DATA SCIENCE EDUCATION IN SECONDARY SCHOOLS: TEACHING AND LEARNING DECISION TREES WITH CODAP AND JUPYTER NOTEBOOKS AS AN EXAMPLE OF INTEGRATING MACHINE LEARNING INTO STATISTICS EDUCATION

Rolf Biehler, Yannik Fleischer, Lea Budde, Daniel Frischemeier, Dietrich Gerstenberger, Susanne Podworny, Carsten Schulte
Universität Paderborn, Germany
yanflei@math.uni-paderborn.de

In the setting of design-based research, the second version of an experimental course on data science is implemented accompanied by research. The three modules of the course focus on "data and data detectives", "machine learning" and a combination of both in working on a final project. In this paper, we will focus on the topic "decision trees" which is part of "machine learning". The students learn approaches of how to build decision trees manually from data using the tree plugin of CODAP. Further on, they learn to design and code an algorithm with Python that automatically generates trees. Afterwards, the algorithm is applied to real data sets with the support of Jupyter Notebooks. The instructional approach provides a deep content knowledge, which also serves as a basis for discussing the difference between humans' and machines' building decision trees and the societal implications of implementing them in practice.

INTRODUCTION

Mathematics is both part of our world and hidden in it (Heymann, 2003). In today's world, such hidden mathematics is constitutive for numerous applications of automated decision-making processes. Applications such as suggestions of content or advertisements on online platforms, diagnoses in healthcare, evaluation of legal issues (Dressel & Farid, 2018) or even support for election campaigns (Issenberg, 2012) are implemented through automatised decision-making systems. The use of automated decision-making is accompanied by a social discourse, which is partially characterised by euphoric exaggeration of the performance and the value on the one hand and anxious rejection of so-called "Frankenstein algorithms" on the other (sueddeutsche.de, 5.9.18). Decision trees represent one kind of model in the field of machine learning. With an increasing societal relevance, there is a growing demand for data-driven procedures to be taken up in school education (Biehler & Schulte, 2018; Engel, 2017; Ridgway, 2016). Automated decision models are linked to everyday experiences of students (e.g. personalised advertising) and dealing with underlying algorithms and mathematics allows for a re-evaluation and well-founded reflection on its opportunities and limitations in different contexts. This corresponds to the motif of "world orientation" in general mathematics education (Heymann, 2003) and can be helpful for people to understand the digitalised world and for an informed participation in a discourse on machine learning. Therefore, we created and implemented a teaching module on machine learning for upper secondary level in the setting of design-based research. In this paper, we present the part of this machine learning module about the method of decision trees. The target is to show our elementarisation of the content and to give insights in the practical implementation. An outlook on the accompanying research about potential obstacles to understanding is given as well.

PROJEKT DATA SCIENCE UND BIG DATA IN DER SCHULE (PRODABI)

The teaching module that we present in the following was created in the context of ProDaBi project (www.prodabi.de) which is an interdisciplinary joint project of didactics of mathematics and didactics of computer science at the University of Paderborn. The project was initiated by the Deutsche Telekom Stiftung and has been funded since 2017. One aim of the project is to develop and implement a data science curriculum, initially in the upper secondary level, and then for the lower secondary level in the form of a system of stand-alone modules. The project began with hosting an international symposium. Based on this symposium, a theoretical foundation for curricular goals and contents was developed (Biehler et al., 2018). Practical testing and evaluation takes place in a so-called project course (non-obligatory course in grade 12), which is organised in cooperation with two Paderborn upper secondary schools for the second time in the 2019/20 school year. The first module deals with the basics of statistics and data exploration, furthermore basic knowledge of the

In: P Arnold (Ed.), *New Skills in the Changing World of Statistics Education*
Proceedings of the Roundtable conference of the International Association for Statistical Education (IASE), July 2020, Online. Voorborg, The Netherlands.

programming language Python and relevant libraries for data science such as *pandas* (McKinney, 2010). The second module deals with machine learning focusing on two methods: decision trees as representatives of transparent models and artificial neural networks as representatives of less transparent" black box" models. The third module consists of project work in which the students apply their previously acquired skills to an authentic question with real data (prediction of free parking spaces). In this paper, the focus is on the teaching module about decision trees.

THE MACHINE LEARNING METHOD OF DECISION TREES

Decision trees are classification models that predict a target variable from other variables (e.g. predicting a disease from medical features). The hierarchically organized decision rules of a decision tree can be displayed in a directed tree structure. Therefore, the decision making of a tree model is highly transparent and understandable. In terms of machine learning, decision trees are generated from a (training) data set where the target variable is known as well as a set of predictor variables. Self-learning algorithms such as ID3 (Quinlan, 1986) use training data to examine and compare different predictor variables in order to find the best ones suited to predict the target variable. In a greedy manner, the variable with the best rating is placed at the top of the tree and recursively, further variables are selected for the next levels of the tree structure until the data set is perfectly classified or no variable is left.

The key task in this approach is to assess how well a predictor variable is suited to predict the target variable. We discuss the case of a categorical target variable. To assess a predictor variable, the data set is split with respect to the different values of the predictor variable into a complete set of subsets. In each subset we find a frequency distribution of the values of the target variable. As prediction we choose the value of the highest frequency. The frequency distributions in these subsets thus provide information on how many cases are misclassified, i. e. how well the predictor variable is suited to predict the target variable. These frequency distributions can be assessed with different measures such as the misclassification error or the entropy function from information theory (Topsøe, 1974). After assessment of all data splits, the split with the highest measure is chosen and the procedure is repeated recursively regarding all subsets. A decision tree typically suffers from overfitting, which affects performance in predicting new data. Overfitting means that specific characteristics of the training data are learned. This is because the subsets become smaller over different steps of the creation process so that the resulting frequency distributions become less generalisable and more specific to the concrete data set. Therefore, the tree is pruned retrospectively with regard to test data so that splits with a negative influence on the performance are taken back. Decision trees are well-performing classification models that form the basis for even more sophisticated methods as random forests (Hastie et al., 2009). Moreover, decision trees are intuitively understandable representations of complex data sets whose decision-making is transparent.

THE DESIGN OF THE DECISION TREE MODULE

When designing the module, we had to decide, which computational tools to use, which data for introducing and applying decision trees to and how to elementarise and organise the stepwise learning process towards the automatic decision tree algorithm and its practical use in modelling and deployment processes. In the coursework before the topic of decision trees, the students had learned to use CODAP for an initial exploring a data set on media use of students (JIM-PB). Then the students were introduced to the programming language Python and to its use with Jupyter Notebook. Python was than introduced as a tool for data exploration with more comprehensive and flexible features than CODAP. In the decision tree module, we can utilise the acquired competences with these tools. The suitability of the tools for the topic decision trees is explained and exemplified below. We designed a module with a basic part where students derive the systematic approach of a decision tree algorithm without programming, an optional advanced part where the students implement parts of the algorithm themselves, and an application part where students apply the algorithm to new data. The application part is realisable in two different forms, with or without programming.

THE DATA

In the whole project course, we use different data sets which represent different kinds of data. Among some small data examples, we mainly regard three data sets:

- Data about parking spaces in Paderborn (raw data, time-series data)
- MNIST data set (LeCun & Cortes, 1998) (tidy data, image data)
- JIM-PB data on the media use of adolescents (tidy data, survey data)

All three data sets are used in context of decision trees. The JIM-PB data set is a self-collected data set on the media use of young people from different schools in Paderborn (n = 215, 94 variables)

index	Playing_OnlineGames	Gender	Using_Twitter	Using_Snapchat	Using_Instagram	Youtube_..._MusicClips	Youtube_LetsPlay	Youtube_..._FunnyClips	Youtube_..._SportClips	Youtube_FashionBeauty	Own_Tablet	Own_Computer	Own_GameConsole	Own_E_Reader
1	frequently	male	rarely	rarely	frequently	frequently	frequent...	rarely	rarely	rarely	True	False	True	False
2	frequently	male	rarely	frequently	frequently	frequently	frequent...	frequently	frequently	rarely	False	True	False	False
3	frequently	male	frequently	frequently	rarely	frequently	frequent...	frequently	frequently	rarely	False	False	True	True
4	rarely	female	rarely	rarely	frequently	rarely	rarely	rarely	rarely	rarely	False	False	False	False
5	rarely	male	rarely	frequently	frequently	rarely	rarely	rarely	frequently	rarely	False	False	True	False
6	rarely	female	rarely	frequently	frequently	frequently	rarely	rarely	rarely	rarely	False	False	False	False
7	frequently	male	rarely	frequently	frequently	frequently	frequent...	frequently	rarely	rarely	False	True	True	False
8	frequently	female	rarely	frequently	frequently	frequently	rarely	rarely	rarely	rarely	True	False	True	True
9	rarely	female	frequently	frequently	frequently	rarely	rarely	frequently	rarely	rarely	False	False	False	True

Fig. 1: Extract of the JIM-PB data set (didactically reduced form, n = 53, 15 variables)

The questionnaire is based on the JIM study (Behrens & Rathgeb, 2017) and was enriched with additional questions. For some variables (e.g. Playing_OnlineGames, Using_Instagram, etc.) the frequency of use is asked in a seven-tiered scale (never - daily), other characteristics (e.g. Own_Computer) are queried in a binary scale (true, false) and the remaining characteristics (e.g. Number_of_Apps) require numerical input. For better usability of tools, we prepared a didactically reduced form of the data in which all scales are transferred to binary scales (see Fig. 1). Considering non-binary categorical variables in CODAP is possible, but the display of these in the tree tool is not easy to understand for beginners because the tree tool creates binary data splits whether the variable is binary or not.

DIGITAL TOOLS

In the decision tree module of the project course we mainly use two different digital tools: the web-based data analysis tool CODAP (Finzer, 2017) and Jupyter Notebook (Toomey, 2017). CODAP includes an interactive decision tree plug-in (Engel et al., 2018) which is shown in figure 2. With this tool, it is possible to manually create a decision tree for a data set and to dynamically show statistical measures like the misclassification error. The illustration in figure 2 is based on the reduced JIM-PB data set from figure 1. The 15 variables are shown at the bottom. The target variable is "Playing_OnlineGames" with the values "frequently" or "rarely". The user can select the listed predictor variables from the bottom section by drag and drop. The tool automatically splits the data set with respect to the predictor variable and evaluates the resulting subsets.

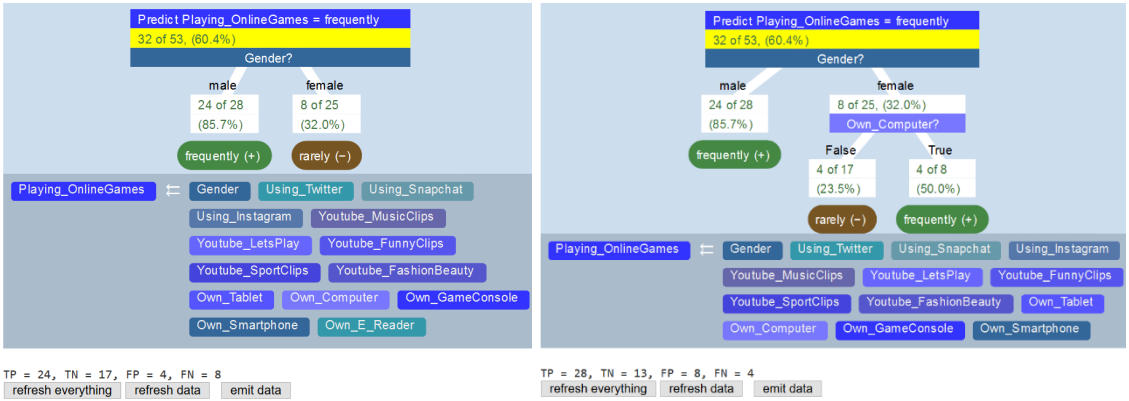

Fig. 2: CODAP Decision Tree Plug-In

In figure 2 (left), the predictor variable "gender" is selected. The illustration shows that out of 28 male persons, 24 frequently play online games. Out of 25 female persons, only eight play frequently. The resulting decision tree classifies according to the majority principle and predicts that male persons play online games frequently and female persons play rarely. To improve the tree, the

user can add additional features at the end of the branches (see figure 1, right). That way, a decision tree grows step by step. The user can evaluate the quality of the tree with regard to the data set by the displayed numbers of True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN). From these values, the misclassification error and other statistical quality criteria, such as sensitivity or specificity can be calculated. The CODAP decision tree tool has different limits. Only binary splits are possible, and it is not possible to automatically generate optimal trees. However, these limits are not negative as a starter. The handling of the tool is intuitive and after getting used to it, it is suitable to explore different ways of generating decision trees. Students can test predictor variables that they intuitively judge as relevant or that they found as relevant in their previous exploration of the data set. For further exploration, we recommend the reader to visit (https://codap.concord.org/releases/latest/static/dg/en/cert/index.html#shared=117414).

The tool Jupyter Notebook (www.jupyter.org) is an interactive cell-based programming environment, which we use with the programming language Python with its powerful libraries such as pandas. However, Juypter Notebook is not only a programming environment, but more like a teaching tool with cells for code and cells for explanatory texts and pictures. The tool is perfectly suitable for interactive tutorials where outputs can be displayed directly under a code cell. That way, code can easily be edited and repeatedly rerun in an interplay with regarding explanatory elements of the notebook. Additionally, it is possible to hide code and create interactive widgets in which a user can vary parameters and run implemented algorithms without explicitly using Python code. That way, Jupyter Notebook can be used as teaching tool also for students with no coding competence. For example, we developed a small decision tree library for teaching purpose, which is integrated in a Jupyter Notebook through a graphical user interface.

ELEMENTARISATION OF THE TEACHING MODULE

Within the ProDaBi project, we have developed a teaching module on machine learning with decision trees, which was implemented in the recent project course. The elementarisation of the content is based on Fleischer (2019). The resulting module consists of 12 lessons, and the content structure is as follows:

1. Introduction to the concept of data split *(1 lesson)*
2. Development of an algorithmic approach by manual creation of decision trees with CODAP *(3 lessons)*
    - Key Task: Comparison of different data splits based on misclassification error to assess which predictor variable is most suitable
    - Formulation of an algorithm for systematically creating an optimal (good) decision tree on training data (expressed as pseudocode)
3. Introduction of the entropy measure in comparison to the misclassification error as a quality measure *(2 lessons)*
4. Implementation of an algorithm with Python based on the developed pseudo code *(3 lesson)*
5. Applying the decision tree algorithm to data *(3 lessons)*
    - Expansion of the algorithm through post-pruning with test data to reduce overfitting
    - Creation, analysis and evaluation of resulting decision trees
6. Critical reflection upon the deployment of decision tree models in different scenarios *(1 lesson)*

The module includes a basic part without programming (1. and 2.), an advanced part that requires knowledge in the programming language Python (3. and 4.), and a final application part (5. and 6.) which can be implemented with or without programming. In the basic part, the students explore the process of creating a decision tree with the CODAP decision tree tool. An application context of such a tree is the personalised advertising in online platforms because such platforms own similar personal data and advertise for different products like online games. With specific tasks, the students work on different aspects of how to choose the next predictor variable and of evaluating a decision tree with statistical criteria. For example, the students formulate rules for the decision-making in the leaves based on the conditional frequency distributions and rules for the comparison of two different splits. This provides the basis for systematically creating a decision tree. As a next step, the students formulate - in group work - an algorithm in pseudo code language based on their actions when constructing a tree. In the advanced part of the module, students implement parts of such an algorithm that automatically creates decision trees from data sets for a given target variable. For this

purpose, we use Python in Jupyter Notebooks. Leading to a performant algorithm, the entropy function is introduced for measuring the quality of a split. Different advantages of the entropy are exemplified with the given data. The students implement the algorithm in division of labor, so that different groups each implement one of three functions: first defining all possible splits, second identifying the best split according to the entropy measure, and third dividing a data set for a given split. Finally, the three functions are merged in a recursive decision tree algorithm.
In the application part of the module, the students work with our decision tree library, which uses parts of their own coding. We integrated that into a special Jupyter Notebook with a graphical user interface so that no coding is needed. The resulting Jupyter Notebook is used like a tool to automatically create trees and evaluate the resulting models with test data. That leads to the necessity of pruning. Therefore, the students first manually prune single nodes and after understanding the algorithmic approach of post-pruning, it is also automated. The resulting pruned decision trees usually perform well on training data and also on test data. As next step, different models are interpreted and critically reflected. In class, for example, the algorithm provided the decision tree model shown in figure 3 for predicting gender based on a data set that contains data about 200 persons and 94 characteristics of their media behavior. In that case numeric scales were used, where 1 means "never" and 7 means "daily". The white nodes of the tree contain split criteria and the yellow nodes contain the classification value at the end of each branch. In every node of the decision tree the (conditional) frequency distribution of the target variable is displayed, in which the left value represents the number of females and the right value represents the number of males.

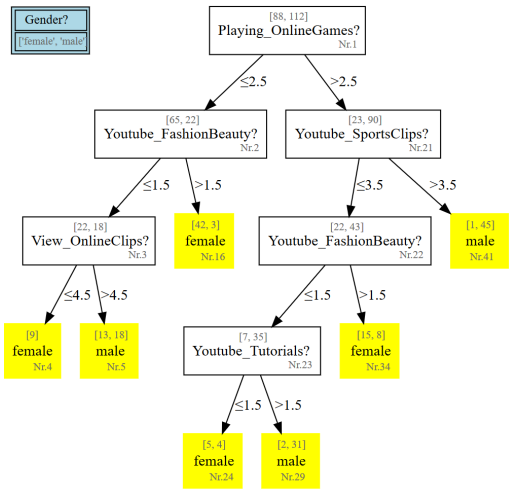

Fig. 3: Decision tree resulting from JIM-PB data that predicts the gender from media use

For example, the tree classifies people as male if they frequently play online games and watch sports videos and as female if they rarely play online games and frequently watch fashion videos. Test data is classified correctly to 85 percent with this decision tree model. Such an example allows for a reflection on automated decision making with regards to different aspects. On the one hand, the rules of decision-making are questioned content-wise. On the other hand, the accuracy of such automated decision models are examined by way of different statistical measures. Furthermore, the recent research of fairness in machine learning is adopted which uses statistical measures to evaluate how fair a decision model is in terms of sensitive attributes. In total, the students get qualified to evaluate a decision tree model statistically and to reflect about social implications of possible applications. The module is designed to use the basic part and the application part which treat the systematic approach and the statistical evaluation for contexts without programming.

OUTLOOK

In our accompanying research, we analyse students' work and assess students' understanding by a comprehension test. In the comprehension test the students understanding of six different aspects was assessed: understanding of general application scenarios, of the concept of data split, of the comparison between different data splits by misclassification error, of the systematic algorithmic

approach of creating a tree, of an interpretation of the entropy function and of the advantages of entropy towards misclassification error. We evaluated comprehension test of nine students: the understanding of aspects of the basic part of the module was good for almost all students. That means, the tasks of general application scenarios, concept data split, comparison of splits and the algorithmic approach were solved well or very well with only few exceptions. The tasks dealing with entropy were solved worse (well or very well only two out of nine students). This is not a surprise because the entropy is the most complicated part of the module. Moreover, in the next implementation we will shift the focus towards the statistical analysis and evaluation of existing decision models and the modeling process itself instead of the growing process. Students will document their analysis and evaluation results in form of interactive computational essays that will show achievements and shortcomings of students understanding. As these data are not yet available at the time of writing, we focused on the description of the course and its theoretical background.

REFERENCES

Behrens, P. & Rathgeb, T. (2017). *JIM-Studie 2017 - Jugend, Information, (Multi-)Media, Basisstudie zum Medienumgang 12- bis 19-jähriger in Deutschland*. Stuttgart: Medienpädagogischer Forschungsverbund Südwest.

Biehler, R., Budde, L., Frischemeier, D., Heinemann, B., Podworny, S., Schulte, C. & Wassong, T. (Hg.). (2018). *Paderborn Symposium on Data Science Education at School Level 2017: The Collected Extended Abstracts*. Universitätsbibliothek Paderborn.

Biehler, R. & Schulte, C. (2018). Perspectives for an interdisciplinary data science curriculum at German secondary schools. In R. Biehler, L. Budde, D. Frischemeier, B. Heinemann, S. Podworny, C. Schulte & T. Wassong (Hg.), *Paderborn Symposium on Data Science Education at School Level 2017: The Collected Extended Abstracts* (S. 2–14). Universitätsbibliothek Paderborn.

Dressel, J. & Farid, H. (2018). The accuracy, fairness, and limits of predicting recidivism. Science Advances, *4*(1).

Engel, J. (2017). Statistical literacy for active citizenship: A call for data science education. *Statistics Education Research Journal*, *16*(1), 44–49.

Engel, J., Erickson, T. & Martignon, L. (2018). Teaching and learning about tree-based methods for exploratory data analysis. In M. A. Sorto, A. White & L. Guyot (Hg.), *Looking back, looking forward. Proceedings of the Tenth International Conference on Teaching Statistics (ICOTS10, July, 2018), Kyoto, Japan*. International Statistical Institute.

Finzer, W. (2017). *Common Online Data Analysis Platform*. www.codap.concord.org

Fleischer, Y. (2019). *Elementarisation and Didactical Analysis of the Topic "Decision Trees in Machine Learning" for Implementation in Mathematics Classroom* [Master thesis]. Universität Paderborn Institut für Mathematik, Paderborn.

Hastie, T., Tibshirani, R. & Friedman, J. H. (2009). *The elements of statistical learning: Data mining, inference, and prediction*. New York: Springer.

Heymann, H. W. (2003). *Why teach mathematics? A focus on general education*. Dordrecht; Boston: Kluwer Academic Publishers.

Issenberg, S. (2012). How President Obama's campaign used big data to rally individual voters. *MIT Technology Review*, *116*(1), 38–49.

LeCun, Y. & Cortes, C. (1998). *The MNIST database of handwritten digits*. http://yann.lecun.com/exdb/mnist/

McKinney, W. (2010). Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference*.

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning*(1), 81–106.

Ridgway, J. (2016). Implications of the Data Revolution for Statistics Education. *International Statistical Review*, *84*(3), 528–549.

Toomey, D. (2017). *Jupyter for data science - Exploratory analysis, statistical modeling, machine learning, and data visualization with Jupyter*. Birmingham: Packt Publishing Ltd.

Topsøe, F. (1974). *Informationstheorie: Eine Einführung*. *Teubner Studienbücher*. Vieweg+Teubner Verlag. https://doi.org/10.1007/978-3-322-94886-1