University of Duisburg-Essen
Faculty of Business Administration and Economics
Chair of Econometrics

A Forecasting Study on Swedish Wine Sales

Statistical Learning

Term Paper

Submitted to the Faculty of
Business Administration and Economics
at the
University of Duisburg-Essen

from:

Jonathan Berrisch, Timo Rammert

Reviewer:	Dr. Thomas Deckers
Deadline:	Aug. 27th 2019

Name:	Jonathan Berrisch	Timo Rammert
Matriculation Number:	3071485	3030862
E-Mail:	Jonathan@Berrisch.biz	t.rammert03@gmail.com
Study Path:	M.Sc. Economics	M.Sc. Economics
Semester:	4^{th}	2^{nd}
Graduation (est.):	Summer Term 2020	Summer Term 2020

Contents

List of Figures **II**

List of Tables **II**

List of Abbreviations **II**

1 Introduction **1**

2 Data and Variables **1**

3 Validation Approach **3**

4 Analysis **3**

4.1 Mean Regression and Linear Regression 3

4.2 Least Absolute Shrinkage and Selection Operator 4

4.3 Principal Components Regression and Partial Least Squares 5

4.4 Splines 5

4.5 Decision Tree Methods 6

4.5.1 Overview 6

4.5.2 Single Regression Tree 6

4.5.3 Bootstrap Aggregation 8

4.5.4 Random Forest 10

4.5.5 Boosting 12

5 Robustness **14**

6 Conclusion **14**

References **16**

Software-References **19**

A Appendices **19**

A.1 Dataset 19

A.2 Additional Partial Dependence Plots 20

List of Figures

1 Share of Missing Values in the Wine Dataset. 2
2 Histogram and Estimated Density of the Litre Variable. 3
3 Relation of Coefficients and Shrinkage. 4
4 Principal Component One and Two. 5
5 RMSE Values of Different Spline Models 6
6 Example of a Regression Tree. 7
7 Example of a Pruned Tree. 8
8 Bagging: RMSEs at Different Tree Sizes 9
9 Bagging: Variable Importance. 9
10 RMSEs of the Random Forest for Different Parameters 10
11 Random Forest: Variable Importance. 11
12 Random Forest: Partial Dependence Plots. 12
13 RMSEs of the Boosting Model for Different Parameters 13
14 Boosting: Variable Importance Plot. 13
A1 Random Forest: Partial Dependence Plots. 20
A2 Boosting: Partial Dependence Plots. 20

List of Tables

1 Frequency of Variable Types. 2
2 RMSEs of every Fold for every Method. 15
A1 Summary Statistics of the Dataset. 19

List of Abbreviations

bagging Bootstrap Aggregation 8
lasso Least Absolute Shrinkage and Selection Operator 4
pcr Principal Components Regression 5
pls Partial Least Squares 5
RMSE Root Mean Squared Error 3

1 Introduction

This paper presents a forecasting study of Swedish wine sales. This study is based on the friberg gronqvist wine dataset, which is publicly available and was previously analyzed by Friberg and Grönqvist (2012). Recent technological advancements led to a significant increase in statistical methods that are computationally demanding while potentially outperforming classical statistical methods, especially in terms of forecasting. Those techniques are usually referred to as statistical or machine learning methods. The methods and procedures we introduce in this paper are based on 'An Introduction to Statistical Learning with Applications in R' (James, Witten, Hastie, & Tibshirani, 2014).

We present various models which are increasingly complex. Those models potentially gain forecasting power while losing interpretability. The applied models range from linear regressions to tree-based methods like random forests and boosting. The goal is to develop the best possible model to forecast weekly wine sales in litres in out of sample data.

Although the main goal is an accurate forecast, some of the methods used in this paper can also be used to quantify the impact of expert reviews on the sales, if there is any. The effect of expert opinions on consumer demand was analyzed in previous research. Notably, Hilger, Rafert, and Villas-Boas (2011) used a field experiment to study a review-based demand effect using wine score labels in a retail grocery chain. They find that providing information based on expert opinions increases the sales; this effect does depend on the score a wine has gotten. Ashenfelter and Jones (2013) analyze, wether expert reviews influence the prices of wine. They conclude that expert reviews contain public as well as private information. They only validate an influence on the highest rated wines. Furthermore, Bicknell and MacDonald (2012) delivered research concerning the market for wine from New Zealand. They validate a significant regional influence on wine prices. This influence is substantially lower for wine, which is designated for the export market.

The remainder of this paper is structured as follows. Chapter 2 gives an overview of the dataset and the preprocessing. Chapter 3 describes the validation approach. Chapter 4 presents the models and their specification. Chapter 5 presents the results of some robustness checks. Chapter 6 provides the results, and concludes.

2 Data and Variables

The dataset used in this paper contains 145179 observations with 59 variables. The variables include the name of the wine, the country of origin, its price, the amount sold per week in litres, the taste segment, and variables related to different

reviews, among others. A table with summary statistics of the dataset can be found in appendix A.1 in table A1.

We need to omit two variables beforehand, namely "*time_segm_price*" and "*artikpr*", because those are combinations of other existing variables. Hence inclusion would lead to the problem of multicollinearity. After omitting those two, we have left the following types of variables (see table 1). As the analysis is based on data from Sweden, all the levels of the factor variables are denoted in Swedish.

Table 1: Frequency of Variable Types.

Date	factor	logical	numeric
1	8	19	29

Figure 1 depicts the number of missing values in each variable. The number of complete cases would be zero without further selection. Therefore we exclude every variable with a ratio of missing values that exceeds 50%. In consequence, 41416 observations with 45 variables are used for building the forecasting models.

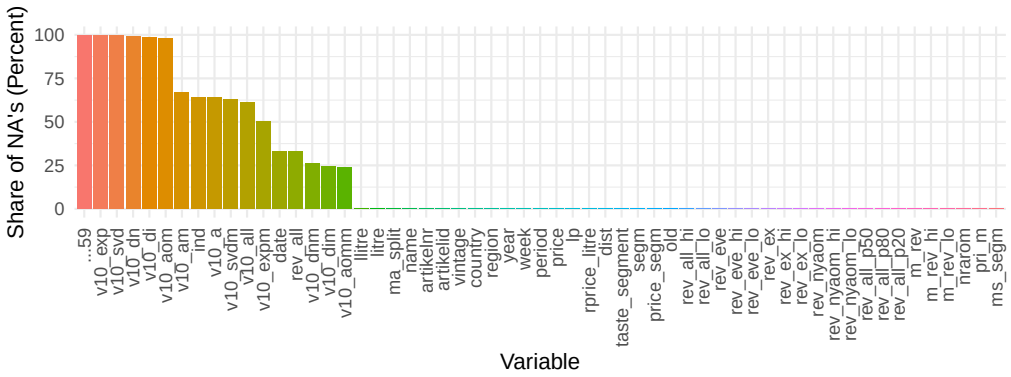

Figure 1: Share of Missing Values in the Wine Dataset.

Figure 2 depicts the distribution of the variable "*litre*", which is the dependent variable in our models. One can see that the observations are heavily skewed. The maximum is at 184200 litres sold weekly while its minimum is near zero with only 0.75 litres sold per week.

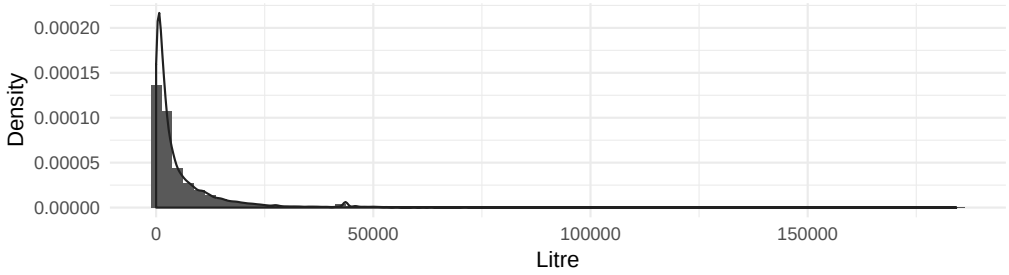

Figure 2: Histogram and Estimated Density of the Litre Variable.

3 Validation Approach

Sampling may influence the model selection process because a sample could be in favor of one specific method while another sample could lead to different results. Therefore we validate the results of each method using 5-fold cross-validation. This reduces the influence of sampling when building the training and the test set while it is computationally feasible.

In order to compute a prediction, the test set must include, at least, all features of the training set. Due to the high number of levels in the countries and name variable, this was not always the case. Therefore, we are using only the intersection of training and test features to be included in the estimation.

4 Analysis

4.1 Mean Regression and Linear Regression

At first, a mean regression and a linear regression are calculated. Those models are representing the baseline. For comparison of the different models, the Root Mean Squared Error (RMSE)

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

is calculated.

The mean regression yields an average RMSE of $\approx$ 13340 litres sold per week. The linear regression where all variables are included yields an average RMSE of $\approx$ 5572. The latter result is probably influenced by overfitting. To cope with this problem, we are using variable selection and dimension reduction methods, which are discussed in the proceeding sections.

4.2 Least Absolute Shrinkage and Selection Operator

The Least Absolute Shrinkage and Selection Operator (lasso) is a method combining linear regression and shrinkage of the coefficient estimates to do variable selection. It fits a linear model that is constrained by a penalty term, i.e., the lasso coefficients minimize:

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p}|\beta_j|.$$

Figure 3 depicts the relation between lambda and the coefficients. The cross-validated log lambda, which minimizes the test error is ≈ -0.16. At that level a total of ≈ 666 out of 713 coefficients are nonzero[1]. The other coefficients are precisely zero due to the sparsity property of the lasso estimator (Cf. James et al., 2014, p. 219.). Furthermore, the plot includes abbreviated variable names of the ten biggest coefficients. All of those coefficients are dummies for specific wine names. The average RMSE of the lasso approach is ≈ 5547.76. This is a slight improvement compared to the linear baseline model. The lasso slightly reduces the feature set while also reducing the RMSE. This is evidence supporting our prior assumption that the linear model was exposed to the problem of overfitting. Furthermore, the coefficients of the lasso model indicate that the name variable is the most important variable to explain the litres sold per week. It is followed by the vintage, the region, the taste segment, and the country of the wine. The least important variables are the review variables which are not present in the 400 biggest coefficients.

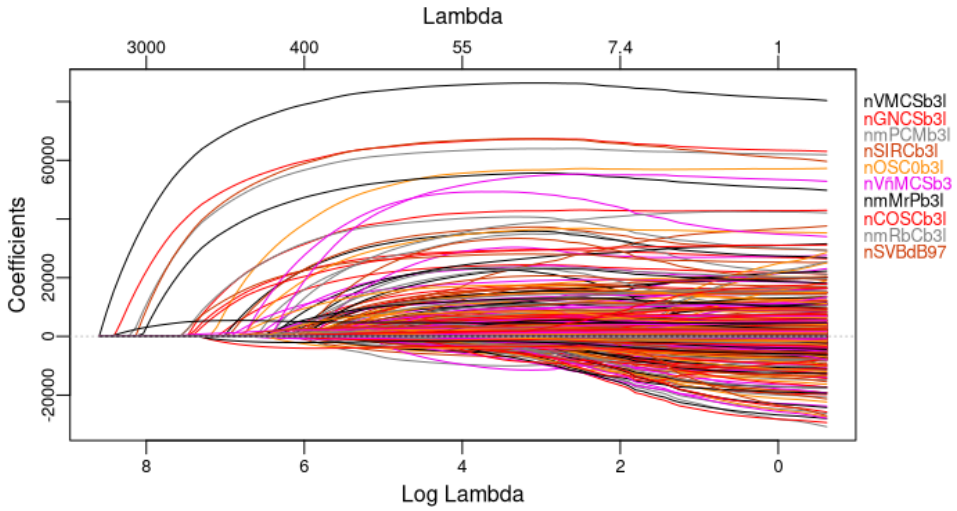

Figure 3: Relation of Coefficients and Shrinkage.

[1] ≈ 666.6 to be exact, but we could not resist to round down.

4.3 Principal Components Regression and Partial Least Squares

While lasso is a technique for variable selection, Principal Components Regression (pcr) and Partial Least Squares (pls) are other methods that reduce the dimension of the feature set by modifying the features. Thus, the least-squares estimation is performed using a transformation of the explanatory variables. Thereby, pls includes the dependent variable when computing the new feature set while pcr builds the principal components independent of the dependent variable (Cf. James et al., 2014, p. 219.).

Figure 4 shows the first two principal components from a pcr. The first two principal components represent roughly 3% of the total variance. This is a sign that the pcr does not work well on our data, which might be due to a large number of dummy variables.

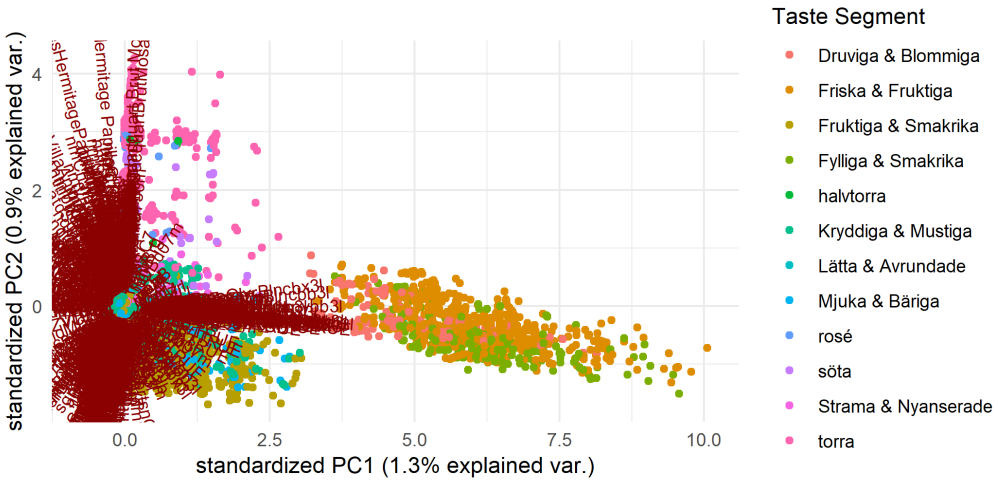

Figure 4: Principal Component One and Two.

The results confirm the expectation. pcr has a cross-validated RMSE of $\approx$ 5546.05 while pls leads to an RMSE of $\approx$ 6201.15. While $\approx$ 635 principal component where included in the pcr, only $\approx$ 35.2 components where used in the pls. Thus pls significantly reduced the dimension of the featureset. While sacrificing interpretability of the results, it leads to a higher RMSE.

4.4 Splines

The previous methods assumed linear relationships between the independent variables and the dependent variable. Considering the number of dummy variables in the feature set, the usage of nonlinear methods is quite limited. However, to recognize potential nonlinear relationships we add natural splines to the variables *year*, *price*, and *rprice_litre* (Cf. James et al., 2014, p. 271.). Those variables indicate the year in which the wine was distributed, the price and the real price

with Jan 2004 as base respectively. Each natural spline was allowed to have up to 20 knots, which is sufficient to estimate complex linear relationships. The respective RMSEs depending on the knots are presented in figure 5.

Splines can improve the prediction a little, which indicates that at least some nonlinear relationship is present. However, like the methods before, splines reduce the interpretability of the coefficients. In this case, this tradeoff is not offset enough by the gain in precision.

Furthermore, figure 5 shows, that the RMSEs depend to some extent on the fold which was used for cross-validation. This approves our theoretically founded motivation to use cross-validation.

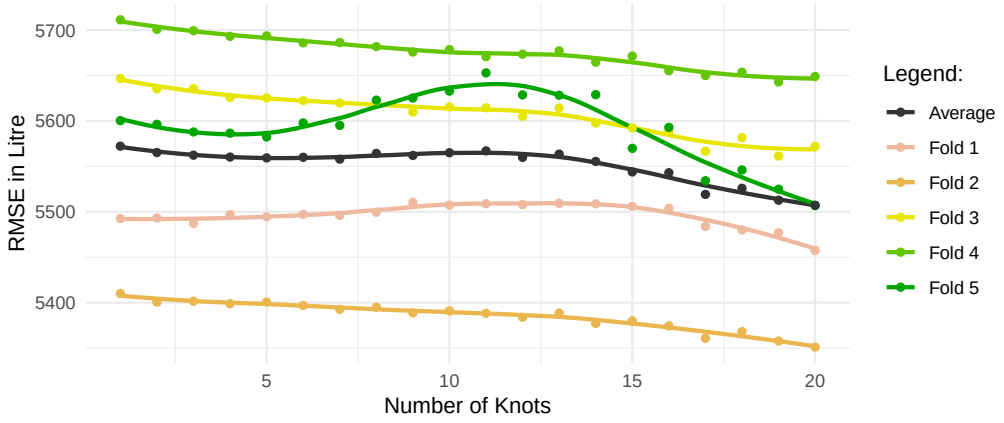

Figure 5: Regression with Splines: Dependency between Knots and RMSE.

4.5 Decision Tree Methods

4.5.1 Overview

Tree-based methods utilize decision rules to split the feature space into several different regions. 'Since the set of splitting rules used to segment the predictor space can be summarized in a tree, these type of approaches are known as decision tree methods' (James et al., 2014, p. 303). Simple tree-based methods use single regression trees. One can combine different regression trees for a single prediction this usually improves the predictive power, but the interpretability becomes more complex. Those approaches, namely bagging, random forests, and boosting, are described in more detail when being applied.

4.5.2 Single Regression Tree

For growing a regression tree, 'the algorithm needs to automatically decide on the splitting variables and split points [...] the tree should have' (Hastie, Tibshirani, & Friedman, 2013, p. 307). The tree is grown when a split is found that minimizes

the sum of squared residuals. The splitting is done with a greedy algorithm, i.e., at first, all the data is split into just two groups while the search of the splitting variable and split point includes all possible variables and points (Cf. Hastie et al., 2013, p. 307.). After the data is split this process is repeated for the obtained two split regions until the tree is large enough that the terminal nodes reach a pre-defined minimum size.

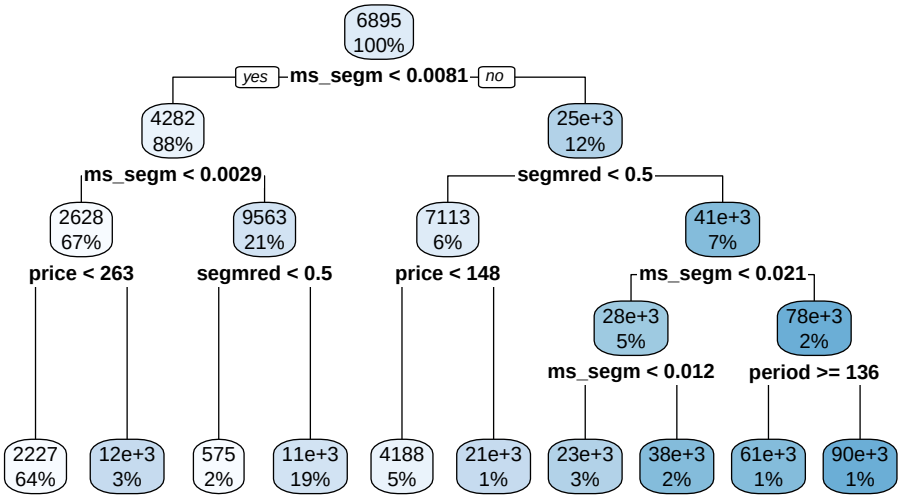

Figure 6: Example of a Regression Tree.

Figure 6 shows that only a few of the available variables are used. These are namely "*ms_segm*", "*segmred*", "*price*", and "*period*" and split the dataset into 10 parts, i.e., the tree has got 10 terminal nodes. The figure does also tell us what share of the data can be sorted to each node, as well as the mean price of the allocated observations. The regression trees build in the other four folds of the cross-validation are almost the same as the shown tree. The selected variables indicate that according to this method, the market share, the color of the wine, the price as well as the period are the most important variables. Concerning the graph and the variable "*segmred*" it is important to keep in mind that this is a dummy variable meaning that a value < 0.5 (the left track) stands for a wine that is not red. Conversely, the right track stands for red wines.

As regression trees may tend to grow quite large, a possible way to get smaller trees is pruning back the grown larger tree. The goal of pruning is 'to select a subtree that leads to the lowest test error rate [which] we can estimate [...] using cross-validation or the validation set approach' (James et al., 2014, p. 308). This is done by *cost complexity pruning*.

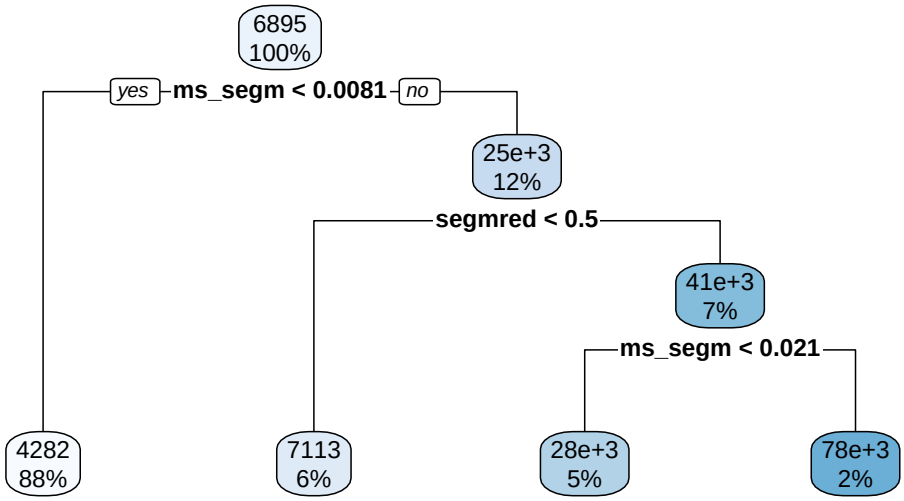

Figure 7: Example of a Pruned Tree.

The pruned tree (see Figure 7) does only utilize the dummy whether the wine is red wine and the mean market share within color during weeks the wine is distributed. "*ms_segm*" is used twice; thus, the pruned tree splits the dataset into four terminal nodes. While the pruned tree thus might be easier to interpret, it does also come with a higher RMSE. The mean cross-validated RMSE of the pruned trees is ca. 7935.64, while the mean RMSE of the conventional regression trees is about 6494.75.

4.5.3 Bootstrap Aggregation

Bootstrap Aggregation (bagging) is a general method of repeatedly taking bootstrap samples from the dataset, estimating a model for every sample and averaging the predictions of every model afterward. The bagging algorithm can also be used to improve decision tree models. For this, a bootstrap sample is drawn before growing the tree. This process is repeated several times to grow multiple trees. Finally, the predictions of the trees are averaged. By bootstrapping bagging brings down the variance and thus, addresses the main problem of single regression trees which are very dependent on the actual sample.

Figure 8 shows the RMSE values of different bagging models. The bootstrap sample size was chosen to be two-thirds of the total number of observations. One can see that even small models with down to 15 trees significantly outperform the single tree model of the previous chapter.

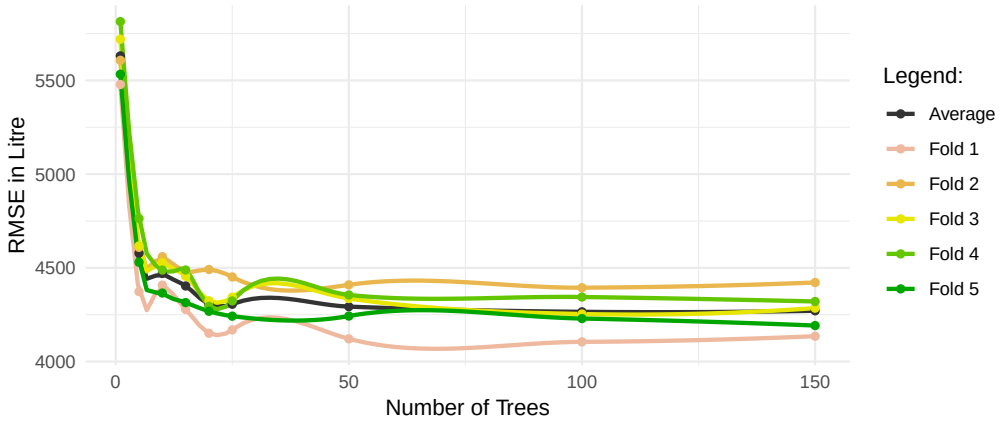

Figure 8: Bagging: RMSEs at Different Tree Sizes (Smoothed).

To further analyze the bagging model, we estimated a bagging model with 25 trees for the whole dataset. Figure 9 presents the variable importance this model. It presents those 20 variables which would increase the RMSE the most when being omitted. The plot shows that the influence of "ms_segm" which is the market share of the wine as well as "segmred" which is a dummy, indicating if the wine is red, substantially contribute to our model. Intuitively, the price also influences how many litres per week are sold. More surprisingly, however, is the fact, that also the date substantially contributes to our mode. This indicates that prices have changed over time.

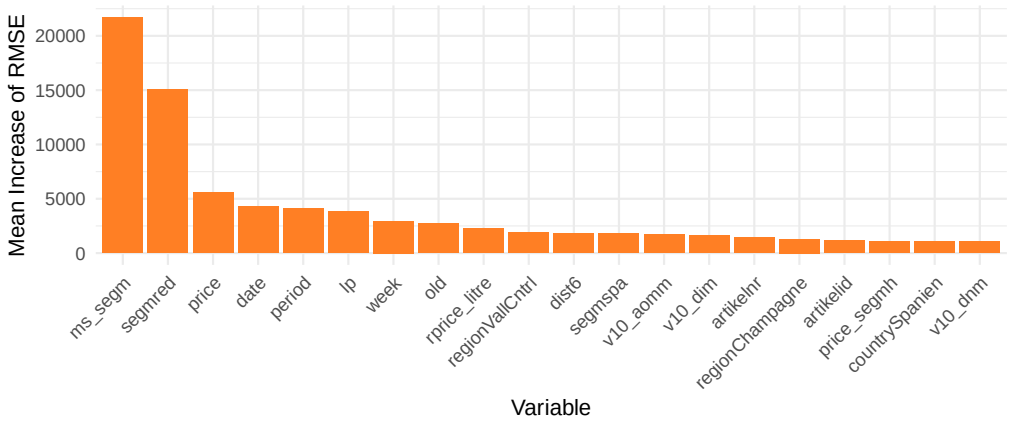

Figure 9: Bagging: Variable Importance.

Bagging, in general, considers all possible variables at each split. This makes bagging computationally demanding. Moreover, this may lead to similar trees at each iteration. Thus, the trees are probably highly correlated.

4.5.4 Random Forest

The random forest algorithm is a special case of bagging. The name random forest arises from the fact that in contrast to the usual bagging algorithm, at each split of the tree building process, only a pre-specified number of randomly chosen variables are taken into consideration. This results in a different tree for each iteration of the random forest and therefore decorrelates the trees. In consequence, the random forest reduces the overall variance in comparison to bagging. This effect is especially valid for homogenous datasets in which the trees of the bagging algorithm are highly correlated.

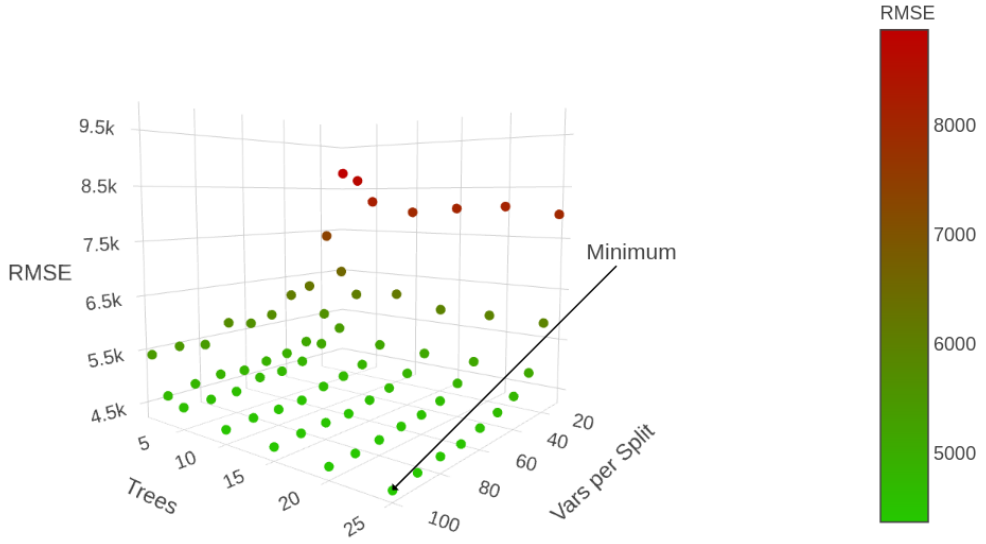

Figure 10: Random Forest: Dependency between RMSE, the Number of Trees and the Number of Variables Included at each Split.

We calculated RMSE values for a range of Random Forest models to evaluate which combination of the number of trees to grow, and the number of variables taken into account at each split leads to the best results. The results are plotted in Figure 10. The smallest RMSE is ≈ 4374. Moreover, the plot shows, reveals that tiny models profit substantially from increasing complexity while bigger models profit only by a small margin. The model with 20 trees and 80 variables at each split still has a RMSE of ≈ 4456, which is an increase of less than 1.9%.

The bagging model with 25 trees (RMSE of ≈ 4305) outperforms the random forest model (with 25 trees and 100 variables) only by a small margin (RMSE of 4374.330). However, training the random forest demands only 15% of the time, that the bagging model needs.

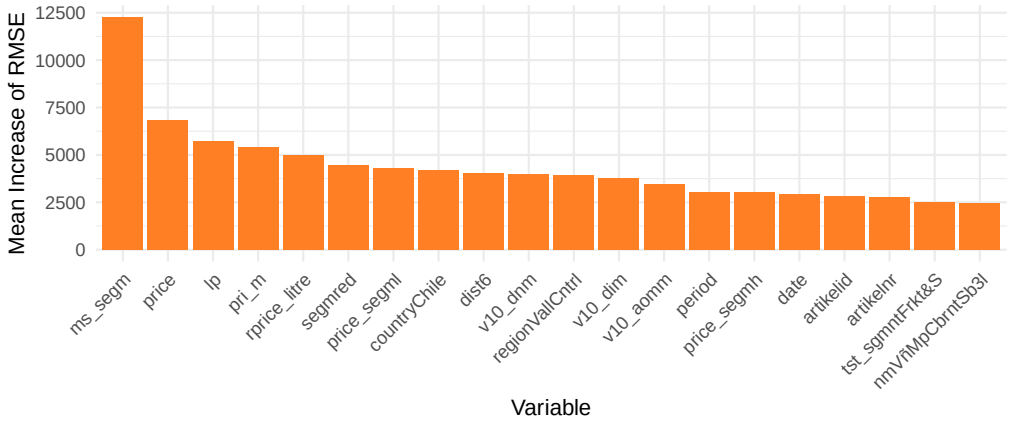

Figure 11: Random Forest: Variable Importance.

Figure 11 shows the variable importance plot for the 20 most important variables of the "optimal" random forest, which minimizes the RMSE in the given range. According to this plot the most important variable, in terms of the mean increase of the RMSE, is "*ms_segm*", followed by four variables with regard to the price: "*price*", "*lp*", the logarithmized price, "*pri_m*", the mean real litre price (base Jan 2004) during the weeks the wine is distributed and "*rprice_litre*", real litre price (base Jan 2004). Thus, the litres of wine sold per week is mainly determined by its market share and its price. Although, there are many more variables which, according to the random forest model, are important. One should note that those variables do not necessarily have the highest marginal effects.

Figure 12 shows the partial dependence plots for the 10 most important variables. The plots present the marginal effect of the variable on the litres sold per week. Those plots reveal some nonlinear dependencies in the data which explains the poor performances of the linear models. Furthermore, the figure validates some intuitive assumptions. For example, a higher market share leads to more litres sold, while high real prices lead to fewer litres sold. Besides, the influence of "*v10_dim*", which is the "Dagens Industri" review, is especially large, when the review is excellent. The complete partial dependence plot with all 20 most important variables can be found in the appendix A.2 in figure A1 to keep this chapter at a reasonable size.

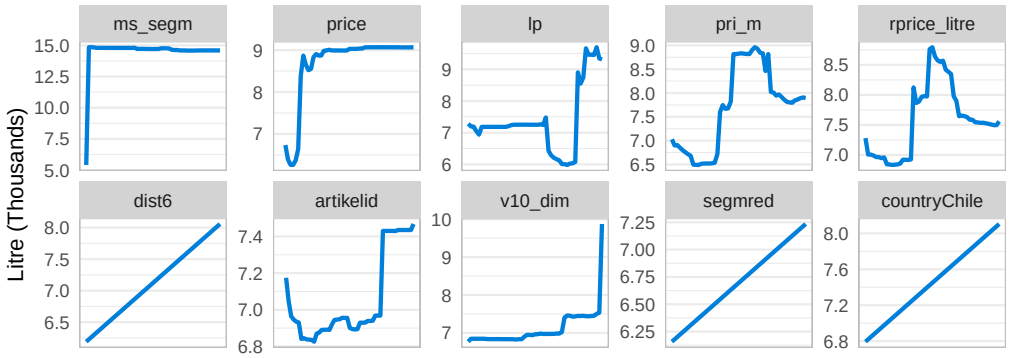

Figure 12: Random Forest: Partial Dependence Plots.

4.5.5 Boosting

In contrast to random forests boosting grows tree sequentially meaning that for every new tree information from already grown trees is used (Cf. James et al., 2014, p. 322.). For boosting rather small regression trees are used to improve the boosting tree slowly. This is achieved by taking the residuals from the current boosting tree, fitting a (small) tree to these residuals and then fitting this tree into the boosting tree in areas where the performance of the boosting tree could be improved (Cf. James et al., 2014, p. 322.). Thus the existing tree is expanded with the new tree. This means that, in contrast to bagging, the newly grown trees do depend on the earlier grown trees.

The boosting models possess three parameters for tuning: The number of trees, the interaction depth, and the shrinkage parameter λ. The resulting RMSEs for different combinations of those tuning parameters are shown in figure 13. Another tuning parameter implemented in the function for running boosting in R, *gbm()*, is the *bag.fraction*. This argument defines which fraction of the data is used to build the next tree. The default value is 0.5, which means that 50% of the training observations are samples (without replacement) each time a new tree is estimated. This adds randomness to the estimation, which reduces the problem of overfitting, but it also adds variance to the in-sample predictions (Cf. Elith, Leathwick, & Hastie, 2008, p. 806.). We set this value to 1. This outperformed the default setting of 0.5 by a $\approx 1.9\%$ decrease in out of sample cross-validated RMSE. Hence all observations in the training set have been used at every iteration.

According to our calculations, the optimal boosting model is the one with 25 trees used, an interaction depth of 15 and a shrinkage parameter of 0.4.

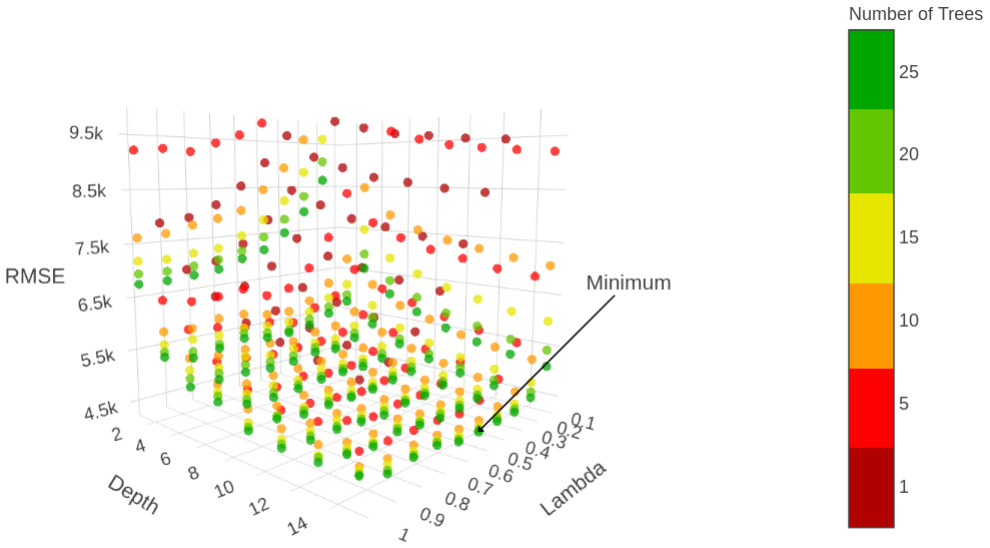

Figure 13: Boosting: Dependency between RMSE, Lambda, the Depths and the Number of Trees that are Grown.

Figure 14 shows the importance of the 20 most important variables according to the optimal boosting model. The values shown are the relative importances in percent. The most important variable is "*ms_segm*", while the second most important variable is "*segmred*", followed by "*period*" and "*price*". This also does confirm the results of the single tree as well as the pruned tree, as those utilized exactly those variables.

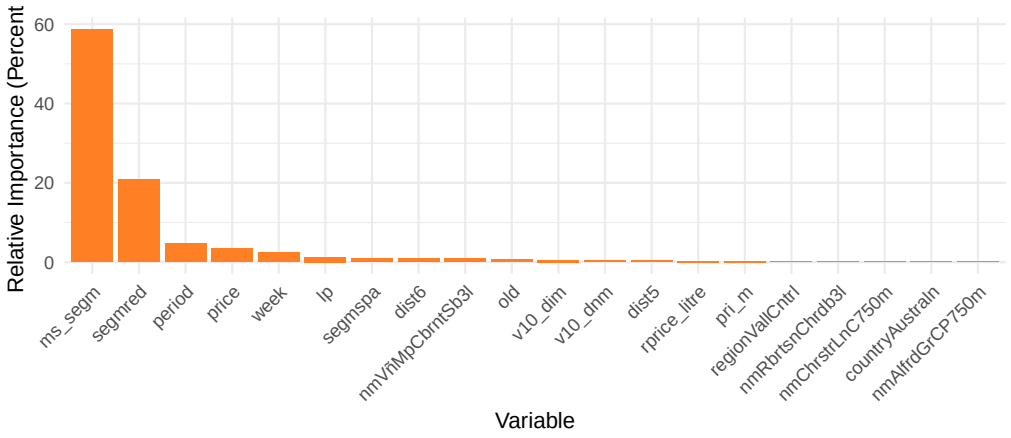

Figure 14: Boosting: Variable Importance Plot.

The corresponding partial dependence plots are very similar to those from the random forest model. Therefore, and to keep this paper at a reasonable size, they are not discussed here but can be found in figure A2 in appendix A.2.

5 Robustness

We performed various alternative estimations to validate the robustness of our results. This was done by re-estimating the best random forest model and comparing the new results. As noted in chapter 2, we excluded "*time_segm_price*" and "*artikpr*". Those variables are combinations of other variables; thus, they represent interactions terms. Including them yields a mean RMSE of ≈ 4862.917 and therefore does not improve our model.

Another crucial decision was to exclude all variables with more than 50% missing values. Reducing this to 20% means excluding 6 additional variables. Those variables are all related to wine reviews. This increases the complete cases substantially from 41416 to 144994, which is nearly the whole dataset. Estimating the random forest model yields a RMSE of ≈ 5010.796, which is an underperformance compared to the model developed in section 4.5.4. This shows that the loss of explanatory power by excluding those 6 variables is not compensated by a higher precision due to the gain of observations.

6 Conclusion

In this paper, we developed a forecasting model based on the friberg gronqvist wine dataset. We used different statistical learning methods and evaluated them in terms of their cross-validated out of sample RMSE. Table 2 presents the results for the most relevant models.

At first, we estimated a mean regression and a linear regression model without variable selection to establish a baseline. The linear regression yields a substantial improvement over the mean regression, which hints at some explanatory power of our variables.

Following, we introduced the lasso to achieve a reduction of the feature set to reduce the possible overfit and therefore improve the out of sample forecasts. The lasso yields a small improvement compared to the basic linear regression model. Additionally, we used pcr and pls for dimension reduction. While pls reduces the feature set to ≈ 35 components, neither pls nor pcr outperforms the lasso approach.

We considered splines as a way to introduce nonlinearity to our model. We calculated the model using a wide range of knots to identify even complex nonlinear structures. The splines model with 20 knots performs best but only marginally outperforms the lasso.

Afterward, tree-based methods where applied. The results show that the performance of a single regression tree is not sufficient to outperform the previous methods. In consequence, we used different methods to combine multiple deci-

Table 2: RMSEs of every Fold for every Method.

Model	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean
Mean Regression	13334	13462	13205	13329	13371	13340
Linear Regression	5493	5410	5647	5711	5600	5572
Lasso	5491	5394	5635	5707	5512	5548
Lasso Loglitre	5861	5526	6014	5883	5777	5812
PCR	5485	5404	5629	5705	5507	5546
PLS	5773	5802	6054	7204	6174	6201
Splines (1 Knot)	5493	5410	5647	5711	5600	5572
Splines (20 Knots)	5457	5351	5572	5649	5507	5507
Single Tree	6392	6373	6594	6680	6436	6495
Single Tree (Pruned)	7923	7861	8143	7901	7850	7936
Bagging (5 Trees)	4373	4609	4617	4763	4530	4578
Bagging (25 Trees)	4169	4451	4343	4324	4242	4306
Random Forest (Best)	4243	4317	4492	4487	4332	4374
Boosting (Best)	4425	4455	4548	4632	4336	4479

sion trees. Namely bagging, random forests, and boosting. We validated the performance of each model, using a range of parameters.

Our study shows that only a few variables determine the litres of a wine sold per week. Among those are the mean market share within the color of the wine, its price (in different variations) the color of the wine, its place of origin, and the time. Partial dependence plots revealed a nonlinear influence of many variables. Furthermore, red wine is sold in a higher quantity than other wines; the overall litres sold per week decreased chronologically. We find little evidence that expert reviews and ratings have a great influence on the sales; if they have, they tend to have a strong influence only when they are excellent. This contrasts the results of some studies mentioned in the introduction.

In conclusion, the two best performing models in terms of the cross-validated out of sample RMSE are the bagging model with 25 trees and the selected random forest model. While the bagging model yields the best cross-validated out of sample RMSE, it is computationally demanding. Therefore we propose the use of a random forest model with 25 trees and 100 variables considered at each split. The use of more trees, more variables, or both barely increases the forecasting performance. Moreover, the proposed random forest model performs only little worse than the bagging model, but it demands substantially less computation time, which is why we prefer the application of a random forest model in contrast to a bagging model.

References

Ashenfelter, O., & Jones, G. V. (2013). The Demand for Expert Opinion: Bordeaux Wine. *Journal of Wine Economics*, *8*(3), 285–293. doi:10.1017/jwe.2013.22

Bicknell, K. B., & MacDonald, I. A. (2012). Regional reputation and expert opinion in the domestic market for New Zealand wine. *Journal of Wine Research*, *23*(2), 172–184. doi:10.1080/09571264.2012.676541

Elith, J., Leathwick, J. R., & Hastie, T. (2008). A working guide to boosted regression trees. *Journal of Animal Ecology*, *77*(4), 802–813. doi:10.1111/j.1365-2656.2008.01390.x

Friberg, R., & Grönqvist, E. (2012). Do Expert Reviews Affect the Demand for Wine? *American Economic Journal: Applied Economics*, *4*(1), 193–211. doi:10.1257/app.4.1.193

Hastie, T., Tibshirani, R., & Friedman, J. H. (2013). *The elements of statistical learning : Data mining, inference, and prediction* (2. ed., corr. at 7. print.). New York, NY: Springer. Retrieved from http://digitale-objekte.hbz-nrw.de/storage2/2015/11/27/file_5/6530880.pdf

Hilger, J., Rafert, G., & Villas-Boas, S. (2011). Expert opinion and the demand for experience goods: An experimental approach in the retail wine market. *The Review of Economics and Statistics*, *93*(4), 1289–1296. doi:10.1162/REST_a_00117. eprint: https://doi.org/10.1162/REST_a_00117

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2014). *An introduction to statistical learning : With applications in r*. New York, NY: Springer New York. Retrieved from https://doi.org/10.1007/978-1-4614-7138-7

Software-References

Breiman, L., Cutler, A., Liaw, A., & Wiener., M. (2018). *Randomforest: Breiman and cutler's random forests for classification and regression*. R package version 4.6-14. Retrieved from https://CRAN.R-project.org/package=randomForest

Croissant, Y., Millo, G., & Tappe, K. (2019). *Plm: Linear models for panel data*. R package version 2.1-0. Retrieved from https://CRAN.R-project.org/package=plm

Friedman, J., Hastie, T., Tibshirani, R., Simon, N., Narasimhan, B., & Qian, J. (2019). *Glmnet: Lasso and elastic-net regularized generalized linear models*. R package version 2.0-18. Retrieved from https://CRAN.R-project.org/package=glmnet

Greenwell, B., Boehmke, B., Cunningham, J., & Developers, G. (2019). *Gbm: Generalized boosted regression models*. R package version 2.1.5. Retrieved from https://CRAN.R-project.org/package=gbm

Henry, L., & Wickham, H. (2019). *Purrr: Functional programming tools*. R package version 0.3.2. Retrieved from https://CRAN.R-project.org/package=purrr

Hlavac, M. (2018). *Stargazer: Well-formatted regression and summary statistics tables*. R package version 5.2.2. Retrieved from https://CRAN.R-project.org/package=stargazer

Izrailev, S. (2014). *Tictoc: Functions for timing r scripts, as well as implementations of stack and list structures*. R package version 1.0. Retrieved from https://CRAN.R-project.org/package=tictoc

Kuhn, M., Wing, J., Weston, S., Williams, A., Keefer, C., Engelhardt, A., ... Hunt., T. (2019). *Caret: Classification and regression training*. R package version 6.0-84. Retrieved from https://CRAN.R-project.org/package=caret

Lumley, T., & Miller, A. (2017). *Leaps: Regression subset selection*. R package version 3.0. Retrieved from https://CRAN.R-project.org/package=leaps

Mevik, B.-H., Wehrens, R., & Liland, K. H. (2019). *Pls: Partial least squares and principal component regression*. R package version 2.7-1. Retrieved from https://CRAN.R-project.org/package=pls

Milborrow, S. (2019a). *Plotmo: Plot a model's residuals, response, and partial dependence plots*. R package version 3.5.5. Retrieved from https://CRAN.R-project.org/package=plotmo

Milborrow, S. (2019b). *Rpart.plot: Plot 'rpart' models: An enhanced version of 'plot.rpart'*. R package version 3.0.7. Retrieved from https://CRAN.R-project.org/package=rpart.plot

R Core Team. (2019). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. Vienna, Austria. Retrieved from https://www.R-project.org/

Ripley, B. (2019a). *Class: Functions for classification*. R package version 7.3-15. Retrieved from https://CRAN.R-project.org/package=class

Ripley, B. (2019b). *Mass: Support functions and datasets for venables and ripley's mass*. R package version 7.3-51.4. Retrieved from https://CRAN.R-project.org/package=MASS

Ripley, B. (2019c). *Tree: Classification and regression trees*. R package version 1.0-40. Retrieved from https://CRAN.R-project.org/package=tree

RStudio Team. (2019). *Rstudio: Integrated development environment for r*. Version 1.2.1541. RStudio, Inc. Boston, MA. Retrieved from http://www.rstudio.com/

Rushworth, A. (2019). *Inspectdf: Inspection, comparison and visualisation of data frames*. R package version 0.0.4. Retrieved from https://CRAN.R-project.org/package=inspectdf

Sievert, C., Parmer, C., Hocking, T., Chamberlain, S., Ram, K., Corvellec, M., & Despouy, P. (2019). *Plotly: Create interactive web graphics via 'plotly.js'*. R package version 4.9.0. Retrieved from https://CRAN.R-project.org/package=plotly

Therneau, T., & Atkinson, B. (2019). *Rpart: Recursive partitioning and regression trees*. R package version 4.1-15. Retrieved from https://CRAN.R-project.org/package=rpart

Ushey, K., Allaire, J., Wickham, H., & Ritchie, G. (2019). *Rstudioapi: Safely access the rstudio api*. R package version 0.10. Retrieved from https://CRAN.R-project.org/package=rstudioapi

Wickham, H. (2019). *Stringr: Simple, consistent wrappers for common string operations*. R package version 1.4.0. Retrieved from https://CRAN.R-project.org/package=stringr

Wickham, H., François, R., Henry, L., & Müller, K. (2019). *Dplyr: A grammar of data manipulation*. R package version 0.8.0.1. Retrieved from https://CRAN.R-project.org/package=dplyr

Wickham, H., & Henry, L. (2019). *Tidyr: Easily tidy data with 'spread()' and 'gather()' functions*. R package version 0.8.3. Retrieved from https://CRAN.R-project.org/package=tidyr

Xie, Y. (2019). *Knitr: A general-purpose package for dynamic report generation in r*. R package version 1.23. Retrieved from https://CRAN.R-project.org/package=knitr

A Appendices

A.1 Dataset

Table A1: Summary Statistics of the Dataset.

Statistic	Min	Max	Mean	St. Dev.
artikelnr	200,001	4,236,201	1,008,268.000	847,223.200
artikelid	2,000	42,362	10,082.660	8,472.232
year	2,002	2,006	2,003.849	1.157
week	1	53	26.655	15.464
period	1	215	123.181	60.995
litre	0.000	184,200.000	6,100.028	11,361.840
llitre	−0.288	9,999,979.000	6,357,811.000	2,766,722.000
price	38	1,149	109.710	86.164
lp	4,199	7,350,033	4,224,377.000	1,374,042.000
rprice_litre	43	9,997,986	4,786,868.000	3,294,301.000
old	0	6,294	1,013.839	1,670.369
ma_split	0.000	9,992,891.000	103,908.700	731,760.500
v10_a	0.000	9,996,812.000	139,915.000	839,263.000
v10_dn	0.000	1.000	0.866	0.341
v10_di	0.000	6,666,667.000	700,363.200	1,870,385.000
v10_exp	0.000	1.000	0.987	0.115
v10_svd	0.000	1.000	0.983	0.130
v10_aom	0.000	1.000	0.862	0.345
v10_am	0.000	9,642,858.000	843,582.100	2,364,607.000
v10_dnm	0.000	9,791,667.000	2,781,325.000	3,473,845.000
v10_dim	0.000	9,642,858.000	2,152,525.000	3,281,604.000
v10_expm	0.000	9,714,286.000	1,306,189.000	2,935,431.000
v10_svdm	0.000	9,833,333.000	868,634.200	2,474,596.000
v10_aomm	0.000	9,629,629.000	3,414,758.000	2,908,400.000
v10_all	0.000	9,710,261.000	3,998,467.000	2,944,359.000
rev_all	0.000	9,816,509.000	77,025.290	679,549.600
rev_all_hi	0	1	0.038	0.190
rev_all_lo	0	1	0.019	0.137
rev_eve	0	1	0.015	0.122
rev_eve_hi	0	1	0.010	0.098
rev_eve_lo	0	1	0.003	0.058
rev_ex	0	1	0.010	0.101
rev_ex_hi	0	1	0.008	0.089
rev_ex_lo	0	1	0.006	0.079
rev_nyaom	0	1	0.033	0.179
rev_nyaom_hi	0	1	0.029	0.169
rev_nyaom_lo	0	1	0.013	0.113
rev_all_p50	0	1	0.024	0.152
rev_all_p80	0	1	0.017	0.130
rev_all_p20	0	1	0.010	0.101
m_rev	0	1	0.009	0.094
m_rev_hi	0	1	0.003	0.058
m_rev_lo	0	1	0.001	0.031
nrarom	0	793	14.917	74.440
pri_m	0	9,990,796	3,046,590.000	3,513,496.000
ms_segm	0	9,984,732	1,846,902.000	3,136,838.000
ind	0.00000	1.000	0.014	0.097
...59	1.000	1.000	1.000	0.000

A.2 Additional Partial Dependence Plots

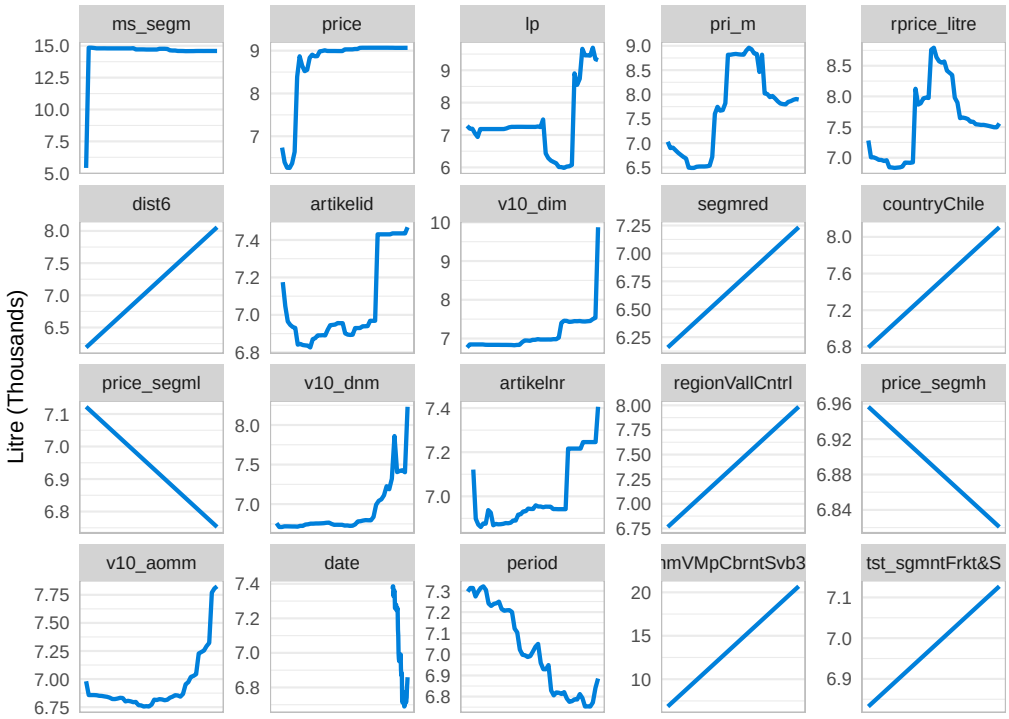

Figure A1: Random Forest: Partial Dependence Plots.

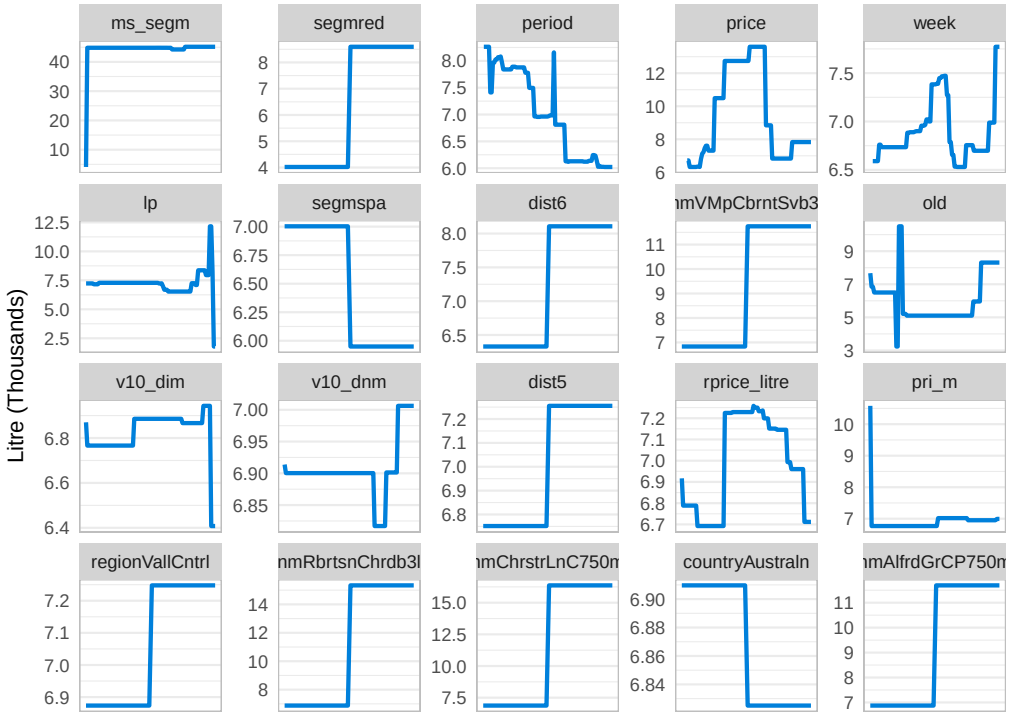

Figure A2: Boosting: Partial Dependence Plots.

Eidesstattliche Versicherung

Ich versichere an Eides statt durch meine Unterschrift, dass ich die vorstehende Arbeit selbständig und ohne fremde Hilfe angefertigt und alle Stellen, die ich wörtlich oder annähernd wörtlich aus Veröffentlichungen entnommen habe, als solche kenntlich gemacht habe, mich auch keiner anderen als der angegebenen Literatur oder sonstiger Hilfsmittel bedient habe. Die Arbeit hat in dieser oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen.

Essen, den _____ _____

Jonathan Berrisch, Timo Rammert