Technische Universität München

Fakultät für Informatik

Seminar Data Mining

Prudsy's Data Mining Cup 2017

Manuel Freytag, Manuel Stroehmer, Matheus Raszl, Nina Pischke

Contents

1 Introduction

The setting of the task of this year's Data Mining Cup by Prudsys was an online pharmacy that makes use of a dynamic pricing strategy in the form of daily article-based price adjustments [Data]. The data provided for this task was real anonymized data. The task was to predict the revenue per user action. The solution was approached as proposed by the CRISP process model, which is defined by: business understanding, data understanding, data preparation, modeling and evaluation [Wirt 00].

2 Business Understanding

In terms of business understanding, research of the online pharmacy market reveals a general trend towards self-medication [BVDV] and price-conscious customers. The decision between brands and generic products appears to be an important factor to some groups of customers. Additionally, the dynamic pricing strategy of the web shop needs to be considered: By aligning prices to the demand continuously, the web shop is able to fully exploit market potential. To sum up, pricing is a very important factor in the business model of an online pharmacy. Thus, this is one of the most crucial attributes to be considered in the feature engineering part later on.

3 Data Understanding

The provided data consists of three data sets: a training data set containing 2,756,003 observations, a classification data set containing 1,210,767 observations and a product data set containing 22,035 observations. The product data provides additional information about the different products that are occurring in the training and classification data. In both, the training and the classification data, an observation represents one user action regarding a particular product. The training data represents all user actions of the online pharmacy within a period of three months. All user actions of the following month are represented in the classification data. In most of the user actions no purchase of the product occurs and therefore the generated revenue of those user actions is zero. When a purchase occurs, it is possible that more than one unit of the product was purchased. The generated revenue of such a user action therefore equals the number of units sold, times the price of the product at that time.

4 Data Preparation

4.1 Data Transformation

4.1.1 Missing Values

Across the various attributes, missing values are represented differently with zeros, empty character vectors or NA values. In some cases, for example with the attribute that lists the competitor price, identifying missing values requires semantic interpretation. A competitor price of zero would mean that the competitor is giving away the product for free, which is highly unlikely. Therefore, zeros were interpreted as missing values. An imputation of missing values in general was not useful, since they were not missing at random according to the distribution of orders. Thus, missing values were replaced by uniform values.

4.1.2 Removing Attributes

Before the model building, attributes were removed from the training data, that do not occur in the classification data. Additionally, IDs and ID-like features were removed. Although the attribute *ProductID* is not a primary key to the training data, it is an ID-like feature with roughly 20,000 different values that identify the different products and can therefore lead to overfitting of the model. Due to the chronological order of the training and classification data set, the *Day* attribute only contains values from 1 to 92 for the training data set and only values from 93 to 123 for the classification data. Therefore the attribute *Day* is also removed, because it would bias the model.

4.2 Feature Engineering

4.2.1 Price Attributes

As described in Chapter 2, pricing is one of the most important factors for customers of an online pharmacy. Therefore, we created 14 attributes in total to model the customers' price perception. Assuming that some users have already seen a product on the online pharmacy platform before, it is possible that they are sensitive to price changes over time. Therefore, the all time high, the all time low and the average price for each product was determined. Additional attributes were created by calculating the absolute and relative differences between the price at the time and all other price attributes, like the competitor price, for each observation.

4.2.2 Modeling the Buying Decision

Often times when users enter an online pharmacy platform, they are looking for a specific product that helps them with a specific medical problem, like a headache. Since there are at least 22,035 different products listed on the observed platform, it is most likely, that more than one product can help with the problem and that the users have to decide between them. In order to model decision spaces of the customers, the products have to be clustered into product clusters that potentially solve the same medical problem. Within each cluster, the products need to be ranked by their appeal to be bought. We decided to use the price as the indicator of product appeal, since it performed better than other attributes like for example the price per content. The diagram shown in Appendix A illustrates a potential buying decision, where a customer wants to buy a cough sirup. The different cough sirups are represented by green pills and the customer decides to buy the cheapest one.

4.2.3 Trend Analysis

We analyzed trends for each product to account for individual time related effects on the customer buying decision. Two approaches were compared to derive potential trends: linear regression, where the slope of the function indicates a trend, and seasonal decomposition of the time series [Clev 90]. The trend analysis was performed, in case we were able to calculate a value for every day. Products with missing data were treated as if they have no trend. We decided to normalize all trend values on the product average of the respective attribute to enable the combination of the product specific trends within one attribute. The resulting p-values of the linear regression were below a 1% significance level. Therefore, we did not use any trend attribute derived by the linear regression. As the observation time for the available data is four months, the only reasonable and non arbitrary cyclic duration is weeks. We performed the trend analysis for multiple attributes. The order rate per product per day is expected to be one of the best indicators for the willingness of a customer to buy a product. Therefore, changing customer trends are important to take into account. However, this attribute can not be calculated for a set with unknown classification. We are forced to continue the cyclic and trend component as a forecast, as order information remains unknown. This reduces reliability of this attribute. Secondly, the number of user actions per day combined with the order rate is expected to indicate the customer's price perception. Appendix B visualizes the mean and standard deviation of the accepted trend attributes.

5 Modeling

5.1 One-label and Two-label approach

In order to predict the revenue, two different approaches were applied. Initially, a two-label approach was implemented. A binary classification predicted the first label, which determined whether an order occurred or not. The second label, the quantity of the item that was potentially ordered, was predicted with linear regression. The revenue was then calculated by multiplying the binary classification outcome, the predicted quantity and the price, as equation 1 illustrates.

$$Revenue(x) = order(x) * quantity(x) * price(x) \tag{1}$$

The performance of the two-label approach, based on an untuned C5.0 algorithm and a simple linear regression, performed significantly worse than a one-label approach with a simple linear regression directly on the revenue. Therefore, the two-label approach was omitted and only the one-label approach was pursued.

5.2 Sampling

With 2,756,003 classified instances we decided to use a large uniform test set to ensure comparability of the results. Additionally, it should resemble the data structure as closely as possible. As the data is based on a time series and the classification set is the last part of this time series, we selected the last 550,000 instances in the training set to resemble the data structure. In general it is always preferable to train on every available instance to reduce the possibility of bias. Due to limitations in computational power, we have not been able to train every used learner on all remaining 2,200,000 instances. However, even with reduced train size both underfitting and overfitting would be revealed by a uniform and sufficiently large test set. We therefore determined a feasible sample size for every learner. This is done to find a good tradeoff between result quality and processing time. For every learner that needed to train on a reduced training set, we used random sampling to avoid selection bias as much as possible.

5.3 Selection of base learners

We based our learner selection on the used MLR package and expected computational demand. To minimize the heterogeneity of additional tuning and data preparation done by each team member, we distributed different categories based on base learners within the team. Ensemble and boosting enhancements to a base learner have to be considered by each team member as well. A list of identified categories can be observed in the Appendix C. We also marked the decision whether or not to include the category in the training and evaluation process. The distribution of learners within the team for the performance comparison was based on the personal expertise and knowledge of the team members.

5.4 Learner specific data preparation

The majority of data preparation is done unanimously for the whole dataset as described in Chapter 4. However, the base learners differ from each other in their assumptions and data compatibilities, thus requiring individual attribute selection. Especially factorial data often cannot be handled by multiple base learners that require a distance metric of some sort, like regression or the k-nearest neighbors algorithm. Binning and dummy encoding enabled partial use of the information provided by the factorial data.

Potentially irrelevant attributes could add noise to the data set and therefore reduce the performance of a learner [Witt 11, p. 308]. Firstly, we used manual attribute selection based on significance levels, data structure and visualization insights. Afterwards we performed automated attribute selection on the potentially relevant attributes to remove noise potential. We made use of both filter and wrapper attribute selection techniques and selected the most fitting attribute selection methods for each learner. This means that for example the chi squared and correlation based filter approach are performed for a linear regression algorithm and compared with the performance of the feed forward wrapper approach.

5.5 Parameter tuning

Finally, we tuned the parameters of each learner to optimize their performance on our dataset. For this we used a tuning grid with a selection of parameters that proved to be the most efficient performance levers according to personal experience. For example, the minimum number of instances in one leaf affects the level of generalization done by a tree learner. This is often more set dependent than the selection of a split criteria where one is dominating in most cases.

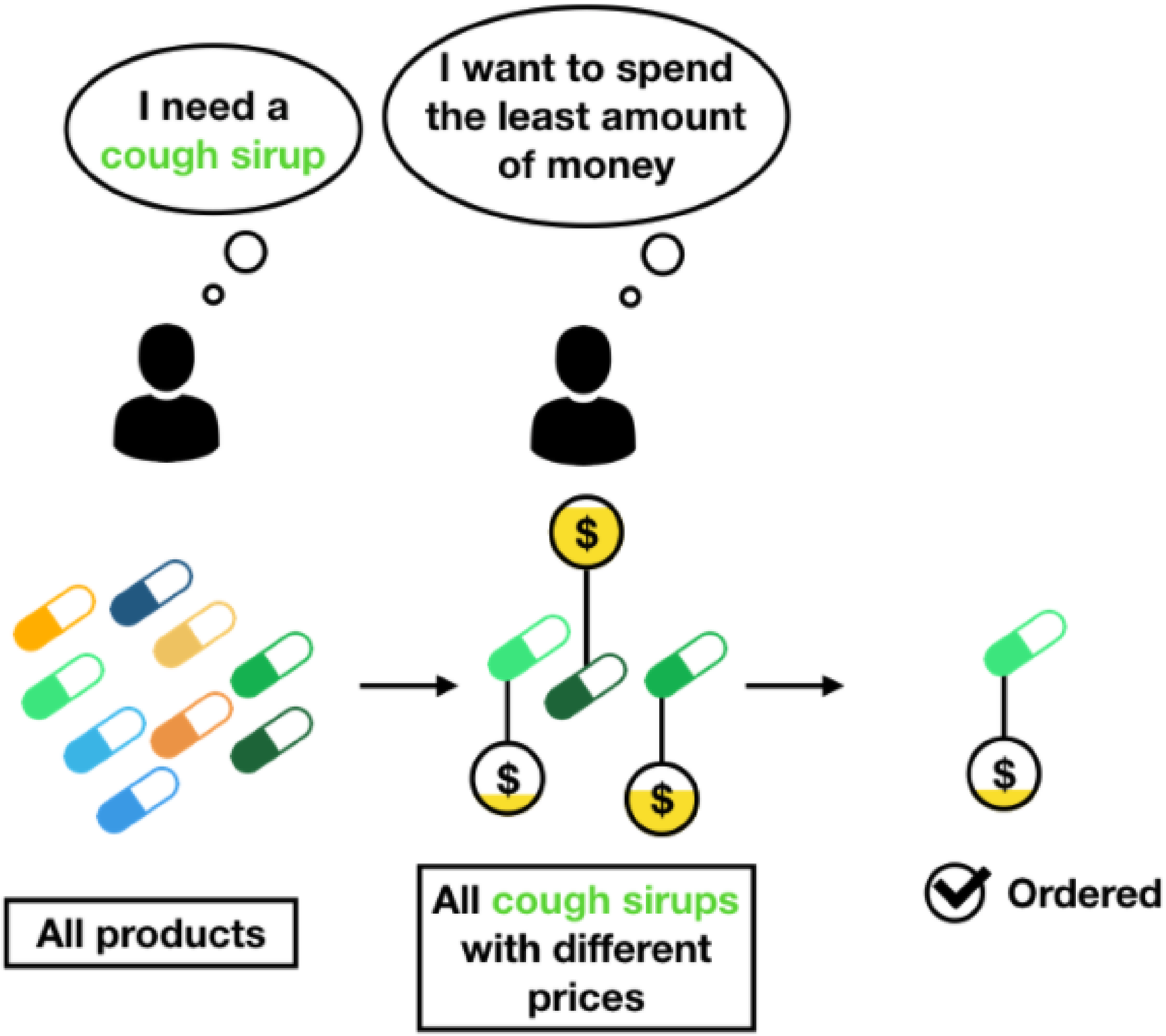
6 Hardware and Software Limitations

The entire preprocessing and modeling was performed with R. We mainly used functions provided by the MLR package. Data exploration, visualization and code explanation was done in part with Python and Jupyther. Every computation was performed on personal laptops with differing processing power. Unfortunately, the computation on the LRZ cluster was not possible because of an shutdown of the servers for three days.

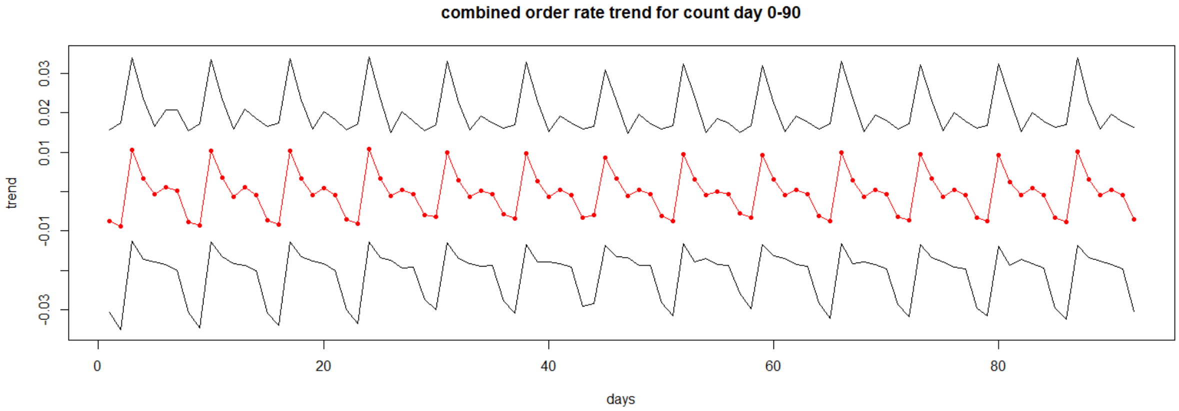
7 Discussion and Results

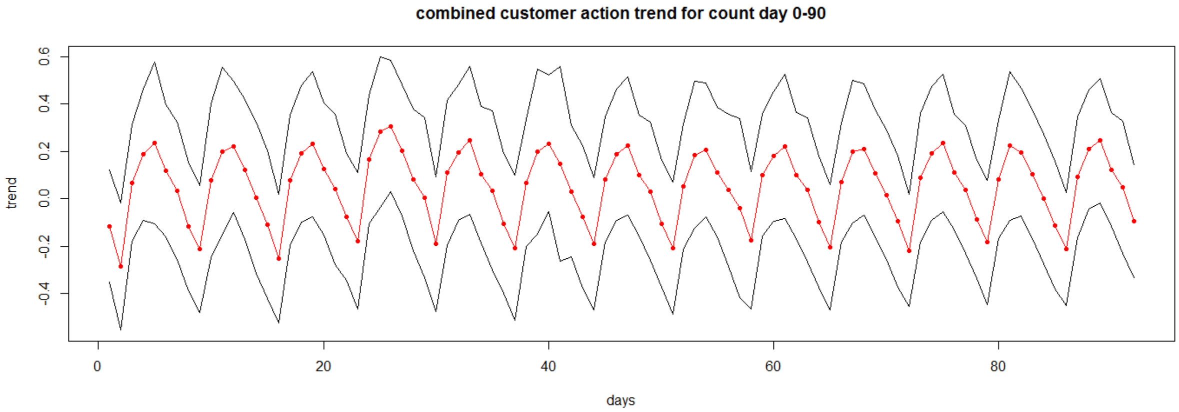
The table in Appendix D lists an excerpt of the different learners we applied, as well as the sample size that was used and the resulting RMSE. Additionally, for each learner, the most important learner-specific data preparation steps and parameter tunings are listed, which lead to the best performance of the respective model. The descriptions of the parameter tunings are relative to the default parameter settings of each learner. For example *Increase of the minimum bucket size* for the conditional interference trees means, that the model performed better when using a higher minimum bucket size than the default size. For benchmarking purposes, a simple linear regression on the raw data set was applied, which resulted in an RMSE of 9.93. Our best performing learner, the conditional interference tree, with respective data preparation yields an improvement of roughly three percent with an RMSE of 9.61. Due to limited computational power, it was not possible to train the model on all of the $\sim$ 2,200,000 instances within reasonable time. Therefore, we trained the model on 600,000 instances instead. We believe, that training the model on all possible instances would lead to a substantial improvement.

A Buying Decision

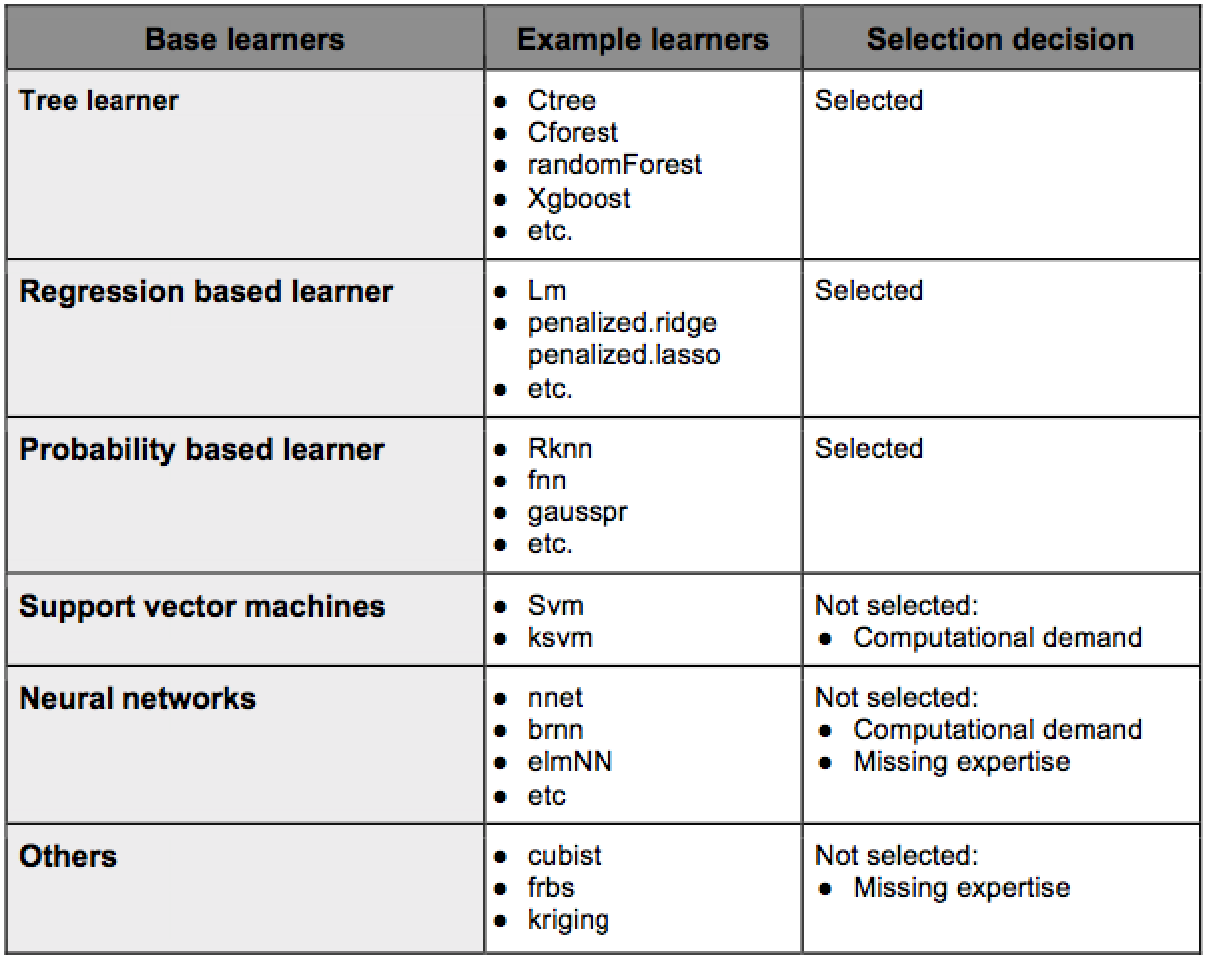

I need a cough sirup
I want to spend the least amount of money
$
$
$
$
All products
All cough sirups with different prices
Ordered

B Trend Analysis


combined order rate trend for count day 0-90

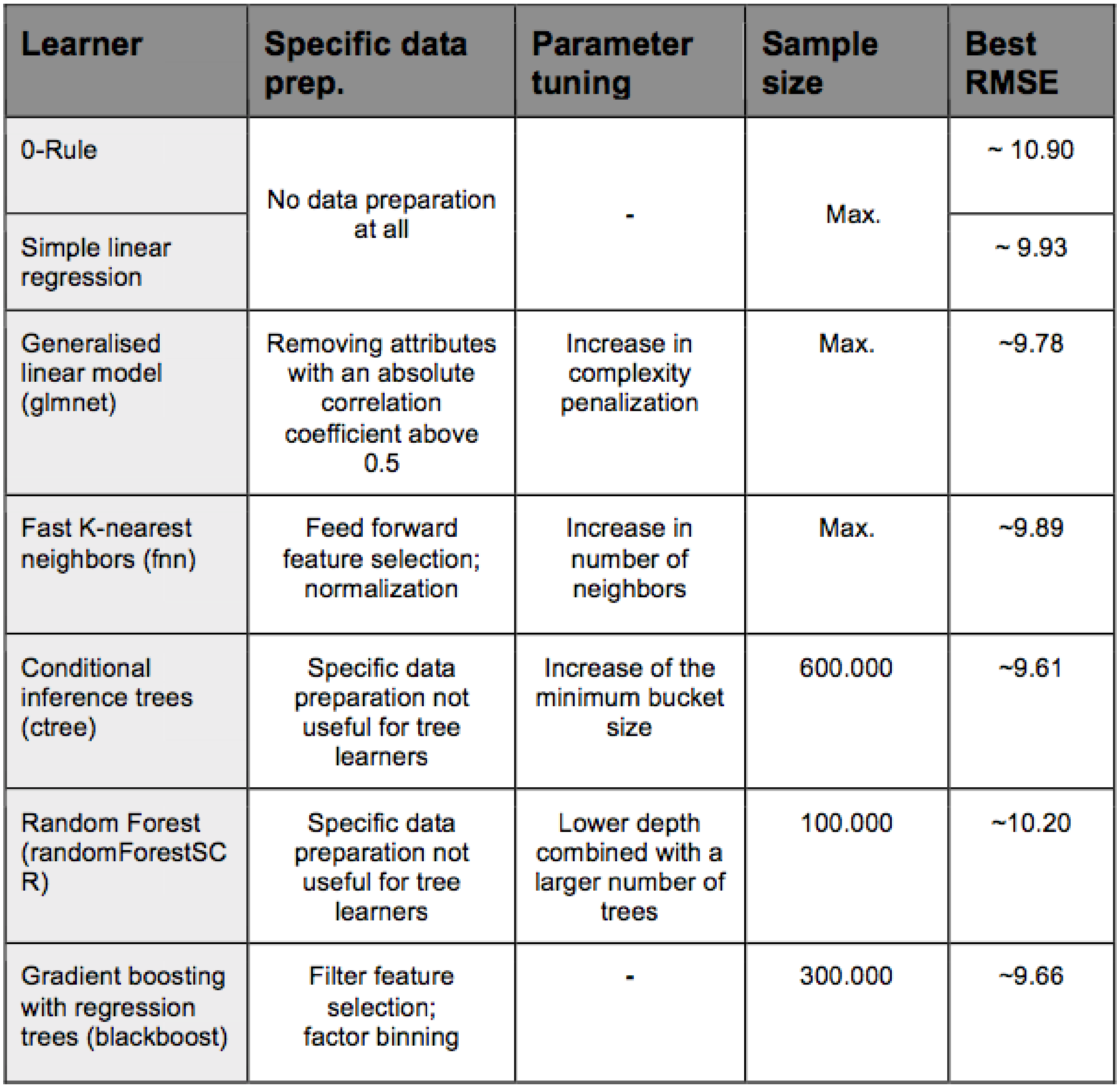

combined customer action trend for count day 0-90

C Base Learners



Base learners	Example learners	Selection decision
Tree learner	• Ctree • Cforest • randomForest • Xgboost • etc.	Selected
Regression based learner	• Lm • penalized.ridge penalized.lasso • etc.	Selected
Probability based learner	• Rknn • fnn • gausspr • etc.	Selected
Support vector machines	• Svm • ksvm	Not selected: • Computational demand
Neural networks	• nnet • brnn • elmNN • etc	Not selected: • Computational demand • Missing expertise
Others	• cubist • frbs • kriging	Not selected: • Missing expertise

D Learners



Learner	Specific data prep.	Parameter tuning	Sample size	Best RMSE
0-Rule	No data preparation at all	-	Max.	~ 10.90
Simple linear regression				~ 9.93
Generalised linear model (glmnet)	Removing attributes with an absolute correlation coefficient above 0.5	Increase in complexity penalization	Max.	~9.78
Fast K-nearest neighbors (fnn)	Feed forward feature selection; normalization	Increase in number of neighbors	Max.	~9.89
Conditional inference trees (ctree)	Specific data preparation not useful for tree learners	Increase of the minimum bucket size	600.000	~9.61
Random Forest (randomForestSCR)	Specific data preparation not useful for tree learners	Lower depth combined with a larger number of trees	100.000	~10.20
Gradient boosting with regression trees (blackboost)	Filter feature selection; factor binning	-	300.000	~9.66

References

[BVDV] BVDVA. “Daten und Fakten zum Arzneimittelversandhandel in Deutschland”. http://www.bvdva.de/daten-und-fakten. Accessed: 2017-06-20.

[Clev 90] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning. “STL: A Seasonal-Trend Decomposition Procedure Based on Loess”. *Journal of Official Statistics*, Vol. 6, No. 1, pp. 3–73, 1990.

[Data] Data-Mining-Cup. “Prudsys Data Mining Cup 2017 - Task”. http://www.data-mining-cup.de/en/wettbewerb/aufgabe.html. Accessed: 2017-06-20.

[Wirt 00] R. Wirth and J. Hipp. “CRISP-DM: Towards a Standard Process Model for Data Mining”. 2000.

[Witt 11] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufman, San Francisco, USA, 2011.