ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG

information
systems

Business Intelligence

– SEMINAR WINTER SEMESTER 2013/2014 –

Data Mining with Decision Trees

– SEMINAR PAPER –

Advisor:
Prof. Dr. Dirk Neumann

Contents

1 Introduction

Decision trees are a useful technology to illustrate decision processes especially in firms. This paper tries to use decision trees in order to illustrate relationships in between a data set and to generate them out of the data with the help of a machine learning algorithm.

First of all I want to present what kind of work has been done up to now. In 2002 Podgorelec et al. recognize the importance of the growing available data and use it to observe how decision trees could help to predict diseases. Zelic et al. (1997) used decision trees too, but for medical reasons. They try to simplify the diagnose of sport injuries with decision trees and compare their results with a bayesian classifier. The decision processes in tourism are observed by Byrd and Gustke in 2007. They visualize decision processes as decision trees and try to explain how stakeholder could be regarded.

The aim of this paper is to show how rents are influenced by variables like the number of houses to rent in the near distance. Decision trees will be the method to illustrate and calculate the relationship.

In the following chapter decision trees as well as two algorithms for the generation and one for the simplification of trees will be explained in more detail. In section 3 decision trees will be calculated with the data, first with classes and afterwards with absolute values. The generated trees will also be used for the prediction of rents and they will be *pruned*. Section 4 summarizes the generated results, section 5 concludes and gives an outlook for further research.

2 Definition of decision trees

In this chapter an explication and a formal description of decision trees will be given. A decision tree is an arborescent figure which visualizes structures of objects by using their attributes. In figure 1 such an exemplary tree is visualized. Each tree starts with the *Root Node*. It is specified in a way that it has only outgoing but no incoming *branches*. Branches are the connections between two different nodes. Figure 1 shows a *binary tree*. These trees only have two outgoing branches for each node. If one starts at the root node and follows one branch one arrives at an interior node. This kind of node has incoming as well as outgoing branches. If one follows one more branch, one arrives at the *leaf*. It is a node which has only incoming but no outgoing branches.

Summarizing decision trees can be used in order to structure decisions which have been made according to some specific rules. The result is an easily comprehensible figure. This is one advantage of decision trees. Even if one does not have specific knowledge about a decision but

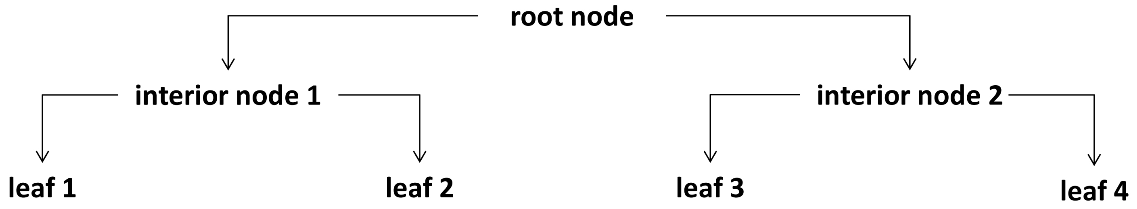

Figure 1: Exemplary binary decision tree.

the decision is represented as a decision tree, one can easily interpret it. Therefore, decision trees are often used in medicine in order to combine the observed results of a patient to get a diagnose (Bishop 2009, p.664). Another advantage is that every leaf has its own model. Regressions only have one model at all. Decision trees can be built with classifications or with regressions if the data is metric. In this paper both methods will be used. Quinlan states another advantage which becomes more and more important. It is possible to generate decision trees with huge data without high computational costs (Quinlan (1986)). The following two sections will explain two algorithms to generate decision trees.

2.1 The ID3-Algorithm

One of the first algorithms for machine learning of decision trees is Quinlan's ID3-algorithm. He specifies this approach in his paper *Induction of decision trees* (1986) where the following definition is taken from. Here, the machine learning of decision trees with classes will be defined. The way the tree is built is iterative. First a random part of the training set is chosen, the so called *window*. Now a decision tree will be generated with this part such that all elements are classified correctly with this tree. In the second step the whole training set will be classified with the generated tree. If the tree classifies all elements correctly the decision tree is generated. If not, some of the wrong classified elements are added to the window and a new tree will be built. Then again the whole training set will be classified and checked if all elements are correctly classified until there are no errors at all Quinlan (1986). states that with this approach it is possible to generate decision trees within a few iterations.

"Suppose there is a number of C objects and T is a test on an object with possible outcomes $\{O_1,O_2,...,O_w\}$. Each object in C will give one of these outcomes for T, so T produces a partition $\{C_1,C_2,...,C_w\}$ of C with C_i containing those objects having outcome O_i" (Quinlan 1986). For a better understanding this relationship is shown in figure 2. The question now is how the objects are classified into the classes. Therefore, the so called *information gain* is needed. "Let C contain p objects of class P and n objects of class N" (Quinlan 1986) and consider the following two assumptions.

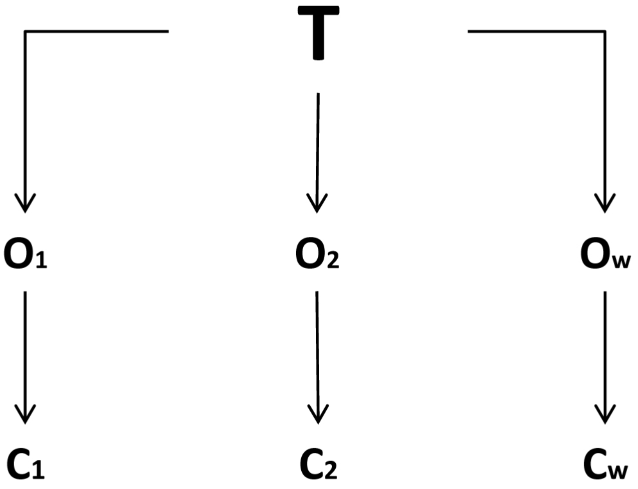

Figure 2: Visualization of the defined tree.

(1) Any correct decision tree for C will classify objects in the same proportion as their representation in C. An arbitrary object will be determined to belong to class P with probability $\frac{p}{p+n}$ and to class N with probability $\frac{n}{p+n}$.

(2) When a decision tree is used to classify an object, it returns a class. A decision tree can thus be regarded as a source of a message P or N, with the expected information needed to generate this message given by" (Quinlan 1986)

$$\mathrm{I}(p,n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}. \tag{1}$$

The binary logarithm is used because the definition is written for binary trees with two outgoing branches for every node. "If attribute A with values $\{A_1,A_2,\dots,A_w\}$ is used for the root of the decision tree, it will partition C into $\{C_1,C_2,\dots,C_w\}$ where C_i contains those objects in C that have value A_i of A. Let C_i contain p_i objects of class P and n_i of class N. The expected information required for the sub tree for C_i is $\mathrm{I}(p_i,n_i)$. The expected information required for the tree with A as root node is then obtained as the weighted average

$$\mathrm{E}(A) = \sum_{i=1}^{v}\frac{p_i+n_i}{p+n}\mathrm{I}(p_i,n_i), \tag{2}$$

where the weight for the i-th branch is the proportion of the objects in C that belong to C_i. The information gained by branching on A is therefore" (Quinlan 1986)

$$\mathrm{Gain}(A) = \mathrm{I}(p,n) - \mathrm{E}(A). \tag{3}$$

The branch one has to take is the one which contains the most information. The maximization of the information gain is at the end the same as the minimization of $\mathrm{E}(A)$ because $\mathrm{I}(p,n)$ is a constant. The algorithm validates all potential attributes A and chooses the one which maximizes the information gain. Then the tree will be generated. The same process is done for the remaining sub trees $C_1,C_2\dots,C_v$ until all objects are classified.

2.2 The CART-algorithm

In this paper decision trees will be generated with the package "rpart" of the statistical software R. It uses another algorithm, the CART algorithm, whereby the acronym CART stands for "Classification and Regression Trees". The difference is the splitting criteria. The CART algorithm does not use the information gain. Instead, it uses the *Gini-index*, which is calculated as follows (Therneau and Atkinson 2013). Choose a random object of one of C different classes with probabilities $\{p_1,p_2,\dots,p_C\}$. Now classify it to a class and use the same distribution. Then the probability of a misclassification is

$$\sum_i\sum_{j\neq i}p_ip_j = \sum_i\sum_j p_ip_j - \sum_i p_i^2 = \sum_i 1 - p_i^2. \tag{4}$$

This measurement represents the disparity and has to be minimized.

2.3 Pruning

Decision trees can become very large. *Pruning* is a method to simplify decision trees, but it is important to take into account that every simplification causes an information loss. There is an optimal degree of pruning which has to be found and which is calculated in the following way. The definition is taken from Therneau and Atkinson (2013). Define $T_1, T_2, \ldots, T_k$ as terminal nodes of a generated tree and $|T|$ as the number of terminal nodes. Then the risk of the tree is defined as

$$\mathrm{R}(T) = \sum_{i=1}^{k} P(T_i)R(T_i), \tag{5}$$

where $\mathrm{P}(T_i)$ is the probability of node i and $\mathrm{R}(T_i)$ the corresponding risk of this node. If one compares equation (5) with regression, then $|T|$ can be seen as the degrees of freedom and $\mathrm{R}(T_0)$ as the residual sum of squares. The cost of adding another variable to the model is now labeled as α, which is defined between zero and infinity. A tree with zero splits per definition has the risk $R(T_0)$. Every other tree's risk is

$$R_\alpha(T) = R(T) + \alpha |T|. \tag{6}$$

The sub tree with minimal costs is then defined as T_α. As Breiman has shown in 1996 the definitions above have the following result. "If T_1 and T_2 are sub trees of T with $R_\alpha(T_1) = R_\alpha(T_2)$, then either T_1 is a sub tree of T_2 or T_2 is a sub tree of T_1; hence either $|T_1| < |T_2|$ or $|T_2| < |T_1|$"(Therneau and Atkinson 2013). With this result it is possible to define T_α as the unique and smallest tree T for which the risk $R_\alpha(T)$ is minimized.

3 Building of decision trees

In this chapter decision trees are going to be built. Therefore the data given from the Chair of Information Systems Research of the Albert-Ludwigs-University Freiburg is used. The data set contains a rent index of the USA with rent prices per square feet in US-Dollars. The data set contains 4696 observations. The mean rent price is $5.59 per square feet and the median is $3.86, thus there seem to be some high rent prices which are the reason for the difference between mean and median. Table 1 shows the available variables in the data set to explain the rent price. As one can see the goal is to observe the effect of advertises and other variables.

Variable	Explication
craigs	Number of other renting objecting in the near distance
crimes	Number of registered crimes from 01/2013 to 08/2013
edge proportion	Proportion of vertical to horizontal edges in the picture of the advertise
height	Height of picture in pixels
hue00 – hue07	Percentage of pixels in the first until the eighth eighth of the picture
images	Number of pictures in advertise
mean blue	Average blue color
mean green	Average green color
mean red	Average red color
mean lightness	Average luminance
mean saturation	Average saturation
tweets	Number of tweets posted in distance
width	Picture width in pixels

Table 1: Available, potential explanatory variables of the data set.

3.1 Decision trees with classes

In a first step a decision tree with classes is going to be built. The data set is going to be splitted into a training- and a test set because in this paper the prediction accuracy of decision trees will also be observed. The weight is 75 % for the training set and 25 % for the test set. The tree will be generated with the training set to observe in a second step, how the generated decision tree can be used for prediction. First of all the rents have to be classified because in the original data set the rents are absolute values. After removing very high rents nine classes have been generated. The classification is shown in table 2.

Class	Rent price in US-Dollar per square-feet	Frequency
1	\$0–\$1	17
2	\$1–\$2	114
3	\$2–\$3	854
4	\$3–\$4	1625
5	\$4–\$5	1408
6	\$5–\$6	529
7	\$6–\$7	92
8	\$7–\$8	43
9	above \$8	4

Table 2: Overview about the classification.

It was not possible to use the variable average saturation because the software was not able to generate a decision tree with this variable. As a result the rent price depends on all other available explanatory variables. The result is shown in figure 5.

The resulting tree contains the number of tweets, the height of the advertise, the number of crimes and the number of houses to rent in the near distance. If there are less than 34 objects to rent in the near distance and less than 108 crimes happened, the result is class number 3 which means that the rent is between \$3 and \$4. But one can also see that there is another combination which leads to the same result. This problem can be solved by pruning the tree. This will be done later in this paper. First the prediction accuracy of the generated tree will be observed. The goal is to predict the right class. To be able to calculate the prediction accuracy a table is created. The table compares the initial with the predicted classes. The diagonal of this table then contains all true predicted classes.

$$\text{Accuracy} = \frac{\sum\limits_{i} x_{ii}}{\sum\limits_{i,j} x_{ij}}. \tag{7}$$

The accuracy is the sum of the elements where the initial class i equals the predicted class j divided by the sum of all elements. The result of the generated model is shown in table 3. The predicted classes are rows and the initial classes are columns. Class 1 is missing since the test set does not contain elements of class 1. The prediction accuracy is 21,59 %. One explication of the prediction accuracy could be the random chosen classes. By fixing the borders the frequency differs between the classes as table 2 shows. One can see that class 3 has much more elements as it would have in case of a uniform distribution of the elements. This could be the reason why class 3 is overrepresented.

Classes	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
2	0	0	0	0	0	0	0	0
3	13	144	79	20	5	0	1	0
4	8	56	206	89	25	3	1	0
5	3	19	115	249	78	9	2	1
6	0	1	3	5	26	7	4	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0

Table 3: Initial versus predicted classes.

As described before one problem of the generated tree in figure 5 is that it has many leaves and actually some of them have the same class. Hence there exist several models for one class. For example a house to rent is classified into class 3 either it has less than 34 other houses to rent in the near distance and less than 108 crimes in the near distance or there happened more than 34 crimes in the near distance and less than 80 tweets have been posted in the near distance. To solve this problem one can prune the tree. If one prunes the tree one has to evaluate between the loss of information and the gain in less complexity. This relationship is shown in figure 3. The abscissa shows the number of splits and the complexity parameter. The ordinate shows an error rate. The more splits a tree has, the smaller the error value becomes. The tree is pruned with a complexity parameter of 0.012 which maximizes the prediction accuracy and is shown in figure 6. First of all one recognizes that there are less leaves left. The unpruned tree has 12 and the pruned tree only has 6 leaves. But there are still leaves with the same class, so that the problem of several models for one class can not be solved. The next task is to check the prediction accuracy. It is calculated in the same way as above and has improved to 25.77 %. Again there is not every class represented, but the ones in the tree are those with the highest frequency of values.

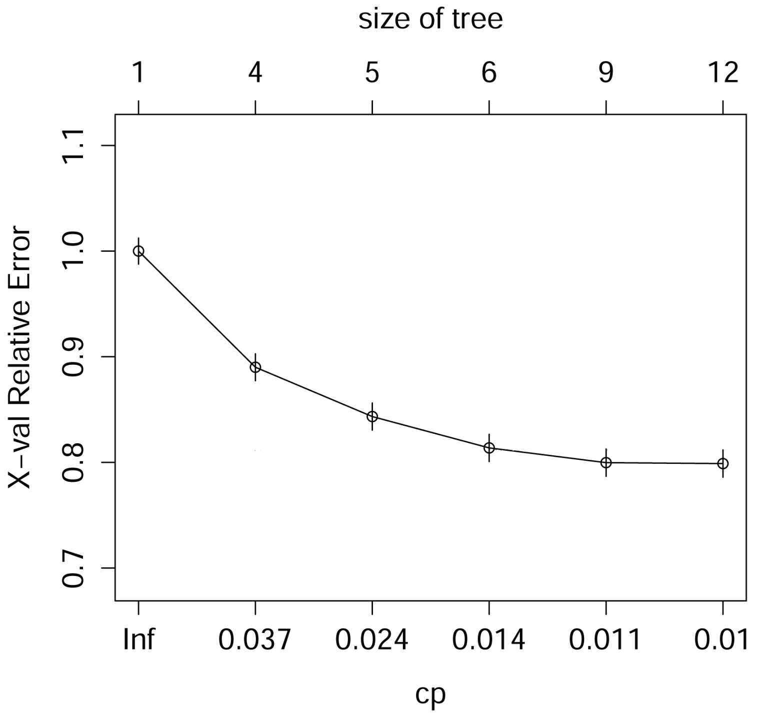

Figure 3: Trade-off between accuracy and less complexity.

To observe if the random chosen classes have an effect of the prediction accuracy the data set has been classified again. This time the aim is to get an uniform distribution between the classes. Now the boundaries are the deciles. The first decile is labeled as class 1, the rent prices in the second decile as class 2 and so on. The precise classification is represented in table 4. The result is shown in figure 7.

Class	Rent price in $ per square feet
1	$0,00–$2,50
2	$2,50–$2,95
3	$2,95–$3,33
4	$3,33–$3,58
5	$3,58–$3,86
6	$3,86–$4,12
7	$4,12–$4,36
8	$4,36–$4,70
9	$4,70–$5,26
10	über $5,26

Table 4: Classes with deciles as boundaries.

In this tree every class only appears in one leaf. In comparison to the generated tree with random chosen classes the used variables in the tree have changed. As before the houses to rent in the near distance, the number of crimes in the near distance, the number of tweets in the near distance and the size of the advertise are represented in the final tree. Additionally the proportion of vertical to horizontal edges in the picture of the advertise and the average blue color are represented in the built tree. The prediction accuracy of this tree is 24.23 %. In order to improve the prediction accuracy it is possible to prune this tree too. Figure 8 shows the pruned tree with a complexity parameter of 0.012, which again maximizes the prediction accuracy. The pruned tree has only 6 leaves left but the prediction accuracy is unchanged. To put it in a nutshell pruning has never reduced the prediction accuracy. In one case it increased. This result has to be compared with the results for regression trees.

3.2 Regression trees

Subsection 3.1 illustrates the generation of decision trees with classified data as well as their use for prediction. Decision trees can also be built with absolute metric values. By using absolute values the decision tree is generated through a regression which uses the analysis of variances as the splitting criteria. As it is done in the section above the data set is split into a training and a test set. Very high rents have been removed again. The resulting tree is too large to illustrate it. The reason is that the data is *overfitted*. Figure 4 illustrates this. The value of the x-error is increasing with every split. This denotes that the prediction accuracy decreases with every further node of the tree. The best tree for such a situation is a tree with the root node only. Varian (2013) describes this problem. If there are *n*-equations it would be useful to have also *n*-independent variables to solve these equations. However such a model only performs well within the given data set. To solve the problem the rents have been logarithmized. As figure 9 and figure 10 show logarithmizing reduces the margin between the smallest and the highest value.[1] The resulting tree is shown in figure 11. The tree has 6 leaves and only the number of crimes, the number of houses to rent and the number of tweets in the near distance are used in the model. Again the next step is the prediction. It is not possible to use a table as a measure of the prediction accuracy for regression trees. Instead one has to calculate the deviation

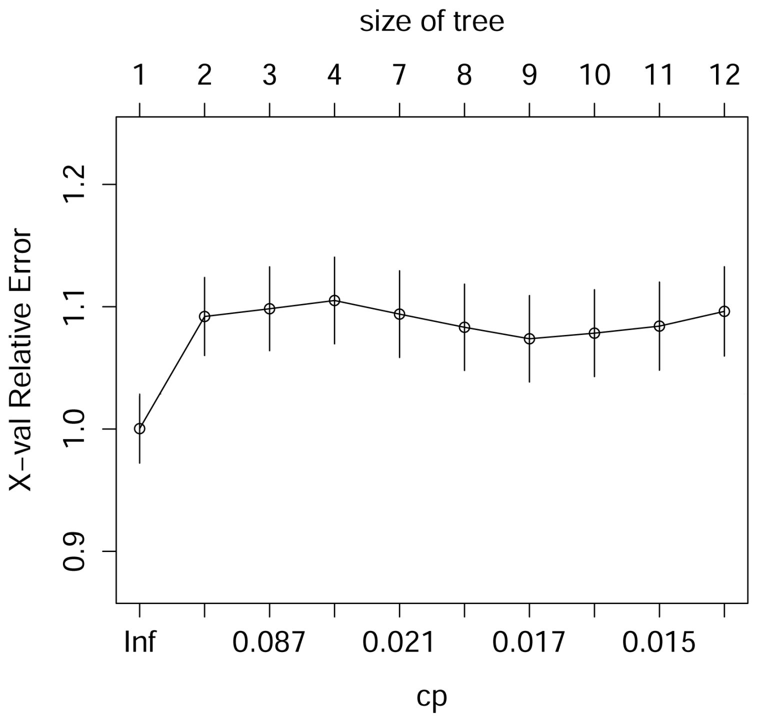

Figure 4: Graphical illustration of overfitting.

[1] It only works if the whole data set is used. Logarithmizing the one with the removed outliers does not solve the overfitting problem.

between the predicted and the observed values. Two measures to calculate this deviation are the Root-Mean-Squared-Error (RMSE) and the Mean-Absolute-Error (MAE)

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(obs_i - pred_i)^2}, \tag{8}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|obs_i - pred_i|\,. \tag{9}$$

The observed value is obs_i and the predicted value is $pred_i$. The closer both measures are to zero the better is the prediction. The tree shown in figure 11 has a RMSE of 0.3009 and a MAE of 0.2069 and can be pruned. This time a complexity parameter of 0.0125 is used. The result is shown in figure 12. It has only 5 leaves with an RMSE of 0.2992 and a MAE of 0.2064.[2]

4 Results

In this chapter a brief summary of the results will be given. Table 5 summarizes all generated trees with the used features and the results for prediction. The first generated tree with random classes has 12 leaves and a prediction accuracy of 21.59 %. The pruned tree with a complexity parameter of 0.012 has an increased prediction accuracy of 25.77 % and only 6 leaves left. In a further step the influence of the random classes is observed. A decision tree with decile classes ends up with 7 leaves and a prediction accuracy of 24.23 %. Pruning with a complexity parameter of again 0.012 reduces the number of leaves to 6 with the same prediction accuracy. All in all the trees with the uniform distributed classes performed better than those with random classes. The second part of this paper was to generate regression trees. It was necessary to logarithmize the rents to be able to generate a decision tree. The resulting tree has 6 leaves with an RMSE of 0.3009 and an MAE of 0.2069. After pruning with a complexity parameter of 0.0125, a tree with 5 leaves and decreased RMSE (0.2992) and MAE (0.2064) results. The results illustrate that pruning has no negative impact on the prediction accuracy, in the majority of cases the effect is positive.

[2] It was not possible to calculate the Mean-Absolute-Percentage-Error.

Tree	log	cp	leaves	Accuracy	RMSE	MAE	splitting method	in figure
Decision tree with random classes	no		12	21.59 %			Class	5
Pruned tree with random classes	no	0.012	6	25.77 %			Class	6
Decision tree with decile classes	no		7	24.23 %			Class	7
Pruned tree with decile classes	no	0.012	6	24.23 %			Class	8
Regression tree with logarithmized rents	yes		6		0.3009	0.2069	ANOVA	11
Pruned tree with logarithmized rents	yes	0.0125	5		0.2992	0.2064	ANOVA	12

Table 5: Summary of the generated trees.

5 Conclusion

This paper had the aim of illustrating decision trees as a machine learning technique by classifying rent prices. In a first step decision trees with classes have been generated. Afterwards decision trees with absolute values have been built. Because of the fact that rents are absolute values, the last generated tree is preferable. The rents are logarithmized and the decision tree is pruned with a complexity parameter of 0.0125. The error measures RMSE and MAE are smaller in comparison to the unpruned tree.

All in all decision trees are an useful method to explain rents. But it is possible to show that the used explanatory variables in the model depend on the classification method. An advantage is the clarity of the results. Another important result is that pruned trees do not predict worser than unpruned trees as stated before in section 4. A problem was overfitting, but it could be solved with logarithmizing.

The easy comprehensibility for non-specialist persons as well as the low computational costs are reasons for further research. Especially in a time period where the amount of data grows heavily, decision trees can be a huge advantage. *Random Forest* could be one very important point because this technology combines individual decision trees and is one possibility to analyze large amounts of data efficiently.

A Figures

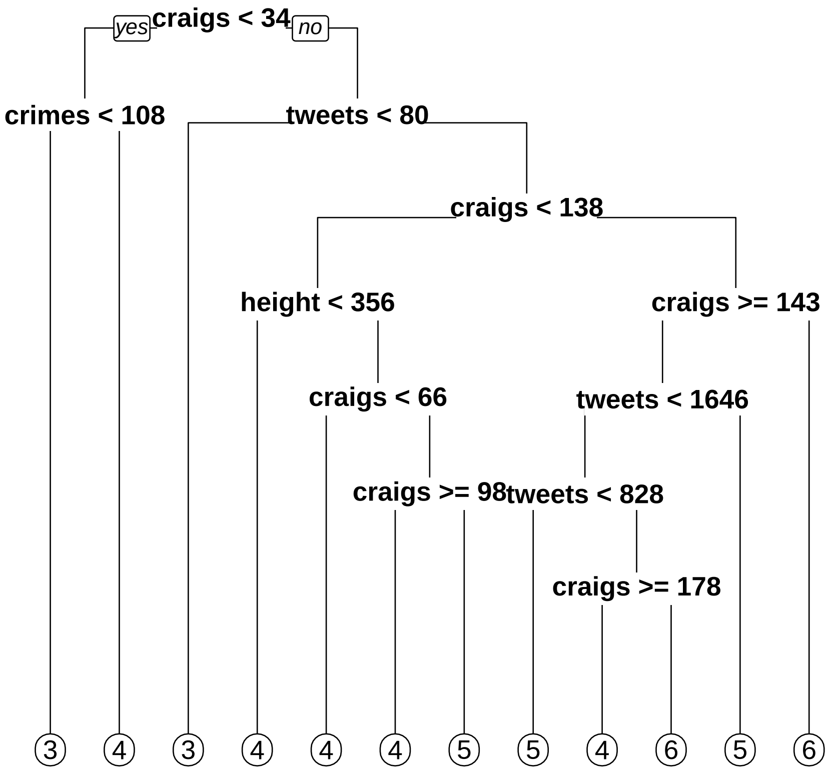

Figure 5: Decision tree with random classes.

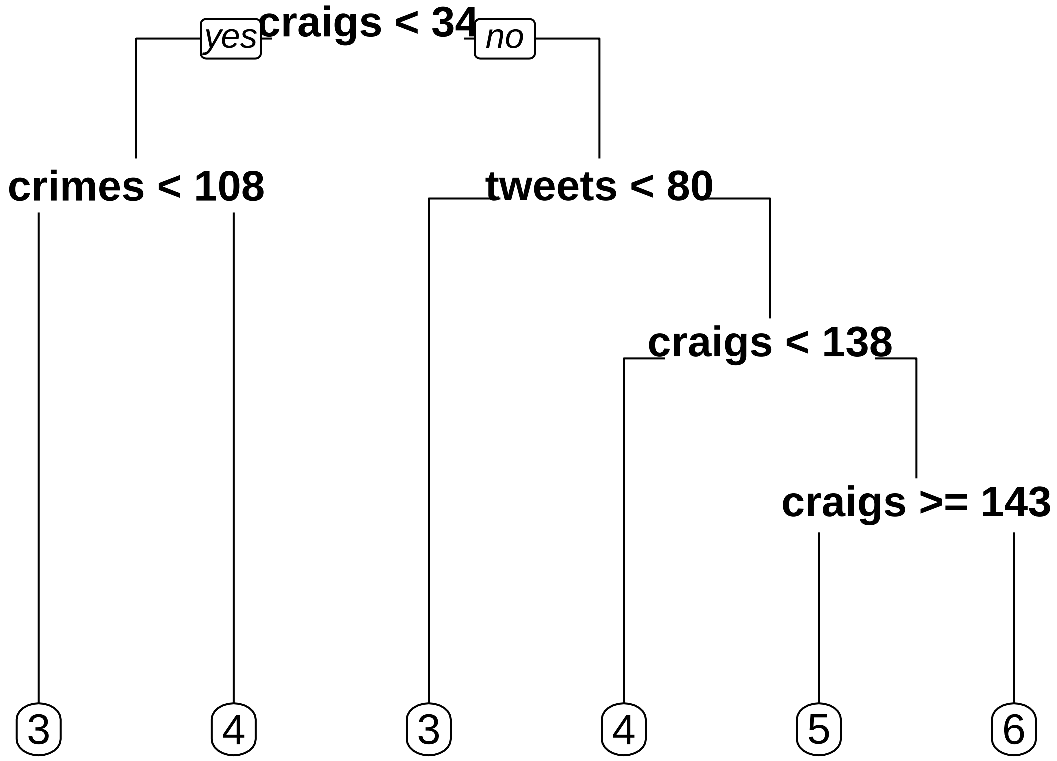

Figure 6: Pruned decision tree with random classes and a complexity parameter of 0.012.

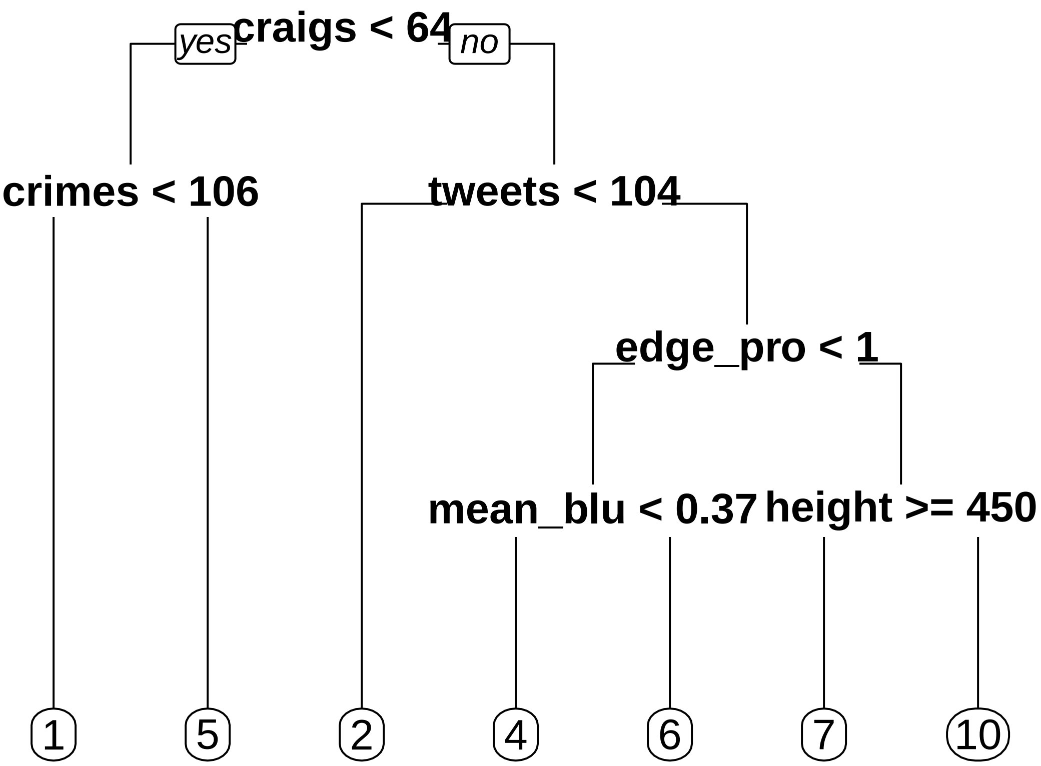

Figure 7: Decision tree with decile classes.

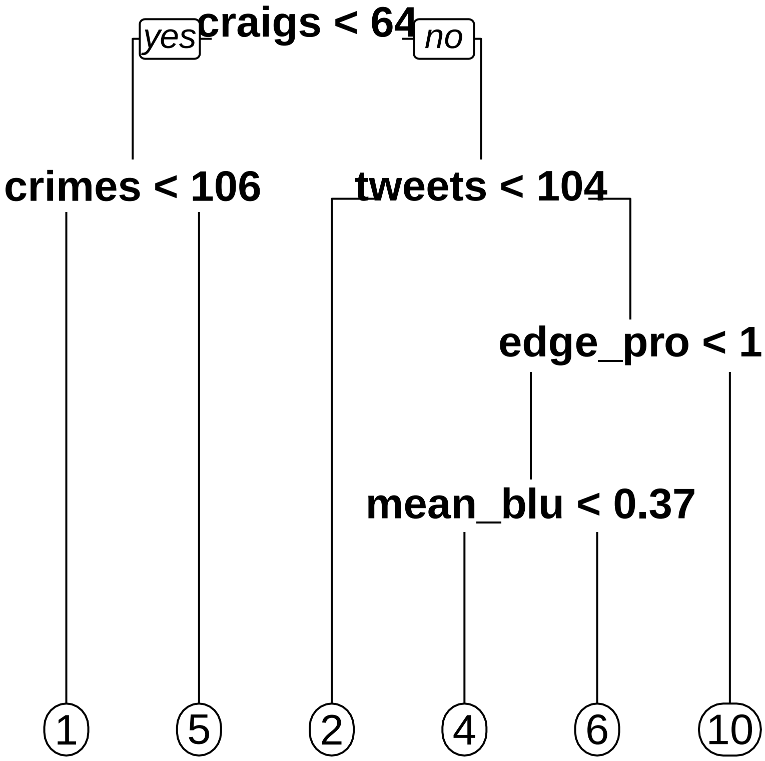

Figure 8: Pruned decision tree with decile classes and a complexity parameter of 0.012.

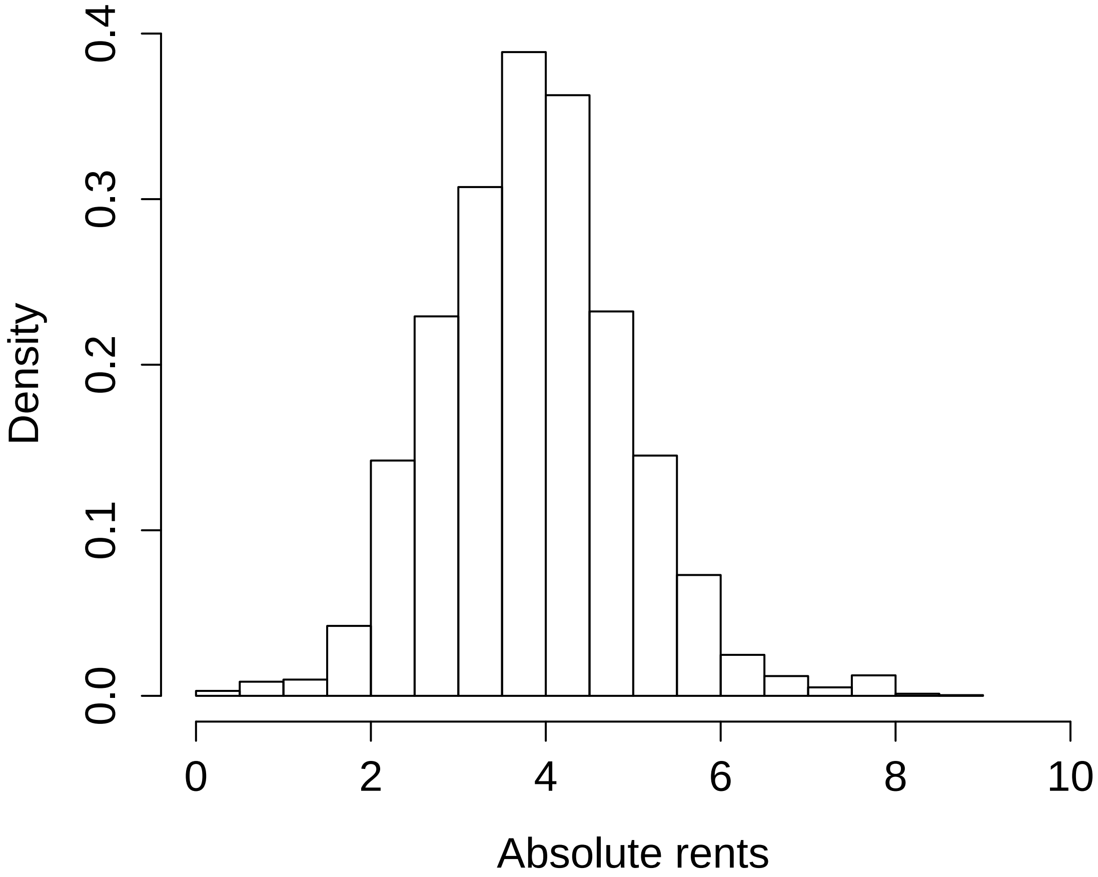

Figure 9: Histogram of the absolute rents.

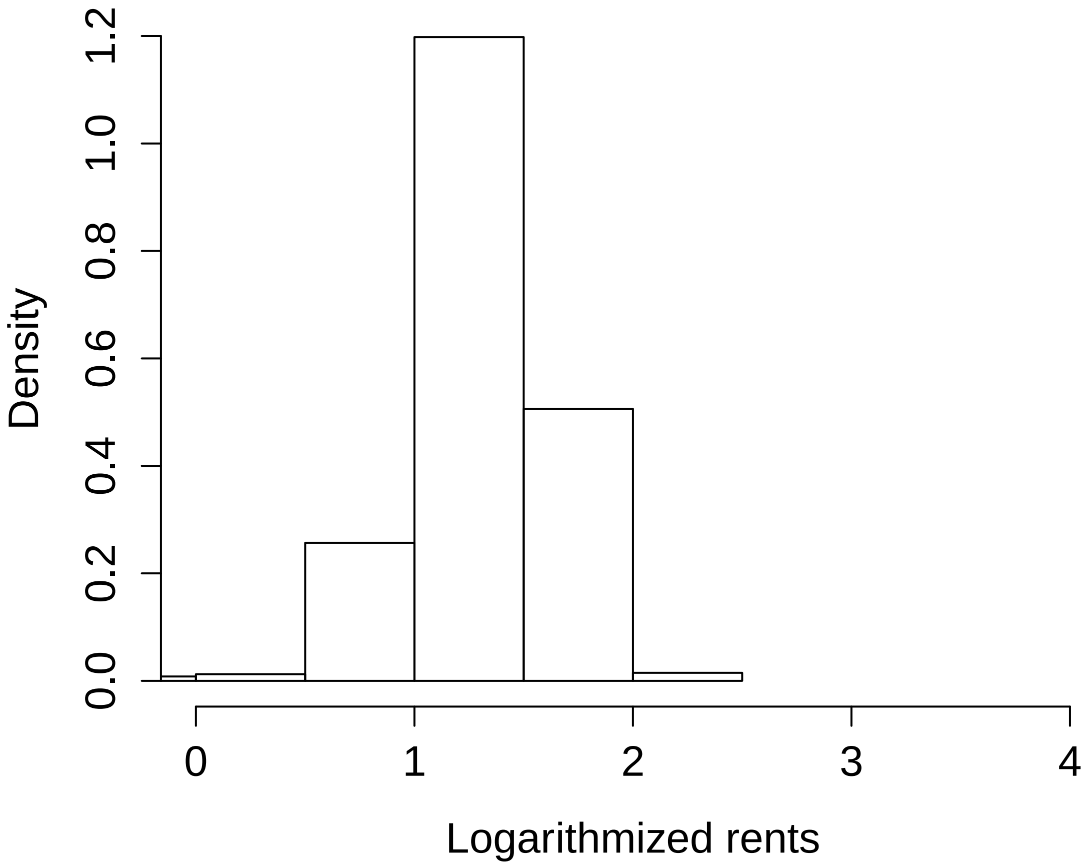

Figure 10: Histogram of the logarithmized rents.

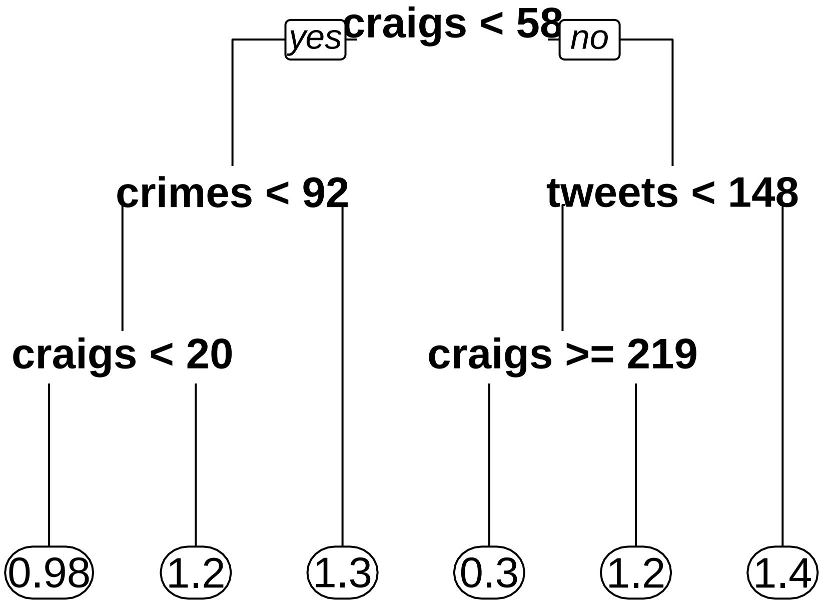

Figure 11: Regression tree with logarithmized rents.

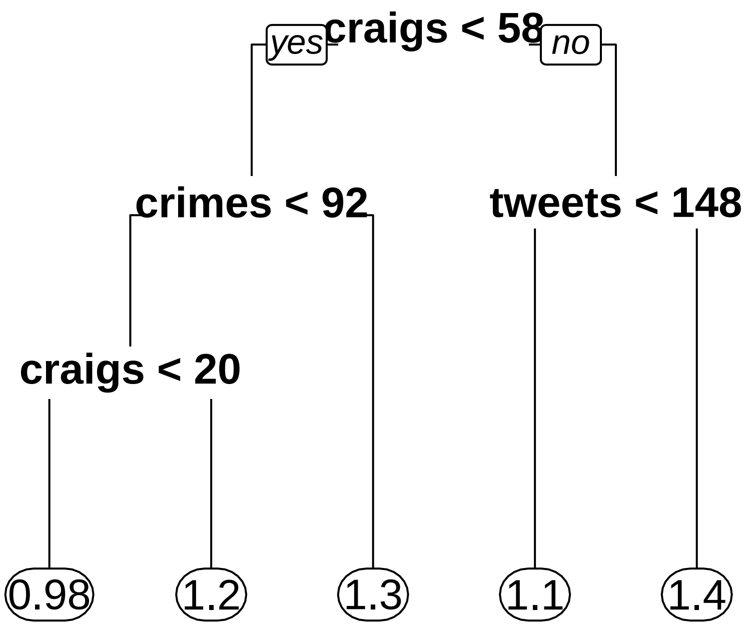

Figure 12: Pruned regression tree with logarithmized rents and a complexity parameter of 0.0125.

B References

Bishop, Christopher M. (2009). *Pattern recognition and machine learning*. Corr. at 8. printing. Information science and statistics. New York and NY: Springer. isbn: 0387310738.

Breiman, Leo (1996). *Technical Note: Some Properties of Splitting Criteria*. In: *Machine learning*, Vol. 24, No. 1, pp. 41–47.

Byrd, Erick T. and Larry Gustke (2007). *Using decision trees to identify tourism stakeholders: The case of two Eastern North Carolina counties*. In: *Tourism & Hospitality Research*, Vol. 7, No. 3/4, pp. 176–193. issn: 14673584.

Podgorelec, Vili et al. (2002). *Decision Trees: An Overview and Their Use in Medicine*. In: *Journal of Medical Systems*, Vol. 26, No. 5, pp. 445–463. issn: 0148-5598.

Quinlan, J. Ross (1986). *Induction of decision trees*. In: *Machine learning*, Vol. 1, No. 1, pp. 81–106.

Therneau, Terry M. and Elizabeth J. Atkinson (2013). *An Introduction to Recursive Partitioning Using the RPART Routines*. In:

Varian, Hal R. (2013). *Big Data: New Tricks for Econometrics*. In:

Zelic, Igor et al. (1997). *Induction of Decision Trees and Bayesian Classification Applied to Diagnosis of Sport Injuries*. In: *Journal of Medical Systems*, Vol. 21, No. 6, pp. 429–444. issn: 0148-5598.

C List of Figures

D List of Tables