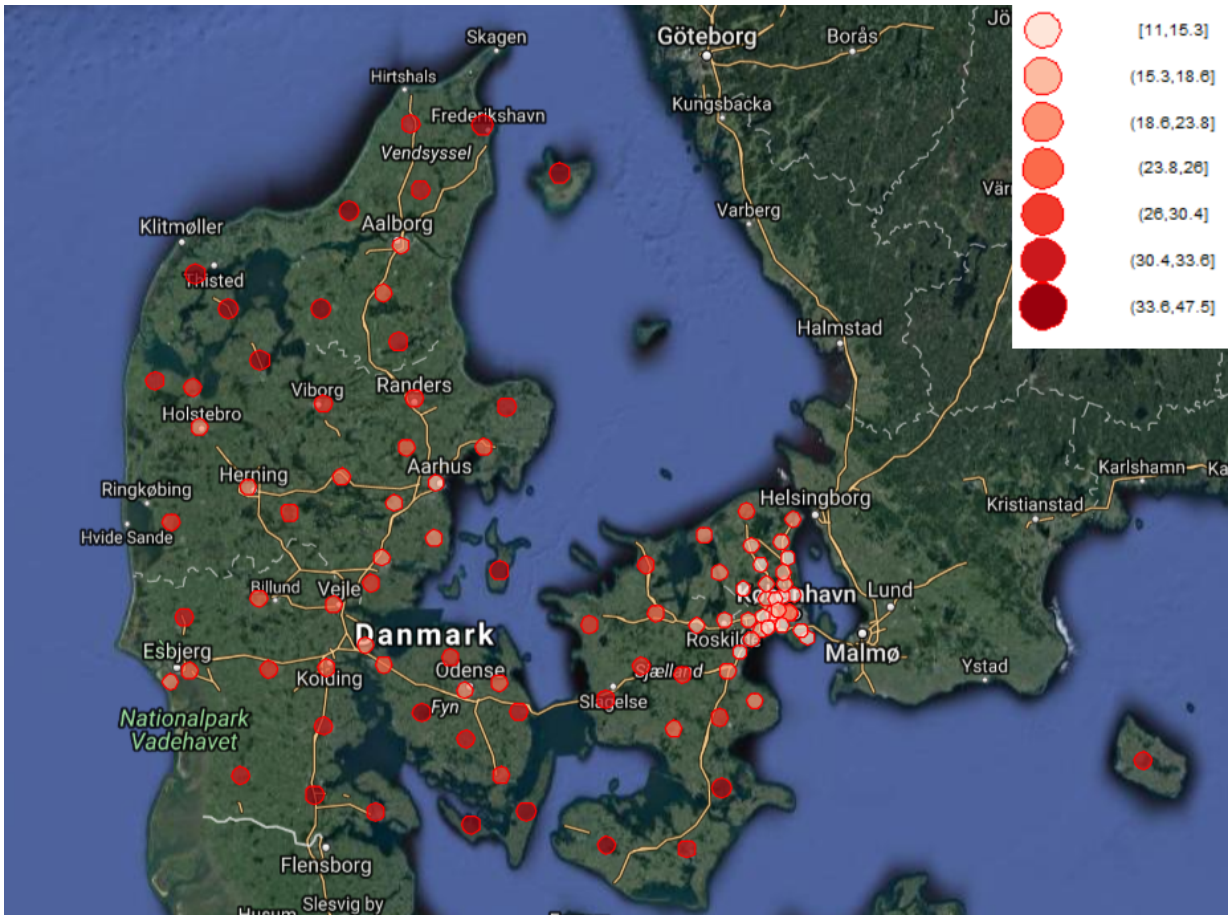

Figure C.12: Cross-country Coefficient of Dispersion

Note: The CoD measures appraisal uniformity and variability and is a relative measure of how much the value ratios differ from the median ratio (the greater the inconsistency in the value ratios).