Business Analytics and Data Science

Assignment

Amanda Maiwald

14/03/2019

Contents

1 Introduction 1

2 Exploratory data analysis 1

2.1 Initial description 1

2.2 Overview 1

2.3 Exploratory data analysis 3

3 Data preparation 5

3.1 Errors and missing values 5

3.2 Factor Variables 6

3.3 Feature Engineering 6

4 Model tuning and selection 8

4.1 Logistic Regression 8

4.2 Random Forest 8

4.3 Gradient Boosting 9

4.4 Cost-Sensitivity 9

5 Special modeling challenge: Analyzing and interpreting variable importance 10

6 Conclusion 12

7 Bibliography 12

1 Introduction

This paper analyzes the real-world data by an online-retailer. Firstly the available data will be analyzed exploratory. Subsequently the data preparation process, that is necessary to build a predictive model, will be described. This entails the cleaning process as well as feature engineering. Afterwards different machine learning models will be considered and after reviewing the results, the model of a random forest will be chosen for the predictive model of this paper. The model will be tuned according to relevant metrics. Since miss classification of products will result in different costs the model will then be critically evaluated according to the cost matrix. Following this, the model will be inspected further by analyzing and interpreting variable importance. Lastly this paper will end with a conclusion.

Table 1: Frequency of returns

Return	Frequency [%]
0	52
1	48

2 Exploratory data analysis

2.1 Initial description

The basis for the following data analysis are two real-world data sets. The observations show orders which have been made in this online shop. The first data set, which from now on will be referred to as known shows information regarding orders that have been made in the past. It consists of 100.000 observations of 14 different variables. It is important to note that known includes the variable `return` which documents whether an ordered item has been returned.

The second data set, which will be referred to as unknown, shows similar information like the known data set. However for these orders it is unknown whether they have been or will be returned. The unknown data set is considerably smaller with only 50.000 observations. The cause for this lies in the fact, that this paper is related to a Kaggle Data challenge were a threshold of 69% AUC had to be passed. The other 50.000 observations of the unknown data set are used to benchmark scores on this platform. Since returned items have not been identified it only consists of 13 variables.

2.2 Overview

2.2.1 Independent Variables

The first variable in both data sets is `order_item_id` which uniquely identifies an ordered item. Then there are two date variables `order_date` and `delivery_date`. Each ordered item also has an `item_id` which identifies a certain kind item but is not unique. Each ordered item has a certain size, described in `item_size`. The color of an item is described in `item_color`. Each item belongs to a certain brand described in the variable `brand_id` and is valued at a price noted in the numeric variable `item_price`. Besides these order-based variables, each observation also includes information regarding the customer of this order. The data frame summary shows relevant characteristics for all variables in the known data set except for the variables `order_item_id`, `item_id`, `item_size`, `brand_id` and `item_color`. These factor variables have a very high number of unique levels and are therefore difficult to visualize.

2.2.2 Data Frame Summary

2.2.2.1 known_f

Dimensions: 100000 x 6
Duplicates: 23629

No	Variable	Stats / Values	Freqs (% of Valid)	Missing
1	order_date [Date]	min : 2016-04-01 med : 2016-10-05 max : 2017-03-31 range : 11m 30d	365 distinct values	0 (0%)
2	delivery_date [Date]	min : 1994-12-31 med : 2016-10-22 max : 2017-07-22 range : 22y 6m 22d	319 distinct values	8292 (8.29%)
3	user_title [factor]	1. Company 2. Family 3. Mr 4. Mrs 5. not reported	78 (0.1%) 375 (0.4%) 3500 (3.5%) 95976 (96.0%) 71 (0.1%)	0 (0%)
4	user_dob [Date]	min : 1900-11-21 med : 1965-01-01 max : 2012-11-12 range : 111y 11m 22d	12121 distinct values	10023 (10.02%)
5	user_state [factor]	1. Baden-Wuerttemberg 2. Bavaria 3. Berlin 4. Brandenburg [12 others]	12975 (13.0%) 14366 (14.4%) 3800 (3.8%) 2200 (2.2%) 66659 (66.7%)	0 (0%)
6	user_reg_date [Date]	min : 2015-02-17 med : 2016-02-15 max : 2017-04-01 range : 2y 1m 15d	775 distinct values	0 (0%)

2.2.3 Dependent Variable

The dependent variable in the data set is the variable `return`. If return is indicated with a 1 the item has been returned. The predictive model will aim to predict the `return` variable for the unknown data set. The data sets have been artificially balanced, meaning that approximately half (48%) of the items have been returned, as shown in table 1.

2.3 Exploratory data analysis

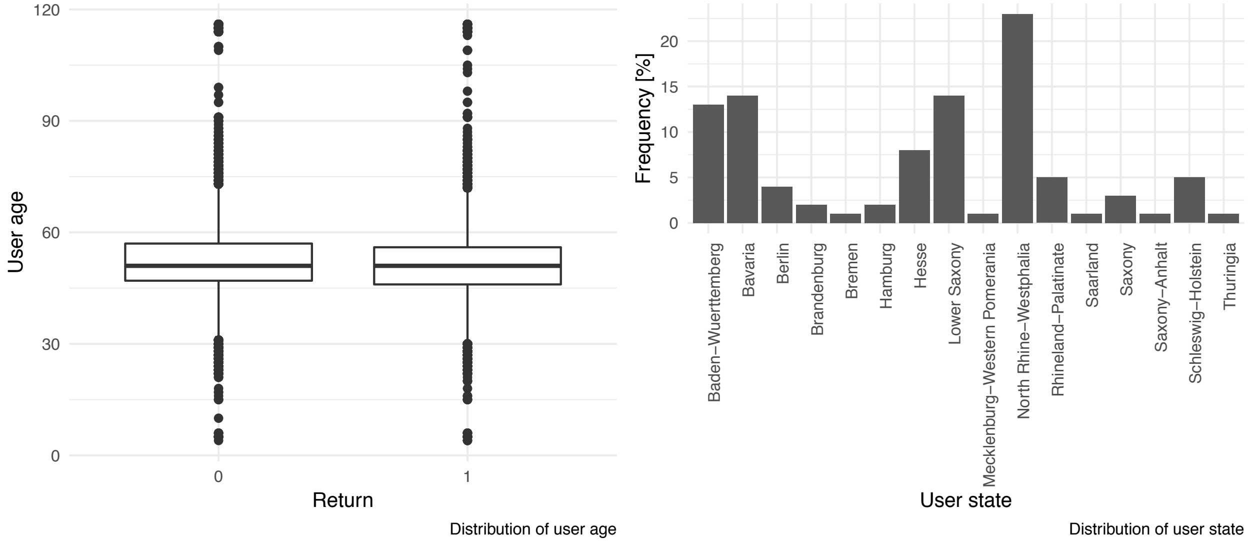
To understand the relationship between the variables and the underlying information, firstly the user-based variables will be explored. These items were ordered by 37.663 different users, meaning

Table 3: Frequency of genders

Return	Frequency [%]
Company	0
Family	0
Mr	4
Mrs	96
not reported	0

that the average user ordered 2.66 items.
With 96% percent, the majority of users are female, as can be seen in table 3.
The average user age is 52 years. However a box plot of the ages reveals that it is widespread with a minimum age of 4 years and a maximum age of 116 years. The user ages are displayed in figure 3. As people who are very young, or very old are unlikely to be ordering clothes online, these appear to be errors. Furthermore there are 5.049 customers who have not reported a date of birth and this variable therefore shows 10.023 missing values.

Figure 3 shows to which percentage users gave certain German states as their address. It can be deduced that by far the most users are from North Rhine-Whestphalia, followed by Lower Saxony, Bavaria and Baden-Wuerttemberg. This reflects the fact that these are the most heavily populated German states (Statistisches Bundesamt, 2018).



Distribution of user age

Distribution of user state

Figure 1: Demographic Data

The item-based variables hide some interesting aspects as well. Each ordered item has an order date and a delivery date. These two dates mark the length of the respective delivery time. The delivery date variable exhibits 8.292 missing values. A look at table 4 reveals, that items with a missing delivery date have never been returned. It can therefore be deduced that these items never reached their customer. This variable will play an important role in predicting returns.

Another oddity that can be found in the `delivery_date` variable is the value `31-12-1994`. There are 959 items that show this peculiar delivery date, while their order date is spread between 2016 and

Table 4: Proportion of items with missing delivery date and their return status

	Delivery date present [%]	Delivery date missing [%]
0	44	8
1	48	0

Table 5: Proportion of items with delivery date of 31-12-1994 and their return status

	Delivery date 31-12-1994 [%]	Plausibel delivery date [%]
0	51	1
1	48	0

2017. The delivery_date of `31-12-1994` is therefore clearly an error and will need to be addressed later on. Table 4 depicts items with a delivery date in 1994 and their relationship to the return variable. 32.95% of items with a delivery date in 1994 have been returned which is considerably less than the overall return ratio of approximately 50%.

The variables `item_id`, `brand_id` and `item_size` are all factor variables with many levels. Their predictive value will be determined during feature selection. It has to be stated, that the `brand_id` variable contains different scales, for example US sizes as well as European sizes. These will need to be addresses during the data cleaning process. The last item-based variable is the `item_price`.

3 Data preparation

3.1 Errors and missing values

The two data sets show some obvious errors and missing values in several variables. Table 6 shows which errors appear and how they have been handled.

The `delivery_date` variable has two important features. These are missing values and the delivery date of `31-12-1994` which yield a significant predictive power. Since the date variables will be used for feature creation and not directly included in to the model the resulting irregularities like negative delivery time will be adjusted in the respective features.
The variable `user_dob` shows some outliers that seem to be unrealistic, e.g. very young or very old people. However since it is difficult to find a clear threshold for a realistic age, these outliers will be included in the model.

Table 6: Errors and missing values

Variable	Errors	Explanation	Strategy
delivery_date	'1994-12-31'	Date contradicts respective order dates	Dummy variable
delivery_date	Missing values	-	Dummy variable
item_size	e.g. 'm' vs. 'M'	Various typos, different size scales	Correction of typos, conversion to consistent scale
item_color	e.g. 'brwon'	Various typos	Correction of typos
user_dob	e.g. '2012-11-12'	Unrealistic birth dates	Ignored
user_dob	Missing values	-	Dummy variable, replacement with median date of birth

3.2 Factor Variables

The data sets at hand include eight factor variables. Most predictive models are limited in regard to the amount of different levels a variable can have. Variables that have more levels need therefore be reduced to fewer levels if they should be included in the model. Table 7 shows the factor variables, their count of levels and how they have been treated.

Table 7: Factor levels

Variable	Level count (known / unknown)	Strategy
order_item_id	100.000 / 50.000	Not included in models
item_id	2656 / 2481	Weight of evidence
item_size	107 / 103	Weight of evidence
item_color	82 / 79	Weight of evidence
brand_id	155 / 149	Weight of evidence
user_id	37663 / 26566	Weight of evidence
user_title	5 / 5	No preparation necessary
user_state	16 / 16	No preparation necessary

3.3 Feature Engineering

3.3.1 Feature Creation

The independent variables contain complex information. In order to use this information effectively new features can be created. Table 8 shows which additional features have been created from which independent variable.

Table 8: Created Features

New feature	Based on	Comment
dob_missing	user_dob	Binary, indicates whether birth date was missing
delivery_time	order_date, delivery_date	Delivery time in days
user_age	user_dob	User age in years
user_reg_time	reg_date	Time since registration in days
delivery_date_missing	delivery_date	Binary, indicates whether delivery date was missing
delivery_date_94	delivery_date	Binary, indicates whether delivery date was in 1994
amount_per_order	user_id, order_date, item_id	Counts how often the same item has been ordered by the same person on the same day
amount_per_item	user_id, item_id	Counts how often the same item has been ordered by the same person
item_size_c	item_size	Levels conversed to one scale
unsized	item_size	Dummy variable
delivery_time_c	delivery_time	Delivery time without negatives
item_free	item_price	Dummy variable
n_orders	user_id, order_date	Amount of orders per user
woe_item_color	item_color	Weight of evidence for item colors
woe_item_size	item_size	Weight of evidence for item sizes
n_items	user_id	Amount of items a user has ordered all time
avg_item_price	user_id, item_price	Average price a user pays for an item
diff_avg_item_price	avg_item_price, item_price	Difference between item price of certain item and average price a user pays
woe_item_id	item_id	Weight of evidence for item ids
woe_brand_id	brand_id	Weight of evidence for brand ids
woe_user_id	user_id	Weight of evidence for user ids

3.3.2 Feature Selection

The variables in the data sets contain an abundance of information that could be used for creating countless variables. However not all variables will improve a predictive model. In order to choose significant features, the above variables were added to a random forest model in a step wise manner and chosen to be included in the model because each of them improved the predictive power of the random forest.

4 Model tuning and selection

In the following section the prediction model will be developed and tuned. First of, a logistic regression will be applied to estimate a possible bench mark. Then the models of a random forest and gradient boosting will be developed, tuned and compared. Predictive models can be evaluated and tuned to different metrics. A widely used metric is the area under the curve (AUC) in regard to the receiver operating characteristics curve (ROC). The AUC measures how well a model distinguishes between different classes, in this case between returned and not returned items. In the following parts it will be used to compare different models. Although the model will ultimately be evaluated in regards to costs, using the AUC to compare models is justified, because the prediction model has to pass a certain AUC threshold.

In order to build a meaningful predictive model, the known data set will be split into a training and test set. Each model will be trained on the training set and then evaluated on the test set. Afterwards the prediction will be uploaded to Kaggle and the AUC score that is calculated online for the unknown observations will be compared to the AUC that was reached on the training set. This will reveal any cases of over fitting the model.

4.1 Logistic Regression

The logistic regression performs very well for linear relationships. The logistic regression model reaches an AUC value of 74% on the test set which is quite high. This might be a symptom of over fitting. This suspicion is confirmed by training the model on the complete known data set and uploading the prediction for the unknown data set to Kaggle. The resulting score is 61% which is significantly below the AUC on the local training set. Since non-linear effects are not captured by the logistic regression, a forest based model might improve this score. In the following sections, two forest based models, random forest and gradient boosting, will be evaluated.

4.2 Random Forest

After calculating a first benchmark AUC with a logistic regression, a random forest model will be built with the same variables. The model of a random forest was chosen for its ease of use and insights it provides on feature importance. After tuning the random forest, it reached an AUC value of 73,47% for the unknown data set on Kaggle. Figure 2 illustrates the tuning process for the mtry variable in regard to out-of-bag error. As it resulted in the lowest OOB-error a mtry value of 2 was chosen.

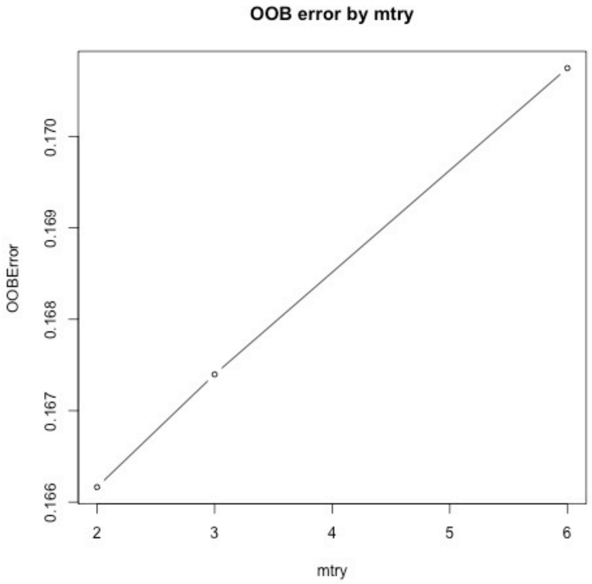


Figure 2: OOB error by mtry

4.3 Gradient Boosting

In order to select the best fitting model a third model was considered: Gradient boosting. However after building the model and tuning its parameters the AUC for the unknown data set amounts only to 68.71%. As an automatic binning function was used, manual binning might improve this score. However since the random forest model reached a considerably higher AUC value, which passed the threshold, gradient boosting was not further pursued.

4.4 Cost-Sensitivity

The predictions obtained through the random forest model are to be used to reduce the number of returns. The aim is to show customers a warning message, who are about to order an item which was predicted to be returned. However false positives and false negatives result in example dependent, asymmetrical costs, as shown in table 9.

Table 9: Cost matrix

Prediction Truth	Item kept (0)	Item returned (1)
Item kept (0) / no intervention	0	$0.5 * 5 + (-)(3 + 0.1 * itemvalue)$
Item returned (1) / warning	$0.5 * (-)itemvalue$	0

There are different options to incorporate costs into a predictive model that can be classified into either data space weighting or algorithmic adaption. Cost-sensitive learning is an active field of research and different approaches entail different advantages and disadvantages. In this paper Bayes risk theory is used to assign each sample to its lowest risk class. This is a straight forward approach which does not require algorithmic adaption. Evaluating more complex strategies like data space weighting or algorithmic adaption would go beyond the scope of this paper.

After calculating Bayes risk, each item is classified as a return item if the probability calculated by the random forest model is higher, than the Bayes risk for this product. In a test set of 40 000 items of the known data set, this leads to 19 080 items being classified as return items. The warning messages that are consequently displayed to customers lead to costs of 194 220 Euro as illustrated in table 10. These are costs that occurred because people were shown warning messages which lead to canceling their order, while they would have kept the item. This can be compared to showing no warning messages at all. Applying the same strategy to the unknown data set, 23 235 items are classified as return items. As the data set is balanced one would expect the number of returns in the unknown data set amounts to approximately 25 000 items. Since the costs for false positives are higher than for false negatives it is prudent that the prediction amounts to less than 25 000. Items with a medium high probability and a high price were rather predicted to be non returners to keep the rate of costly false positives low.

Table 10: Costs and revenue gain test set

Classification strategy	Threshold 0.5	Bayes Risk
Costs without warning messages	513 481 Euro	513 481 Euro
Costs with warning messages	200 926 Euro	194 220 Euro
Net revenue gain	312 555 Euro	319 261 Euro
Warnings displayed	20 601	19 080

5 Special modeling challenge: Analyzing and interpreting variable importance

After building and tuning a random forest model to predict returns a closer look will be taken at the importance of different variables for the return probability. There are two common measure methods for variable importance in a random forest: Out-of-bag variable importance and Gini variable importance. Figure 3 shows these two measures for the variables of the random forest. It can be deduced that the weight of evidence that was calculated for the variables `user_id` and `item_id` as well as the feature created from the `delivery_date` variable `delivery_date_missing` are the most influential variables in regard to decrease in accuracy as well as the mean decrease in node impurity.

Both measures indicate, that the weight of evidence calculated for `user_id` is by far the most important variable. This suggests that some users are more prone to order items that they will later on return than others. The variable `delivery_date_missing` had been identified as valuable during the exploratory data analysis as these items were never shipped to the customer and could therefore not be returned. Another remarkable insight that can be gained through the variable importance plots is, that certain items have a high impact on the return probability.

While the variable `delivery_date_missing` is very important for the prediction of this model in this particular scenario, it would not be possible to use it for a model that predicts return probabilities during the shopping process as the delivery_date is not known at that time.

The variable woe_user_id has to be examined critical as well. It will only be useful in future predictions, if the online shop has a high proportion of loyal customers as opposed to having many customers who buy only once, to have enough data about users return habits. The same can be said about the variable item_id which will only yield valuable predictive power if the cycle in which items are taken in and out of stock is long enough to gather information. For the data sets at hand these two assumption prove to be correct.

A remarkable difference in the two measures can be found for the variable user_state. Compared to all other variables it is the least important in regard to mean decrease in accuracy. This shows that changing the value randomly for user_state does not have a noteworthy effect on the probability accuracy. However for the Gini based mean decrease in node impurity this variable ranks quite high. This discrepancy might be explained by the fact that the model is built on continuous and discrete data. The user_state variable is a discrete variable with 16 different levels. As other discrete variables with a high level count like brand_id, item_size etc. were substituted by their weight of evidence, user_state is the discrete variable with the highest number of levels. This can be explained by the fact, that mean decrease in impurity is biased towards high cardinality variables (Parr, Turgutlu et al., 2018). Figure 4 shows exemplary the partial dependence plot for the variables item_price and n_items. The model predicts higher return probabilities for high values of variables item_price and n_items .

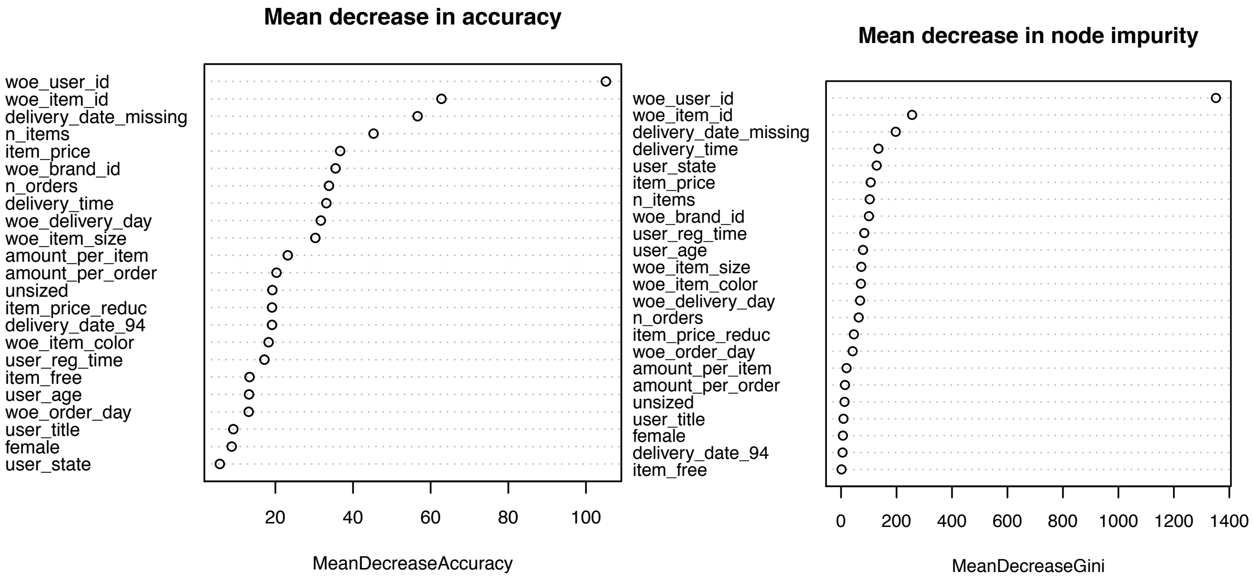


Figure 3: Variable importance

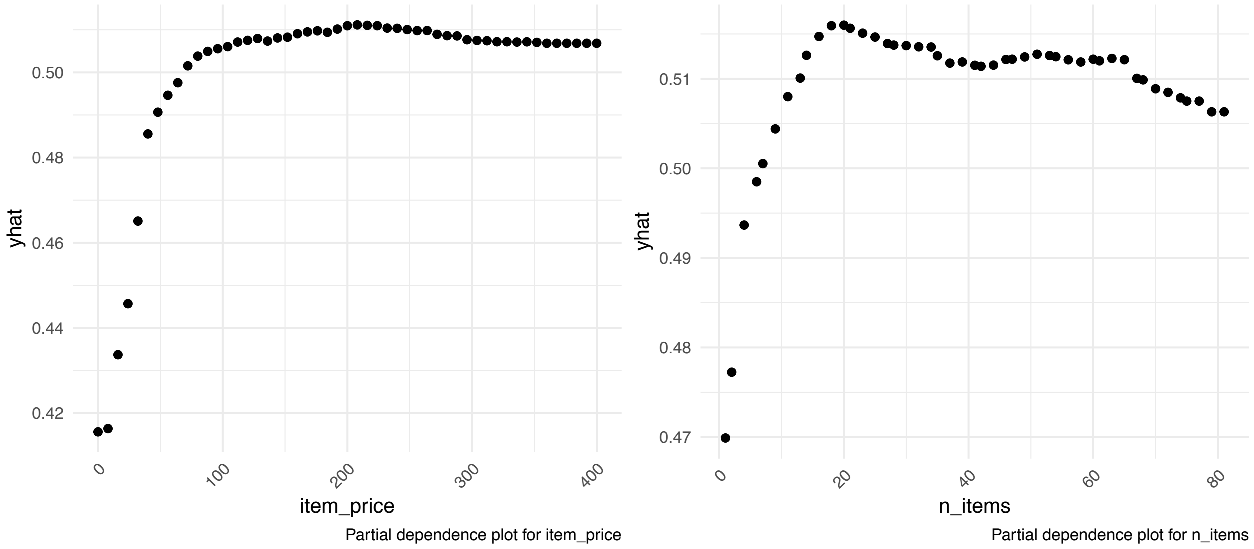


Figure 4: Partial dependence plots

6 Conclusion

This paper described the process of building a predictive model for a real-world data set of an online retailer. During the course of this paper the data cleaning process was described and exploratory data analysis was undertaken. The variables in the data sets were used for further feature creation. Then the model of a logistic regression, a random forest and gradient boosting were built, tuned and compared by their AUC value. Subsequently the random forest model was chosen because of its higher AUC value. Combined with a cost matrix and the Bayes risk strategy the random forest was used to predict which items would be returned to enable the retailer to show warning messages to customers. Finally the importance of variables for the model were inspected and critically discussed.

7 Bibliography

Data: known, unknown

Parr, Turgutlu et al., 2018: Beware Default Random Forest Importance https://explained.ai/rf-importance/index.html

Statistisches Bundesamt, 2018: https://www-genesis.destatis.de/genesis/online/data;sid=AFCEE67605FDEAF3CF312F5B9DB4C55C.GO_2_1?operation=abruftabelleBearbeiten&levelindex=1&levelid=1549976900937&auswahloperation=abruftabelleAuspraegungAuswaehlen&auswahlverzeichnis=ordnungsstruktur&auswahlziel=werteabruf&selectionname=12411-0010&auswahltext=&werteabruf=Werteabruf