Predicting eBay Prices: Selecting and Interpreting Machine Learning Models – Results of the AG DANK 2018 Data Science Competition

Gero Szepannek and Rabea Aschenbruck

Abstract The annual meeting of the work group on data analysis and numeric classification (AG DANK) took place at Stralsund University of Applied Sciences, Germany on October 26^{th} and 27^{th}, 2018 with a focus theme on *interpretable machine learning*. Traditionally, the conference is accompanied by a data science competition where the participants are invited to analyze one or several data sets and compare and discuss their solutions. In 2018, the task was to predict end prices of eBay auctions. The paper describes the task as well as a discussion of the results as provided by the conference participants. These cover aspects of preprocessing, comparison of different models, task specific hyperparameter tuning as well as the interpretation of the resulting models and the relevance of additional text information.

Gero Szepannek · Rabea Aschenbruck
University of Applied Science Stralsund, Zur Schwedenschanze 15, D-18435 Stralsund, Germany
✉ gero.szepannek@hochschule-stralsund.de
✉ rabea.aschenbruck@hochschule-stralsund.de

ARCHIVES OF DATA SCIENCE, SERIES A
(ONLINE FIRST)
KIT SCIENTIFIC PUBLISHING
Vol. 7, No. 1, 2020

DOI: 10.5445/IR/1000125928
ISSN 2363-9881

1 Introduction

The annual meeting of the work group on data analysis and numeric classification (AG DANK) traditionally is accompanied by a data science competition where the participants are invited to analyze one or several data sets and compare and discuss their solutions. In 2018 the meeting took place at Stralsund University of Applied Sciences and the task of the satellite data science competition consisted in predicting end prices of eBay auctions. In addition, each conference is held under one focus topic which has been *interpretable machine learning* in 2018. For this reason the challenge not only became to develop a prediction model of highest accuracy, but the additional question was whether we can understand the model.

The organization of the remaining section is as follows: Section 2 describes the competition data and a summary of the results. While the major part of the submissions has been anonymous in the sense that the explicit methodology used is unknown to the organizers of the competition, in addition to the contributions to the competition several benchmark models have been developed for comparison. These models are described in Section 3. Section 4 is dedicated to the focus theme and attempts to interpret the resulting models are discussed. Finally, Section 5 provides a summary of the obtained results.

2 Description of the Competition

2.1 Data and Task

The data contains final auction prices and several additional attributes from 143 eBay auctions of the video game Super Mario Kart as provided by Diez et al. (2017). The data set consists of 9 variables which are described in Table 1. In general, the variables do contain information on the seller (such as *sellerRate*) or the auction (e.g. *startPr* and *duration*) as well as the explicit item sold (*wheels* and *cond*) which is comparable to the type of data used by other studies (Shmueli, 2010; Ghani and Simmons, 2012) while Ghani and Simmons (2012) used a much larger set of 430 variables. Compared to these studies the number of observations in the contest is small which typically impacts model selection. The target variable (*Price*) is continuous. Therefore, the competition consists of a regression task. In Ghani and Simmons (2012) it has alternatively been proposed to model auction prices as multiple binary classification tasks. Shmueli (2010) uses the logarithm of the end price as target variable. To do so has been left as a decision up to the participants, but the logarithm of the end price has not been used as the target variable for the benchmark models presented in Section 3.

Table 1: Description of the data.

Variable Name	Description	Type
cond	Condition of the selling product	categorical {used, new}
duration	Auction length in days	numeric
nBids	Number of bids	numeric
sellerRate	Seller's rating: no. positive - no. negative ratings	numeric
startPr	Start Price	numeric
stockPhoto	photo that is used for many auctions	categorical {yes, no}
wheels	No. steering wheels (hardware) sold with the game	numeric
title	Auction title text	text
Price	**Target variable: final price in US dollar**	numeric

For the competition the data has been subdivided into training of 100 observations and test data where no price was available to the particpipants for the remaining 43 observations. The task of the competition has been given by prediction of the missing final auction prices of the test data. As an evaluation measure for comparing different submissions the R^2 goodness of fit on the test data has been used.

2.2 Summary of the Results

A total of twelve submissions were provided by both conference participants as well as students (business informatics and management of small and medium enterprises) of a data mining class at Stralsund University of Applied Sciences. Table 2 summarizes all results. For most submissions neither the explicit model nor the performance on the training data are known. For this reason, several benchmark models have been created with the open source statistic software

R (R Core Team, 2019) and will be presented in the subsequent sections[1]: A regression tree (Section 3.1), a random forest (Section 3.2) and a linear regression model using variable selection based on adjusted R^2 (Section 3.3) together with the winning solution: A tuned support vector machine (SVM). Note that the winning solution had some additional preprocessing in terms of outlier removal from the training data. In order to quantify the effect of preprocessing different models have been created with (*) and without removing outliers (Section 3.4).

Table 2: Competition results sorted by R^2 on test data w/o outlier.

	Test Data		**Test Data w/o Outlier**		**Training Data**		
Model	R^2	MAE	R^2	MAE	R^2	MAE	RMSE
Tuned SVM (*)	0.054	8.905	**0.822**	2.532	0.771	3.224	4.52
Default SVM (*)	0.042	9.028	**0.809**	2.627	0.834	2.718	3.99
Default SVM	0.045	9.163	**0.790**	2.778	0.578	3.359	7.93
Solution 1	0.058	9.471	**0.761**	3.130			
Solution 6	0.065	9.525	**0.747**	3.217			
Solution 5	0.050	9.684	**0.744**	3.342			
Solution 2	0.023	10.447	**0.712**	4.061			
Random Forest	0.044	9.858	**0.689**	3.550	0.867	2.736	4.88
Solution 4	0.063	9.830	**0.681**	3.531			
Solution 3	0.063	10.028	**0.660**	3.730			
Linear Regression	0.095	10.356	**0.596**	4.164	0.483	4.897	8.51
Regression Tree	0.071	10.934	**0.519**	4.748	0.468	4.737	8.64

From the results shown in Table 2 it is obvious that the performance on the test data strongly decreases compared to the performance on the training data for the benchmark models where the training performance is known. With R^2 close to 0 all models can be attested to be quite unpredictable. Further analysis of the target variable gives the reason for this: The test data includes one outlier observation with a price of more than \$ 300 which is more than twice as high as all prices in the training data and thus difficult to extrapolate for any model (cf. Figure 1). Nonetheless, such a situation might also occur in real life and thus deserves

[1] Supplementary code is available with this paper.

some further investigation. Here, having a closer look at the (text-)variable *title* provides the answer: *"Nintedo Wii Console Bundle Guitar Hero 5 Mario Kart"*. In fact here, a bundle of two procducts (Mario Kart as well as Guitar Hero 5) is offered which explains the comparatively higher price. This could have been found out by manual inspection of the variable but due to the small number of (training-) data text mining (cf. e.g. Feinerer et al., 2008) would not have been helpful in this situation as has been reported by several participants. In order to avoid over-emphasizing this outlier during model evaluation it has been removed for performance evaluation (column 3 and 4 in Table 2 competition results).

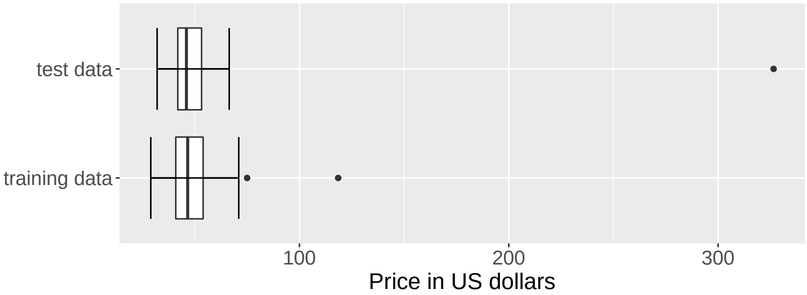

Figure 1: Comparison of prices for training and test data.

In the following Section 3 four different models are presented and discussed with respect to their performance compared to the anonymous submissions of the participants as well as interpretability which has been one of the main topics of the conference. In Section 4 the interpretability of the different models is discussed as well as a framework for model agnostic interpretation that principally can be applied to arbitrary machine learning models. Finally, Section 5 summarizes the conclusions that can be drawn from running the competition.

3 Modelling Approaches

3.1 Regression Tree

For reasons of their easy interpretability regression trees belong to the most popular techniques in Data Science (cf. e.g. KDnuggets (2020)). For this reason a regression tree model (Therneau and Atkinson, 1997) has been built as a reference model using default parameters (i.e. at least 20 observations in each node and 7 observations in each leaf of the tree as well as a penalty of 0.01 on the target criterion for each additional leaf for pruning). Figure 2 shows the resulting model: The number of wheels (i.e. additional hardware being sold together with the software) turns out to be the most predictive variable. Roughly 40% of the games are sold with more than one wheel. Further helpful predictors are the number of bids and the condition (i.e. new or used).

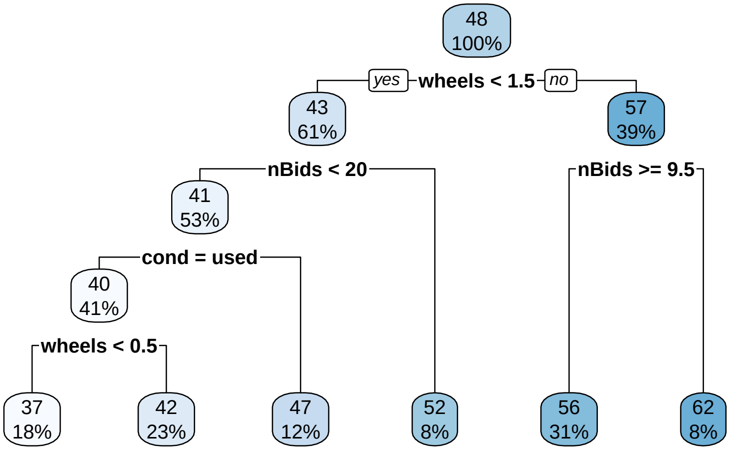

Figure 2: Regression tree model.

A major drawback of trees is their comparatively low performance in many data situations (cf. e.g. Szepannek et al., 2008, 2010) which is also observed on the eBay data.

3.2 Random Forest

Random forests (Breiman, 2001) overcome two major drawbacks of regression trees: Missing flexibility to adapt to data in case of flat trees as well as overfitting the training data for deep trees by bootstrap aggregation where in addition for each split only a random subsample of the variables is considered. One of the most important tuning parameters of random forests is the formerly mentioned number of variables (*mtry*). But when compared to other algorithms random forests are relatively insensitive to hyperparameter tuning (Szepannek, 2017; Probst et al., 2018b,a). Here, a forest has been fit using the default parameters of 500 trees and $mtry = \sqrt{p}$ of the number of variables p which is a common default for classification problems. As a price of the bootstrapping the easy interpretation is lost (cf. Section 4). For the random forest model a drop in performance between training data and test data is observed: Each observation will be selected in roughly 63% of the samples and thus is overfitted by the majority of bootstrap samples (cf. Efron and Tibshirani, 1997).

3.3 Linear Regression and Variable Selection

For reasons of its simplicity and its lack of tunability, a linear regression model can serve as a baseline for further performance comparisons. The only tunable ”parameter” is given by the subset of variables that is used for modelling. Here, a linear regression model has been computed using adjusted R^2 forward variable selection. The iterative selection process is given in Table 3. For the final model all variables except one (*sellerRate*) have been selected. Nonetheless, from step five on only slight improvements of the R^2_{adj} are observed. Therefore, given the small number of observations, one could even think about using less variables.

Table 3: Variable Selection for linear regression.

	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7
duration	0.096	0.334	0.348	0.361	**0.441**		
nBids	0.005	0.350	**0.367**				
cond	0.153	**0.351**					
startPr	0.044	0.324	0.359	**0.435**			
sellerRate	-0.009	0.326	0.348	0.365	0.429	0.436	0.445
stockPhoto	-0.008	0.314	0.350	0.370	0.439	**0.450**	
wheels	**0.320**						

For reasons of its linearity the resulting model is easy to interpret by regarding the coefficients given in Table 4. The variables *cond*, *duration* and *stockPhoto* are without significant effect given all other variables. Of course, the absolute regression coefficients can only be interpreted while simultaneously taking into account for the variability of the corresponding explanatory variables, but it can be easily seen, that e.g. each additional wheel increases the prediction by $ 5.733 and the prediction for a used product as opposed to a new product is $ 3.452 smaller if all other variables take the same values.

Table 4: Linear regression model.

Variable	Estimate	Std. Error	t-Value	Pr(> | t |)
(Intercept)	35.575	4.862	7.317	0.000
wheels	5.733	1.186	4.835	0.000
cond (reference: Used)	-3.452	2.357	-1.464	0.146
nBids	0.790	0.201	3.926	0.000
startPr	0.315	0.081	3.899	0.000
duration	-0.746	0.433	-1.723	0.088
stockPhoto (reference: Yes)	-3.615	2.257	-1.602	0.113

3.4 The Winning Solution: Support Vector Machine and Hyperparameter Tuning

For the support vector machine (SVM) winner solution the data have first been preprocessed into an apropriate format, i.e. first the categorical variables (*cond* and *stockPhoto*) have been recoded into numeric with attributes

$\{-1, 1\}$. As support vector machines rely on distances between observations (Vapnik, 2000) all other (numeric) variables have been z-transformed. Note that the same transformation (i.e. means $\bar{x}$ and standard deviations s as computed on the training data) has also to be used for scaling of the test data. In addition two outliers with respect to the target variable have been removed from the training data that are outside ± 1.58 *Interquartile range* $/\sqrt{n}$ from the median (McGill et al., 1978, cf. Fig. 1).

In order to train a support vector machine a kernel function has to be chosen and afterwards the corresponding kernel parameters must be specified. The winning solution is based on Meyer (2019b) where as a default a radial basis kernel is used. This kernel turned out to be a good choice for different data sets in several benchmark studies (cf. e.g. Szepannek et al., 2008, 2010). In order to facilitate the process of setting search spaces for tuning (i.e. hyperparameters and their ranges) the R package `mlrHyperopt` (Richter, 2017) offers a helpful tool which provides common default search spaces for different classes of algorithms. The two most common parameters for a radial basis kernel are the scaling parameter γ that specifies influence of the distance between observations on the resulting decision boundaries and is given by $e^{-\gamma|x-y|^2}$ for two observations x and y as well as the cost parameter C that controls the trade off between the width of the margin and the loss (Meyer, 2019a).

In order to parameterize a support vector machine these hyperparameters can be tuned using further subdivision of the training data. Optimization of the hyperparameters can be done automatically e.g. using the recent `mlr3` package (Lang et al., 2019). In this case the internal `tune()` function of the `e1071` package (Meyer, 2019b) has been used which optimizes the parameters over a grid based on cross validated mean squared errors on the training data. For the contest a parameter grid $(\gamma, C) \in$ {0, 0.001, 0.0025, 0.005, 0.0075, 0.01, 0.02, 0.025, 0.03, 0.04, 0.05, 0.06, 0.07, 0.075, 0.08, 0.09, 0.1, 0.25, 0.5, 0.75, 1, 5, 10} × {0.1, 0.5, 1, 5, 10, 15, 20, 25, 26, ..., 45, 50, 55, 60, 75, 100} has been investigated. This results in an optimal parameterization of $\gamma = 0.01$ and $C = 5$. In order to investigate the effect of pamameter tuning, two additional models have been computed using the default parameters $\gamma = C = 1$ with (*) and without preprocessing of the training data.

As can be seen in the results shown in Table 2, the three SVM models do outperform all other submitted solutions for this data situation. Furthermore, the notable performance difference between both models demonstrates the effect of hyperparameter tuning. By comparing both default SVMs (with and without (*)) it can be seen that the effect of tuning is stronger than the effect of preprocessing by removing outliers on this data. As opposed to the default SVM, the winning model is not overfitting the training data. A more general analysis of tunability of machine learning models is given in Probst, Wright, and Boulesteix (2018b). Unfortunately, as a price for increased model flexibility of radial basis SVMs the resulting models cannot be understood as easily anymore which is going to be discussed in the following Section.

4 Interpretable Machine Learning

As it has been outlined in the previous section the regression tree as well as the linear regression model can be easily understood while this is not possible for the random forest and the support vector machines. Talking about interpretability of machine learning models one can distinguish between different aspects of interpretability that are linked to different requirements:

1. What are the relevant variables of a model?
2. How do explicit variables influence the predictions by a model?
3. Can we explain a single prediction as a function of the values that the explanatory values do take?

An overview on different existing model agnostic approaches to interpretable machine learning is given in Scholbeck et al. (2019). At the meeting the `R` framework `DALEX` (Biecek, 2018) has been presented which provides a unified interface to solutions for all the above mentioned facets of interpretability. For the purpose of the competition the explanation of single observation has not been of interest. Therefore, we will concentrate on the first two aspects and present examples for this competition task under investigation.

4.1 Identification of Relevant Variables

For linear regression models an understanding of the importance of a predictor variable can be assessed by regarding its (absolute) effects related to the variability of the variable itself. For regression tree models it can be assessed by the tree structure, i.e. which variables are used for splits and at which position of the tree (i.e. to what percentage of the data is the split applied to). For more complex models such as the random forest or the support vector machines used in this study the interpretation of variables is not as obvious.

Variable permutation importance provides a general framework to overcome this issue (cf. e.g. Scholbeck et al., 2019): All observations of the data are randomly permuted for a single variable and the model's performance with regard to an arbitrary statistic is compared to the performance of the model using the original unpermutated data. Any existing dependency between the variable under investigation and the target variable will be removed by permutation and thus the drop in performance can be used to quantify the importance of each single variable for the model's predictive power. As a matter of fact the concept of variable permutation importance can be applied to any machine learning algorithm and any performance measure and has been extensively discussed at the conference. The results have to be interpreted with care as importance is calculated given all other variables enter the model. As a consequence, once calculating the importance of all variables – as it is often done for variable selection in practice – is no meaningful approach but it should rather used for backward selection which in contrast is computationally intense. Furthermore, the interpretation of variable importance might lead to misinterpretations, if there are interactions between several predictor variables (for further discussion see e.g. Groemping, 2009). Nonetheless, it has to be stated that variable importance is currently one of the most important tools to understand the relevance of predictor variables within complex machine learning models.

Figure 3 shows the variable permutation importance for the random forest as well as the tuned svm for the eBay data[2] using `DALEX` for the root mean squared error

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

[2] (For the svm the respective outlier corrected data has been used, cf. Section 3.4.)

performance measure where the y_i denote the true values of the target variable for all observations i and $\hat{y}_i$ are the corresponding predictions by the model. Just like in linear regression and the regression tree the variable *wheels* turns out to be the most relevant also for these two models. In contrast to the random forest the support vector machine appears to be stronger dominated by this variable. The baseline-shift on the x-axis results from the fact raw RMSE values are plotted (as opposed to differences or ratios to the original model) and reflects the superior performance of the svm on the training data (cf. Table 2).

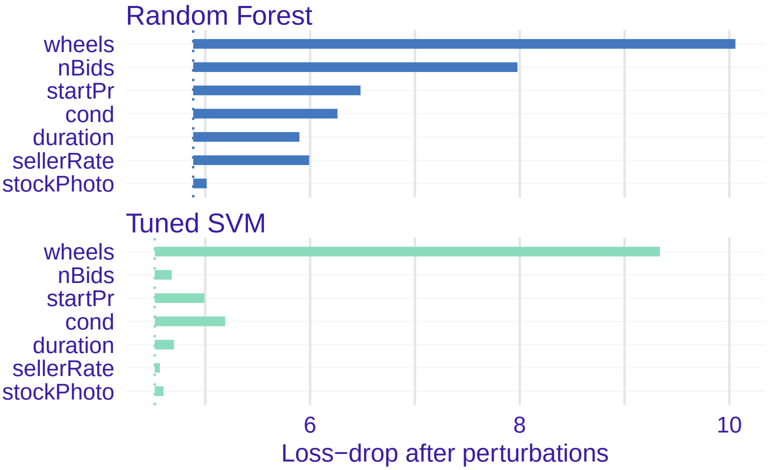

Figure 3: Variable importance for the random forest and the tuned svm.

4.2 How do Explicit Variables Effect Predictions?

The influence of variables' realizations on the prediction is easily understood for the linear model (cf. Section 3.3). For the regression tree it can be directly assessed by the splitting rules. But once again, for more complex models such as the random forest or the svms the interpretation is not evident. A standard approach to answer this question are **partial dependency plots (PDP)** dating back to Friedman (2001): The average prediction keeping one (or several) variable(s) X_s fixed

$$PD(X_s) = \int \hat{f}(X_s, X_c) dP(X_c), \tag{2}$$

where $\hat{f}()$ is the estimated function as given by the model and the vector of predictor variables (X_s, X_c) is here subdivided into two subsets X_s and X_c. In practice, the partial dependency curve is estimated by

$$\widehat{PD}(x_s) = \frac{1}{n} \sum_{i=1}^{n} \hat{f}(x_s, x_{ic}). \tag{3}$$

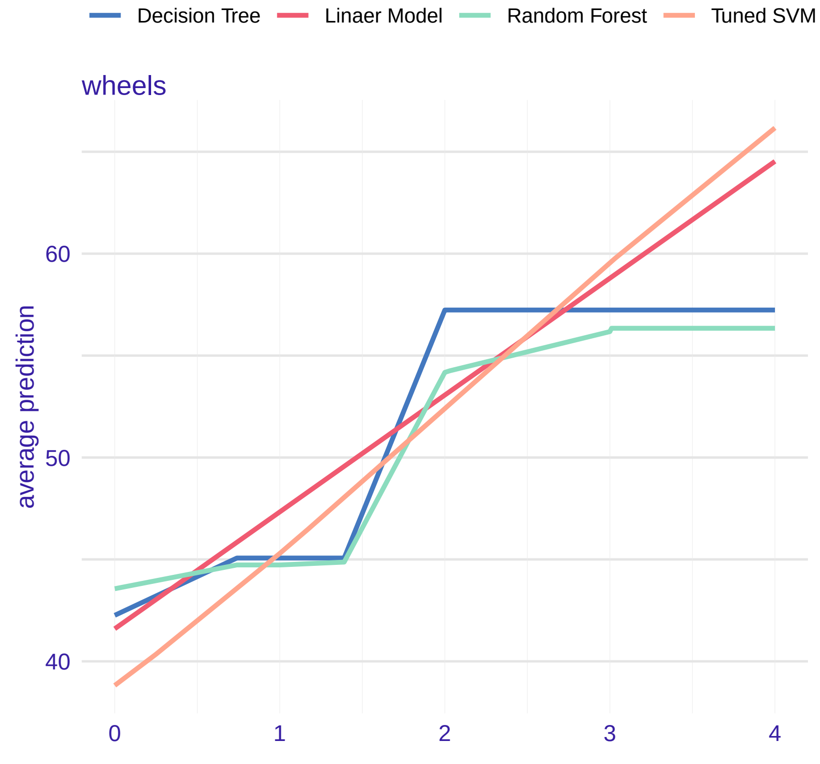

Figure 4: Partial dependency plot for the variable *wheels* and all four models.

As an example, Figure 4 shows the partial dependency plot for the most important variable *wheels*. Of course the effect on the linear regression model is given by a straight line. In contrast, for both the tree as well as the forest model a nonlinear effect can be observed: A strong increase in price is observed if more than one wheel offered saturating for even higher numbers of wheels. Astonishingly, for the winner model also a roughly linear dependency of the price on the number of wheels can be observed, although the dependency is stronger compared to the linear regression model. Note that for the support vector machine this type of dependency is not prespecified by the model. Despite the current hype on explainable machine learning, the proposed methodology has to be used with care. For PDPs e.g. it should be investigated to what extent they are able to explain the model (cf. e.g. Szepannek, 2019).

5 Summary

The paper summarizes the results on the AG DANK 2018 data science competition of predicting eBay prices for Mario Kart games with an additional scope not only on accurate predictions but also interpretation of the models.

As a first result the impact of outliers of the target variable for performance assessment has been investigated. While this can be observed on the training data, it is not possible on the test data. For the contest data the analysis of text (i.e. the auction title) provided further insights but only manual investigation led to results here, as automated text mining requires a large sample size.

A support vector machine with radial basis kernel and tuned hyperparameters showed the best results. The potential effect of hyperparameter tuning has been demonstrated as a second svm using default parameters has been computed for comparison. On this data set hyperparameter tuning not only improved results on the test data but also reduced overfitting.

Finally, options to understand the resulting models of different complexity are discussed and different requirements on interpretability are distinguished. The `DALEX` framework provides an easy interface to several implementations of model agnostic interpretation algorithms in `R`. As an example, variable importance and partial dependency plots for the most important variable are computed on the competition task eBay data set.

As it is outlined by Kusner and Loftus (2020) the increasing penetration of algorithm-based processes into our daily life results in a rising need to develop methodology to ensure algorithm fairness and it is advisable to check for each specific context whether the use of a more complex model is beneficial or whether an interpretable model could be used instead (Rudin, 2019).

Acknowledgements We to acknowledge the two anonymous reviewers for valuable comments that considerably improved the quality of the paper.

References

Biecek P (2018) DALEX: Explainers for Complex Predictive Models in R. Journal of Machine Learning Research 19(84):1–5. URL: `http://jmlr.org/papers/v19/18-416.html`.

Breiman L (2001) Random forests. Machine Learning 45(1):5–32. DOI: 10.1023/A:1010933404324.

Diez DM, Barr CD, Cetinkaya-Rundel M (2017) Openintro: Data Sets and Supplemental Functions from 'OpenIntro' Textbooks. URL: `https://CRAN.R-project.org/package=openintro`. R package version 1.7.1.

Efron I, Tibshirani R (1997) Improvements on cross-validation: The .632+ bootstrap method. Journal of the American Statistical Association (JASA) 92(543):548–560. DOI: 10.2307/2965703.

Feinerer I, Hornik K, Meyer D (2008) Text mining infrastructure in R. Journal of Statistical Software 25(5):1–54. URL: `http://www.jstatsoft.org/v25/i05/`.

Friedman J (2001) Greedy function approximation: A gradient boosting machine. Annals of Statistics 29:1189–1232. DOI: 10.1214/aos/1013203451.

Ghani R, Simmons H (2012) Predicting the end-price of online auctions. URL: `http://www.documentcloud.org/documents/406254-priceprediction.html0`.

Groemping U (2009) Variable importance assessment in regression: Linear regression vs. random forest. American Statistician 63(4):308–318. DOI: 10.1198/tast.2009.08199.

KDnuggets (2020) Top data science and machine learning methods used in 2018, 2019. URL: `https://www.kdnuggets.com/2019/04/top-data-science-machine-learning-methods-2018-2019.html`.

Kusner M, Loftus J (2020) The long road to fairer algorithms. Nature 578:34–36. DOI: 10.1038/d41586-020-00274-3.

Lang M, Bischl B, Richter J, Schratz P, Casalicchio G, Coors S, Au Q, Binderl M (2019) mlr3: Machine Learning in R - Next Generation. URL: https://CRAN.R-project.org/package=mlr3. R package version 0.1.1.

McGill R, Tukey J, Larsen W (1978) Variations of box plots. The American Statistician 32:12–16. DOI: 10.2307/2683468.

Meyer D (2019a) e1071: Misc Functions of the Department of Statistics, Probability Theory Group. URL: https://cran.r-project.org/web/packages/e1071/vignettes/svmdoc.pdf.

Meyer D (2019b) Support Vector Machines – The Interface to libsvm in package e1071. URL: https://CRAN.R-project.org/package=e1071. R package version 1.7.2.

Probst P, Bischl B, Boulesteix AL (2018a) Tunability: Importance of hyperparameters of machine learning algorithms. URL: https://arxiv.org/abs/1802.09596.

Probst P, Wright M, Boulesteix AL (2018b) Hyperparameters and tuning strategies for random forest. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 9(3):e1301. DOI: 10.1002/widm.1301.

R Core Team (2019) R: A Language and Environment for Statistical Computing. Vienna, Austria. R Foundation for Statistical Computing, Vienna, Austria. URL: https://www.R-project.org/.

Richter J (2017) mlrHyperopt: Easy Hyperparameteroptimization with mlr and mlrMBO. URL: https://github.com/jakob-r/mlrHyperopt. R package version 0.1.0.

Rudin C (2019) Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence 1(5):206–215, Springer Science and Business Media LLC. DOI: 10.1038/s42256-019-0048-x.

Scholbeck C, Molnar C, Heumann C, Bischl B, Casalicchio G (2019) Sampling, Intervention, Prediction, Aggregation: A Generalized Framework for Model Agnostic Interpretations. URL: https://arxiv.org/abs/1904.03959.

Shmueli G (2010) To Explain or to Predict. Statistical Science 25(3):289–310. DOI: 10.1214/10-STS330.

Szepannek G (2017) On the practical relevance of modern Machine Learning Algorithms for Credit Scoring Applications. WIAS Report Series 29:88–96. DOI: 10.20347/wias.report.29.

Szepannek G (2019) How much can we see? A note on quantifying explainability of machine learning models. Published via: arxiv. URL: http://arxiv.org/abs/1910.13376.

Szepannek G, Schiffner J, Wilsonl J, Weihs C (2008) Local Modelling in Classification. In: Perner P (ed.), Advances in Data Mining, Springer Lecture Notes in Artificial Intelligence (LNAI), pp. 153–164. DOI: 10.1007/978-3-540-70720-2_12.

Szepannek G, Gruhne M, Bischl B, Krey S, Harczos T, Klefenz F, Weihs C (2010) Perceptually Based Phoneme Recognition in Popular Music. In: Locareck-Junge H, Weihs C (eds.), Classification as a Tool for Research, pp. 731–758. DOI: 10.1007/978-3-642-10745-0_83.

Therneau T, Atkinson E (1997) An introduction to recursive partitioning using the RPART routines. TR Mayo Foundation. URL: `https://www.mayo.edu/research/documents/biostat-61pdf/doc-10026699`.

Vapnik V (2000) The Nature of Statistical Learning Theory. Springer, New York, NY, USA.