ROLF BIEHLER & YANNIK FLEISCHER

BRINGING TOGETHER STATISTICS AND COMPUTER SCIENCE EDUCATION: MACHINE LEARNING BY DECISION TREES GROUNDED IN STUDENTS' DATA EXPLORATION EXPERIENCES

PADERBORN COLLOQUIUM ON ARTIFICIAL INTELLIGENCE AND DATA SCIENCE EDUCATION AT SCHOOL LEVEL, 25TH NOVEMBER 2021

UNIVERSITÄT PADERBORN

ProDaBi
Projekt Data Science und Big Data in der Schule

Deutsche Telekom Stiftung

Project Data Science and Big Data at School (ProDaBi)

Initiated and funded by the Deutsche Telekom Stiftung

ProDaBi I : 2018 - 2020
ProDaBi II: 2020 - 2023

Cooperation at the Paderborn University:

Rolf Biehler	Didactics of Mathematics
Carsten Schulte	Didactics of Computer Science


Associates:

Yannik Fleischer	Didactics of Mathematics
Daniel Frischemeier	Didactics of Mathematics
Lukas Höper	Didactics of Computer Science
Sven Hüsing	Didactics of Computer Science
Susanne Podworny	Didactics of Mathematics





www.prodabi.de/en

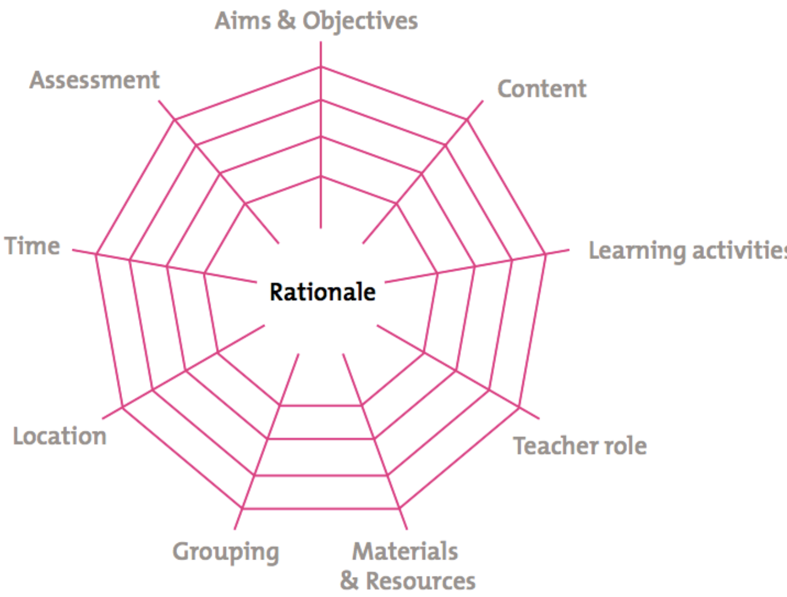
Outline

1. Project overview
2. Predictive modelling with decision trees in ProDaBi
3. Tools for teaching modelling with decision trees
   3.1 Recommender system for food - unplugged with data cards (grade 5/6)
   3.2 Personalized advertisement with JIM data – using CODAP (grade 8-10)
   3.3. Personalized advertisement with JIM data – using Jupyter Notebooks (grade 8 – 12)

4. Some evaluations of students
5. Looking back: Tools and facets of modelling at different levels

Project Goals

- Developing and testing **teaching material** for different grades in the context of design research
- Adapting or developing **digital and unplugged tools** for teaching and learning (data cards, CODAP, Python with Jupyter Notebooks)
- Designing and conducting **professional development courses for teachers**
- Developing **theoretical conceptions including educational goals** for teaching and learning AI and Data Science at school level.



ProDaBi material in different grades

Data Science in grade 12
Yearlong „project course"

Data Science in grade 8 to 10
5 teaching modules + PD courses for teachers

Data Science in grade 5 and 6
2 teaching modules

UNIVERSITÄT PADERBORN

ProDaBi

Projekt Data Science und Big Data in der Schule

Deutsche Telekom Stiftung

Context

Project Course 12

- Data Project Topic
- JIM-Data (CODAP)
- Introduction to Python
- Data Exploration (JN)
- Decision Trees
- Neural Nets
- Project

Modules 8-10

- JIM-Data (CODAP)
- Data Exploration (JN, Arduino)
- Decision Trees (CODAP, JN)
- Data Awareness

Grade 7

An Initiative of the Federal Ministry of Education and Research

Science Year 2019

ARTIFICIAL INTELLIGENCE

Modules 5 / 6

- Decision Trees & Data Cards
- Data Awareness

UNIVERSITÄT PADERBORN

ProDaBi
Projekt Data Science und Big Data in der Schule

Deutsche Telekom Stiftung

Main topics and concepts of ProDaBi

Data Exploration

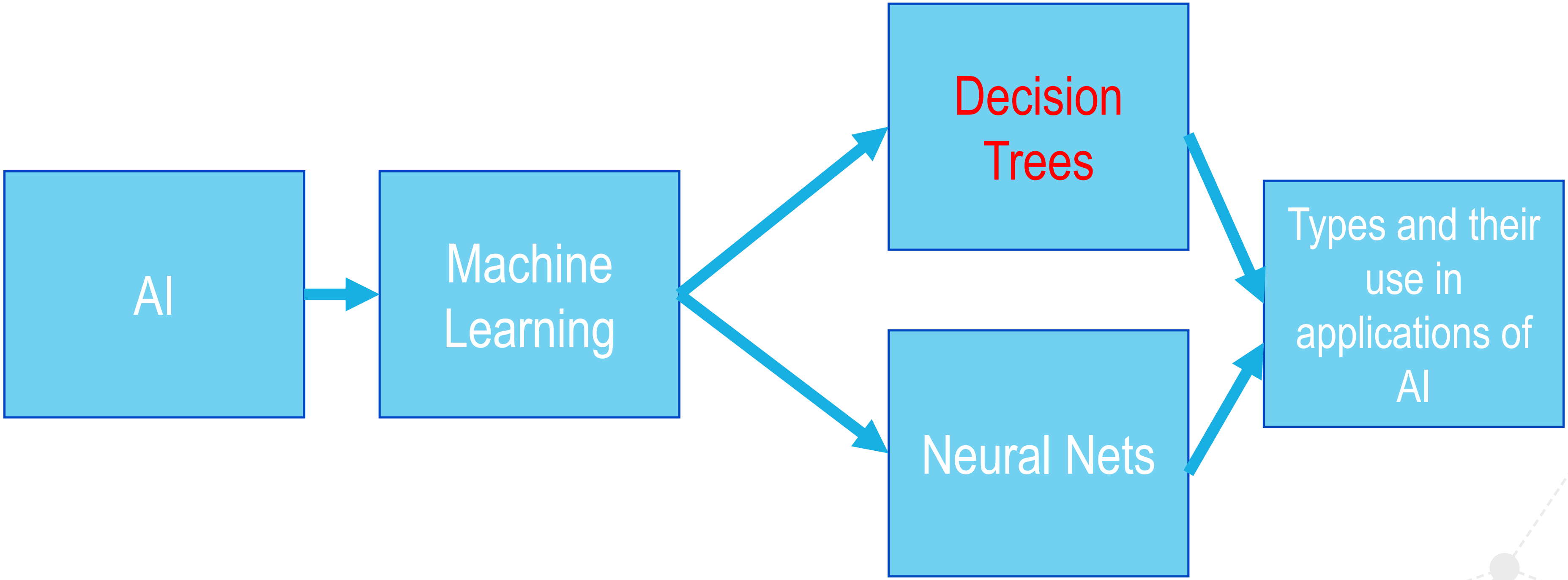
Data Awareness

Epistemic Programming

Machine Learning

Computational Essays; Reproducible and explained Data Analysis and AI

Two types of machine learning: Focus today decision trees



2. Predictive modelling with decision trees in ProDaBi

ML as part of predictive modelling

Predictive modeling is an important facet of data science education, new to traditional statistics and computer science education

Challenges

1. Elementarizing the basic algorithms, developing adequate visualizations and supporting tools
2. Quality assessment of ML models/algorithms
   training data, test test, validation data, bias, range of potential applications
3. Embedding ML in human decision-making scenarios
   realistic, critival view of the power of ML, deployment with ethical monitoring

(Ridgway et al., 2018; Sulmont et al. 2019b, Zieffler et al., 2021)

ML as part of predictive modelling

Predictive modeling is an important facet of data science education, new to traditional statistics and computer science education

Challenges

Our focus today

1. Elementarizing the basic algorithms, developing adequate visualizations and supporting tools
2. Quality assessment of ML models/algorithms
   training data, test test, validation data, bias, range of potential applications
3. Embedding ML in human decision-making scenarios
   realistic, critival view of the power of ML, deployment with ethical monitoring

(Ridgway et al., 2018; Sulmont et al. 2019b, Zieffler et al., 2021)

Why decision trees?

Transparency and teachability

- Relating and contrasting human and ML-built decision trees
- Algorithmic transparency is possible (Adequate mental models)
- Teachable on various levels with various tools with increasing complexity, breadth, and efficiency

Designing and using hybrid human-machine systems

- Students can act as designers of ML/AI not just as trainers of ready-made AI systems
- Possibility of pointing out the role of responsible humans in different stages of creating an AI system

Good start into comprehensive predictive modelling

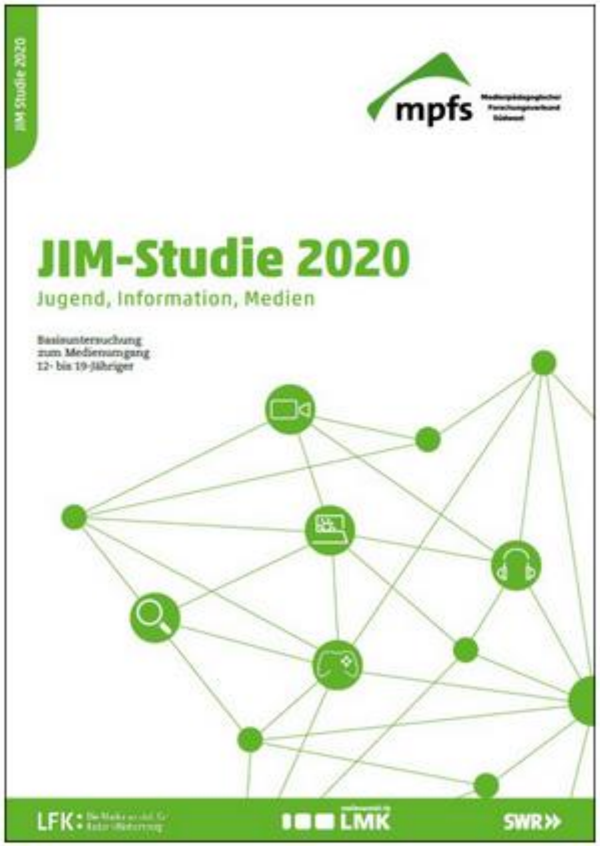
- Teaching artificial neural nets, e.g., can focus on new algorithm

A multivariate dataset: JIM-PB

- Based on an official German survey
- 161 Questions about media use
- We collected data of ~1200 juveniles

Topics:

- Grade, Age, Sex
- Owning digital devices
  - Computer, GameConsole, Tablet, …
- Use of online platforms
  - Instagram, Facebook, TikTok, Youtube, …
- Gaming
- …



Our educational use

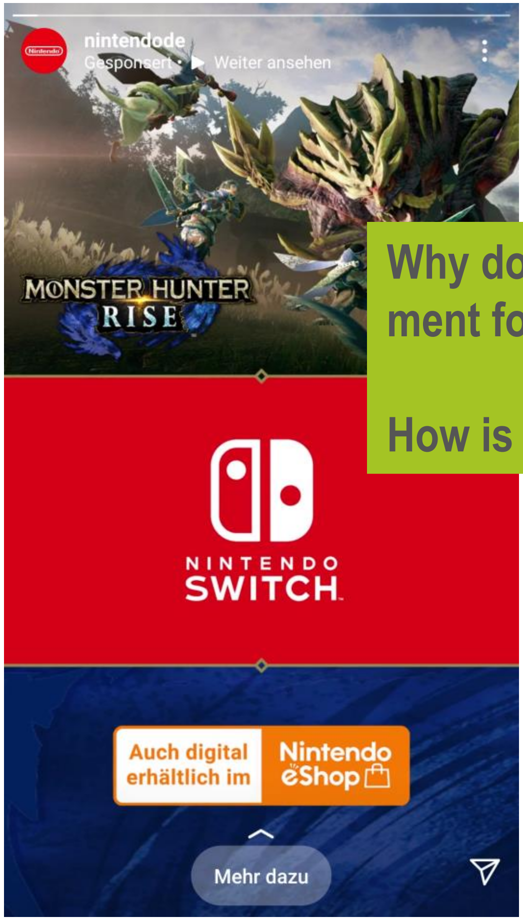
Data exploration of multivariate data

Example for introducing decision trees

A context used in class - Personalized advertisement on online platforms

- Instagram,Youtube, etc.

An example of Yannik's Instagram Feed



Why do I get such an advertise-ment for an online game?

How is this decided?

Predicting frequency of online gaming from other variables

Target variable: frequency of online gaming

Predictor variables:

GameConsole (ownership) yes/no
Computer (ownership) yes/no
Instagram use: rarely/frequently

Assumptions

Target variable is a proxy of „interest in further online games“
Predictor variables are related to the target variable (result of data exploration)
Predictor variables are known to our marketing company

Decision trees: A brief introduction

1. Create a decision tree with level 1, based on the predictor variable that provides the lowest misclassification rate
2. Add further decision steps based on the rest of the variables to further reduce the misclassification rate

UNIVERSITÄT PADERBORN

ProDaBi
Projekt Data Science und Big Data in der Schule

Deutsche Telekom Stiftung

Toy data inspired by Jim data

Target variable

- is to be predicted

	OnlineGames	GameConsole	Computer	Instagram
0	rarely	Yes	Yes	frequently
1	frequently	Yes	No	rarely
2	frequently	Yes	No	rarely
3	frequently	No	Yes	frequently
4	rarely	Yes	No	frequently
5	frequently	Yes	No	rarely
6	frequently	Yes	Yes	rarely
7	rarely	No	No	rarely
8	frequently	Yes	Yes	rarely
9	frequently	Yes	Yes	rarely
10	frequently	Yes	No	frequently
11	rarely	No	No	rarely
12	frequently	No	Yes	frequently
13	rarely	No	No	frequently

predictor variables

- serve to define decision rules for the prediction

UNIVERSITÄT PADERBORN

ProDaBi
Projekt Data Science und Big Data in der Schule

Deutsche Telekom Stiftung

GameConsole as the first candidate for a splitting variable

	OnlineGames	GameConsole	Computer	Instagram
0	rarely	Yes	Yes	frequently
1	frequently	Yes	No	rarely
2	frequently	Yes	No	rarely
3	frequently	No	Yes	frequently
4	rarely	Yes	No	frequently
5	frequently	Yes	No	rarely
6	frequently	Yes	Yes	rarely
7	rarely	No	No	rarely
8	frequently	Yes	Yes	rarely
9	frequently	Yes	Yes	rarely
10	frequently	Yes	No	frequently
11	rarely	No	No	rarely
12	frequently	No	Yes	frequently
13	rarely	No	No	frequently

GameConsole?

Yes

	OnlineGames	GameConsole	Computer	Instagram
0	rarely	Yes	Yes	frequently
1	frequently	Yes	No	rarely
2	frequently	Yes	No	rarely
4	rarely	Yes	No	frequently
5	frequently	Yes	No	rarely
6	frequently	Yes	Yes	rarely
8	frequently	Yes	Yes	rarely
9	frequently	Yes	Yes	rarely
10	frequently	Yes	No	frequently

No

	OnlineGames	GameConsole	Computer	Instagram
3	frequently	No	Yes	frequently
7	rarely	No	No	rarely
11	rarely	No	No	rarely
12	frequently	No	Yes	frequently
13	rarely	No	No	frequently

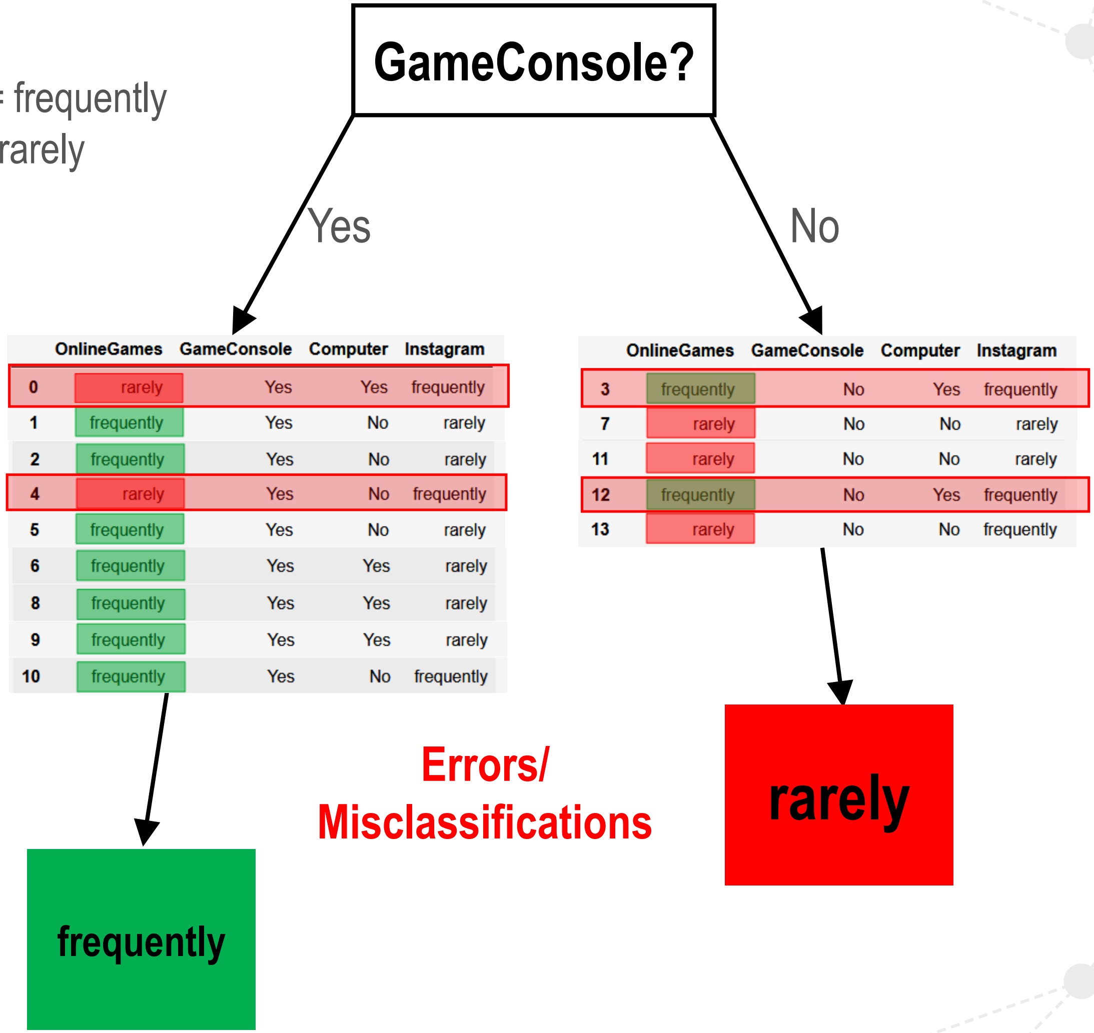
Prediction for OnlineGames **frequently** **rarely**

Tentative Rule 1

If GameConsole = yes, predict onlineGame = frequently
If GameConsole = no, predict onlineGame = rarely

Confusion matrix

Playing_OnlineGames		truth	
		frequently	rarely
prediction	frequently	7	2
	rarely	2	3



	OnlineGames	GameConsole	Computer	Instagram
0	rarely	Yes	Yes	frequently
1	frequently	Yes	No	rarely
2	frequently	Yes	No	rarely
4	rarely	Yes	No	frequently
5	frequently	Yes	No	rarely
6	frequently	Yes	Yes	rarely
8	frequently	Yes	Yes	rarely
9	frequently	Yes	Yes	rarely
10	frequently	Yes	No	frequently

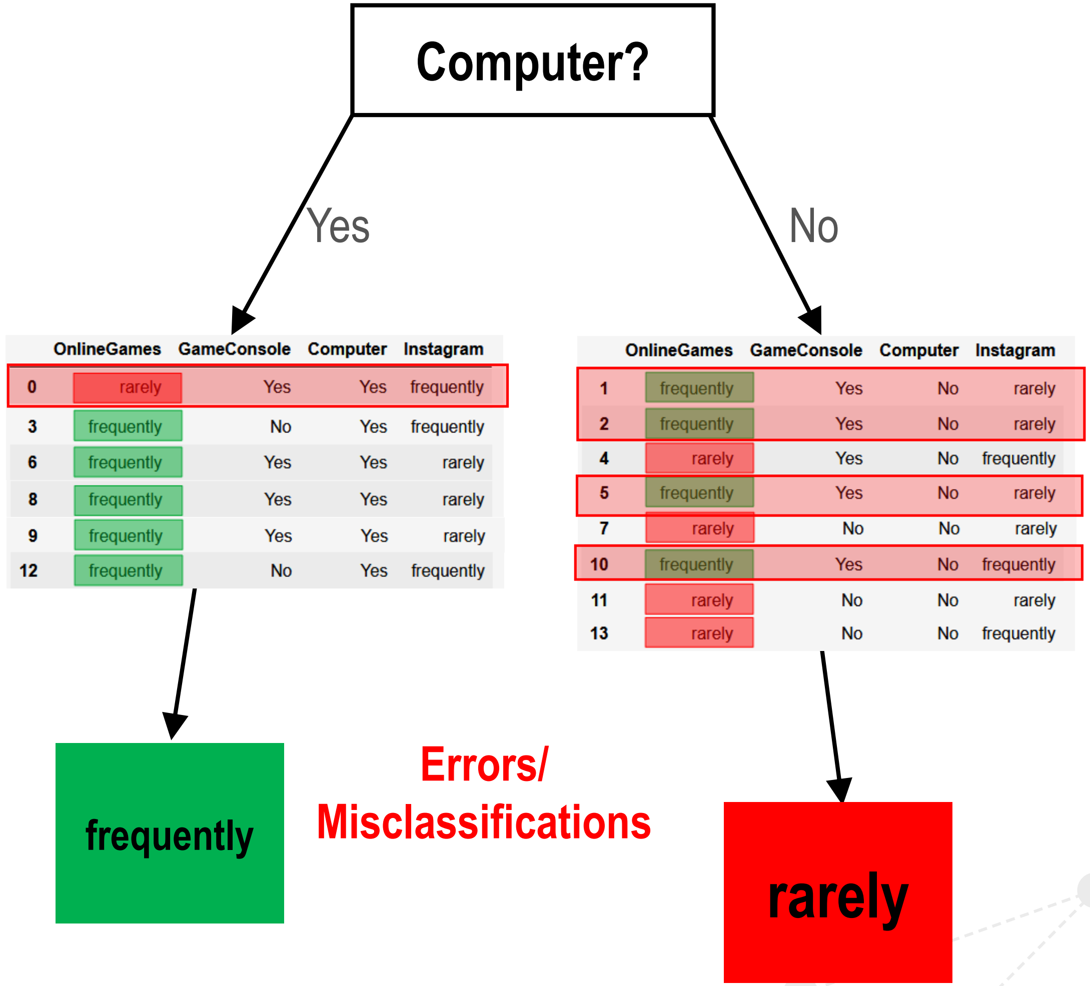
	OnlineGames	GameConsole	Computer	Instagram
3	frequently	No	Yes	frequently
7	rarely	No	No	rarely
11	rarely	No	No	rarely
12	frequently	No	Yes	frequently
13	rarely	No	No	frequently

Split-Variable	Errors	MiscRate*
GameConsole	4	~ 29%

* MiscRate = Misclassification rate

Computer as the second candidate of a splitting variable

Variable	Erros	MiscRate*
GameConsole	4	~ 29%
Computer	5	~ 36%



	OnlineGames	GameConsole	Computer	Instagram
0	rarely	Yes	Yes	frequently
3	frequently	No	Yes	frequently
6	frequently	Yes	Yes	rarely
8	frequently	Yes	Yes	rarely
9	frequently	Yes	Yes	rarely
12	frequently	No	Yes	frequently

	OnlineGames	GameConsole	Computer	Instagram
1	frequently	Yes	No	rarely
2	frequently	Yes	No	rarely
4	rarely	Yes	No	frequently
5	frequently	Yes	No	rarely
7	rarely	No	No	rarely
10	frequently	Yes	No	frequently
11	rarely	No	No	rarely
13	rarely	No	No	frequently

* MiscRate = Misclassification rate

Instagram as the third candidate of a splitting variable

Variable	Errors	MiscRate*
GameConsole	4	~ 29%
Computer	5	~ 36%
Instagram	5	~ 36%

Instagram?

frequently

	OnlineGames	GameConsole	Computer	Instagram
0	rarely	Yes	Yes	frequently
3	frequently	No	Yes	frequently
4	rarely	Yes	No	frequently
5	frequently	Yes	No	rarely
10	frequently	Yes	No	frequently
12	frequently	No	Yes	frequently
13	rarely	No	No	frequently

rarely

	OnlineGames	GameConsole	Computer	Instagram
1	frequently	Yes	No	rarely
2	frequently	Yes	No	rarely
6	frequently	Yes	Yes	rarely
7	rarely	No	No	rarely
8	frequently	Yes	Yes	rarely
9	frequently	Yes	Yes	rarely
11	rarely	No	No	rarely

frequently

Errors/ Misclassifications

rarely

* MiscRate = Misclassification rate

GameConsole as the **best** choice for the **first splitting variable**

Rule 1

If GameConsole = yes, predict onlineGame = frequently
If GameConsole = no, predict onlineGame = rarely

Split-Variable	Errors	MiscRate*
GameConsole	4	~ 29%

GameConsole?

Yes

	OnlineGames	GameConsole	Computer	Instagram
0	rarely	Yes	Yes	frequently
1	frequently	Yes	No	rarely
2	frequently	Yes	No	rarely
4	rarely	Yes	No	frequently
5	frequently	Yes	No	rarely
6	frequently	Yes	Yes	rarely
8	frequently	Yes	Yes	rarely
9	frequently	Yes	Yes	rarely
10	frequently	Yes	No	frequently

No

	OnlineGames	GameConsole	Computer	Instagram
3	frequently	No	Yes	frequently
7	rarely	No	No	rarely
11	rarely	No	No	rarely
12	frequently	No	Yes	frequently
13	rarely	No	No	frequently

Prediction for OnlineGames

frequently

rarely

Looking for further variables to reduce misclassification rate

UNIVERSITÄT PADERBORN

ProDaBi
Projekt Data Science und Big Data in der Schule

Deutsche Telekom Stiftung

GameConsole?

Yes

	OnlineGames	GameConsole	Computer	Instagram
0	rarely	Yes	Yes	frequently
1	frequently	Yes	No	rarely
2	frequently	Yes	No	rarely
4	rarely	Yes	No	frequently
5	frequently	Yes	No	rarely
6	frequently	Yes	Yes	rarely
8	frequently	Yes	Yes	rarely
9	frequently	Yes	Yes	rarely
10	frequently	Yes	No	frequently

No

	OnlineGames	GameConsole	Computer	Instagram
3	frequently	No	Yes	frequently
7	rarely	No	No	rarely
11	rarely	No	No	rarely
12	frequently	No	Yes	frequently
13	rarely	No	No	frequently

	OnlineGames	GameConsole	Computer	Instagram
3	frequently	No	Yes	frequently
12	frequently	No	Yes	frequently

	OnlineGames	GameConsole	Computer	Instagram
7	rarely	No	No	rarely
11	rarely	No	No	rarely
13	rarely	No	No	frequently

Eyeballing shows: Computer has lower MiscRates than Instagram

GameConsole?

Yes

	OnlineGames	GameConsole	Computer	Instagram
0	rarely	Yes	Yes	frequently
1	frequently	Yes	No	rarely
2	frequently	Yes	No	rarely
4	rarely	Yes	No	frequently
5	frequently	Yes	No	rarely
6	frequently	Yes	Yes	rarely
8	frequently	Yes	Yes	rarely
9	frequently	Yes	Yes	rarely
10	frequently	Yes	No	frequently

No

	OnlineGames	GameConsole	Computer	Instagram
3	frequently	No	Yes	frequently
7	rarely	No	No	rarely
11	rarely	No	No	rarely
12	frequently	No	Yes	frequently
13	rarely	No	No	frequently

	OnlineGames	GameConsole	Computer	Instagram
3	frequently	No	Yes	frequently
12	frequently	No	Yes	frequently

	OnlineGames	GameConsole	Computer	Instagram
7	rarely	No	No	rarely
11	rarely	No	No	rarely
13	rarely	No	No	frequently

Eyeballing shows: Computer has lower MiscRates than Instagram

UNIVERSITÄT PADERBORN

ProDaBi
Projekt Data Science und Big Data in der Schule

Deutsche Telekom Stiftung

GameConsole?

Yes

	OnlineGames	GameConsole	Computer	Instagram
0	rarely	Yes	Yes	frequently
1	frequently	Yes	No	rarely
2	frequently	Yes	No	rarely
4	rarely	Yes	No	frequently
5	frequently	Yes	No	rarely
6	frequently	Yes	Yes	rarely
8	frequently	Yes	Yes	rarely
9	frequently	Yes	Yes	rarely
10	frequently	Yes	No	frequently

No

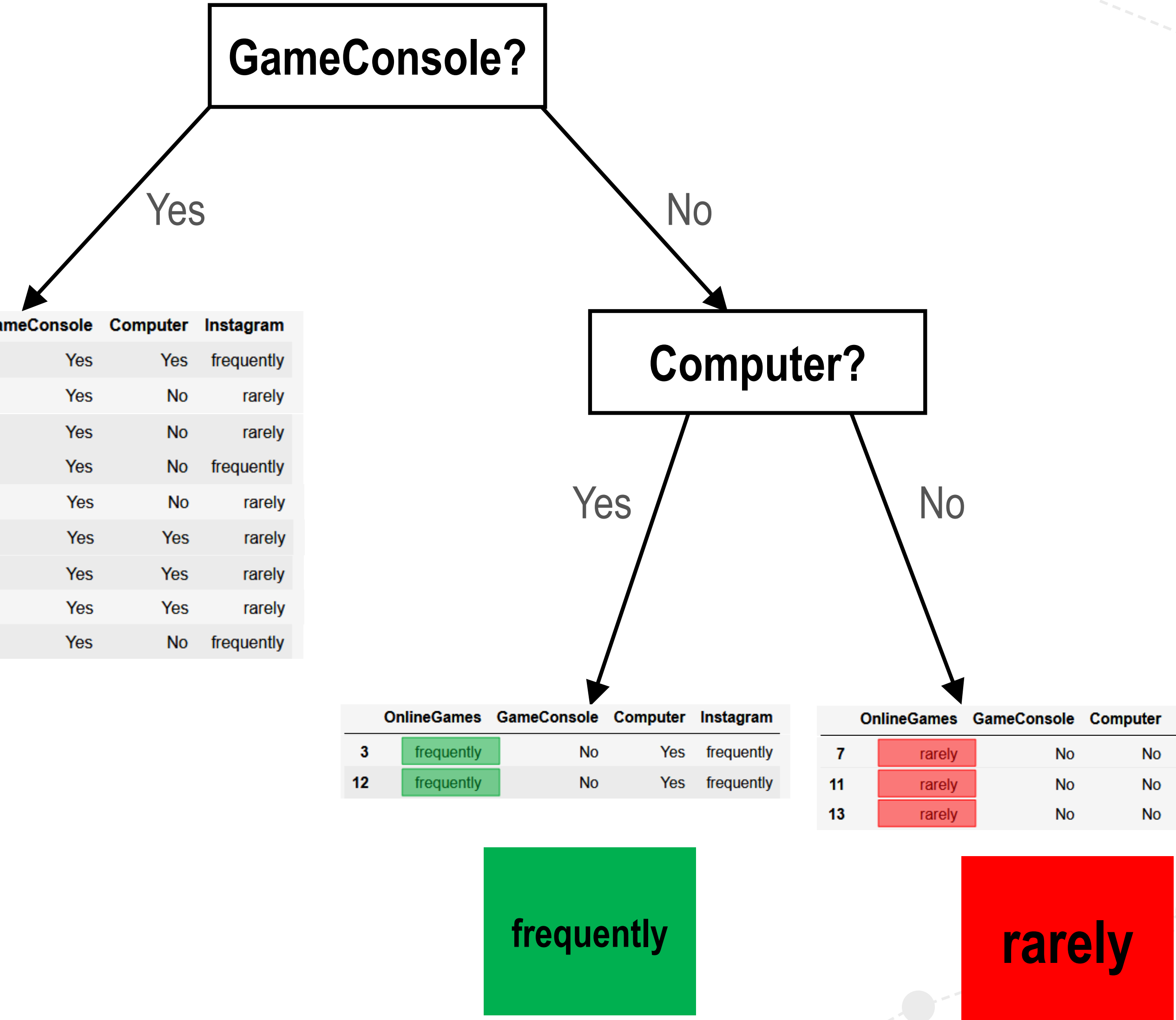
Computer?

Yes

	OnlineGames	GameConsole	Computer	Instagram
3	frequently	No	Yes	frequently
12	frequently	No	Yes	frequently

No

	OnlineGames	GameConsole	Computer	Instagram
7	rarely	No	No	rarely
11	rarely	No	No	rarely
13	rarely	No	No	frequently



GameConsole = Yes:

	OnlineGames	GameConsole	Computer	Instagram
0	rarely	Yes	Yes	frequently
1	frequently	Yes	No	rarely
2	frequently	Yes	No	rarely
4	rarely	Yes	No	frequently
5	frequently	Yes	No	rarely
6	frequently	Yes	Yes	rarely
8	frequently	Yes	Yes	rarely
9	frequently	Yes	Yes	rarely
10	frequently	Yes	No	frequently

GameConsole = No, Computer = Yes:

	OnlineGames	GameConsole	Computer	Instagram
3	frequently	No	Yes	frequently
12	frequently	No	Yes	frequently

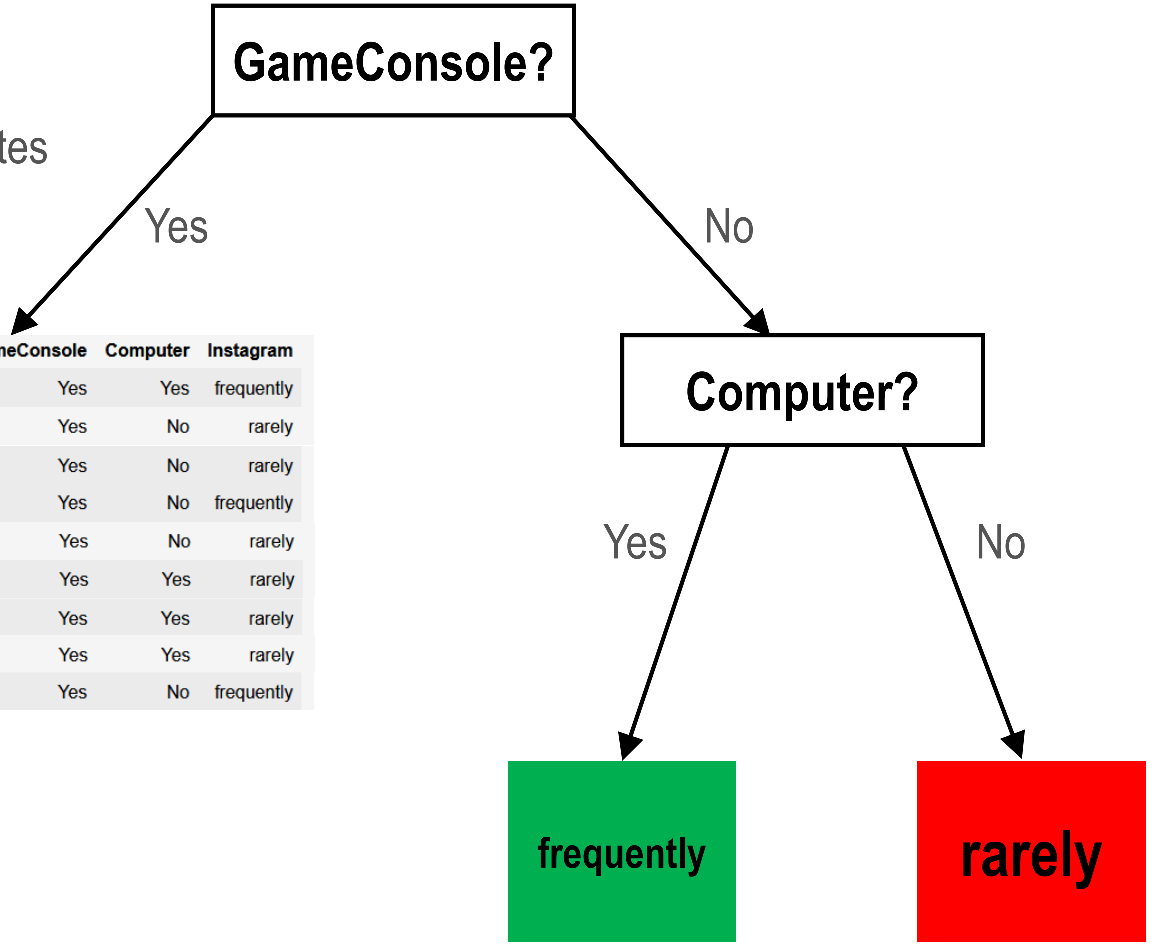
GameConsole = No, Computer = No:

	OnlineGames	GameConsole	Computer	Instagram
7	rarely	No	No	rarely
11	rarely	No	No	rarely
13	rarely	No	No	frequently

Rule 2

Errors 2
MiscRate 14%

Eyeballing shows:
Instagram has lower MiscRates than Computer



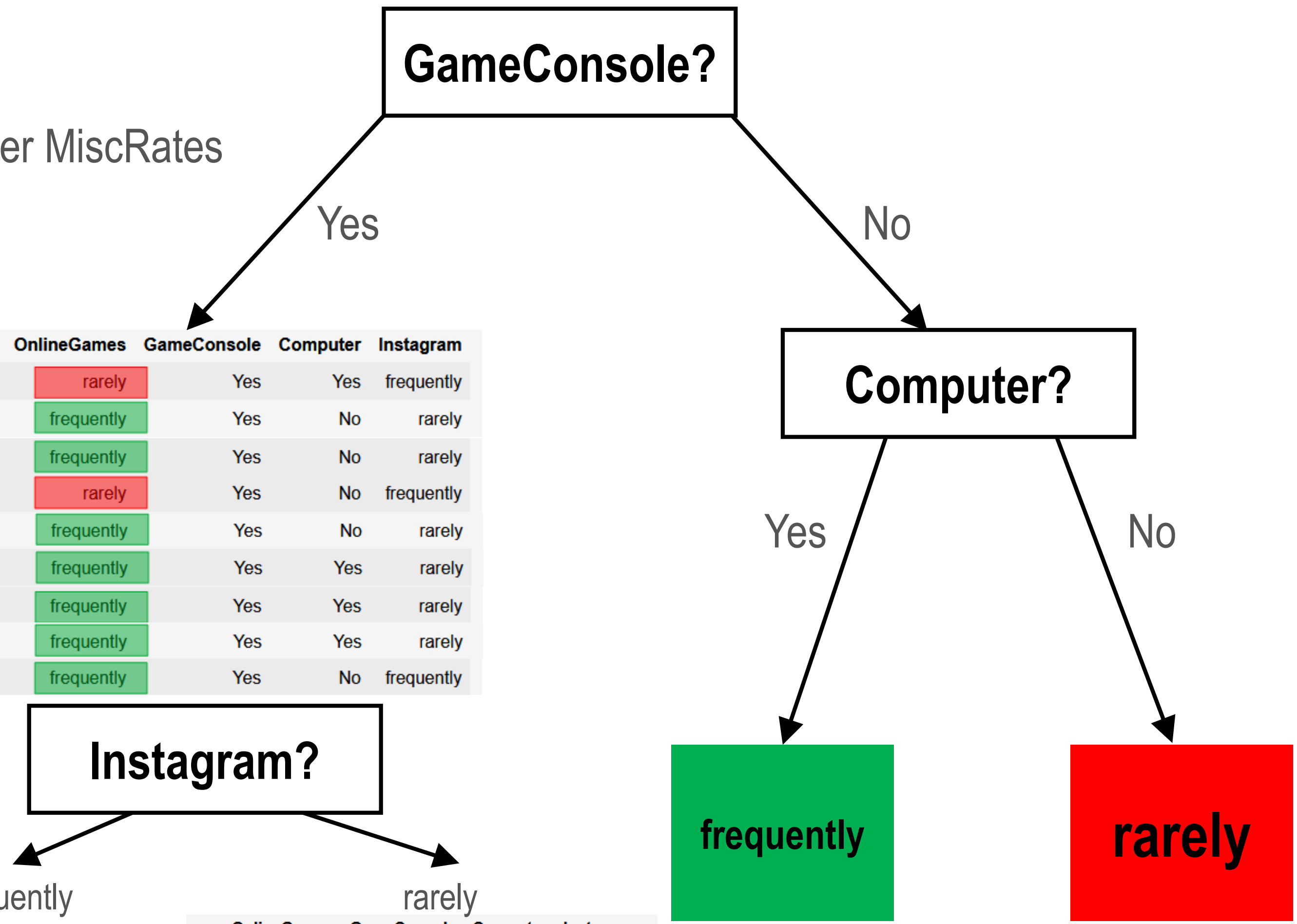
	OnlineGames	GameConsole	Computer	Instagram
0	rarely	Yes	Yes	frequently
1	frequently	Yes	No	rarely
2	frequently	Yes	No	rarely
4	rarely	Yes	No	frequently
5	frequently	Yes	No	rarely
6	frequently	Yes	Yes	rarely
8	frequently	Yes	Yes	rarely
9	frequently	Yes	Yes	rarely
10	frequently	Yes	No	frequently

UNIVERSITÄT PADERBORN

ProDaBi
Projekt Data Science und Big Data in der Schule

Deutsche Telekom Stiftung

Eyeballing shows:
Instagram has lower MiscRates than Computer



	OnlineGames	GameConsole	Computer	Instagram
0	rarely	Yes	Yes	frequently
1	frequently	Yes	No	rarely
2	frequently	Yes	No	rarely
4	rarely	Yes	No	frequently
5	frequently	Yes	No	rarely
6	frequently	Yes	Yes	rarely
8	frequently	Yes	Yes	rarely
9	frequently	Yes	Yes	rarely
10	frequently	Yes	No	frequently

frequently

	OnlineGames	GameConsole	Computer	Instagram
0	rarely	Yes	Yes	frequently
4	rarely	Yes	No	frequently
10	frequently	Yes	No	frequently

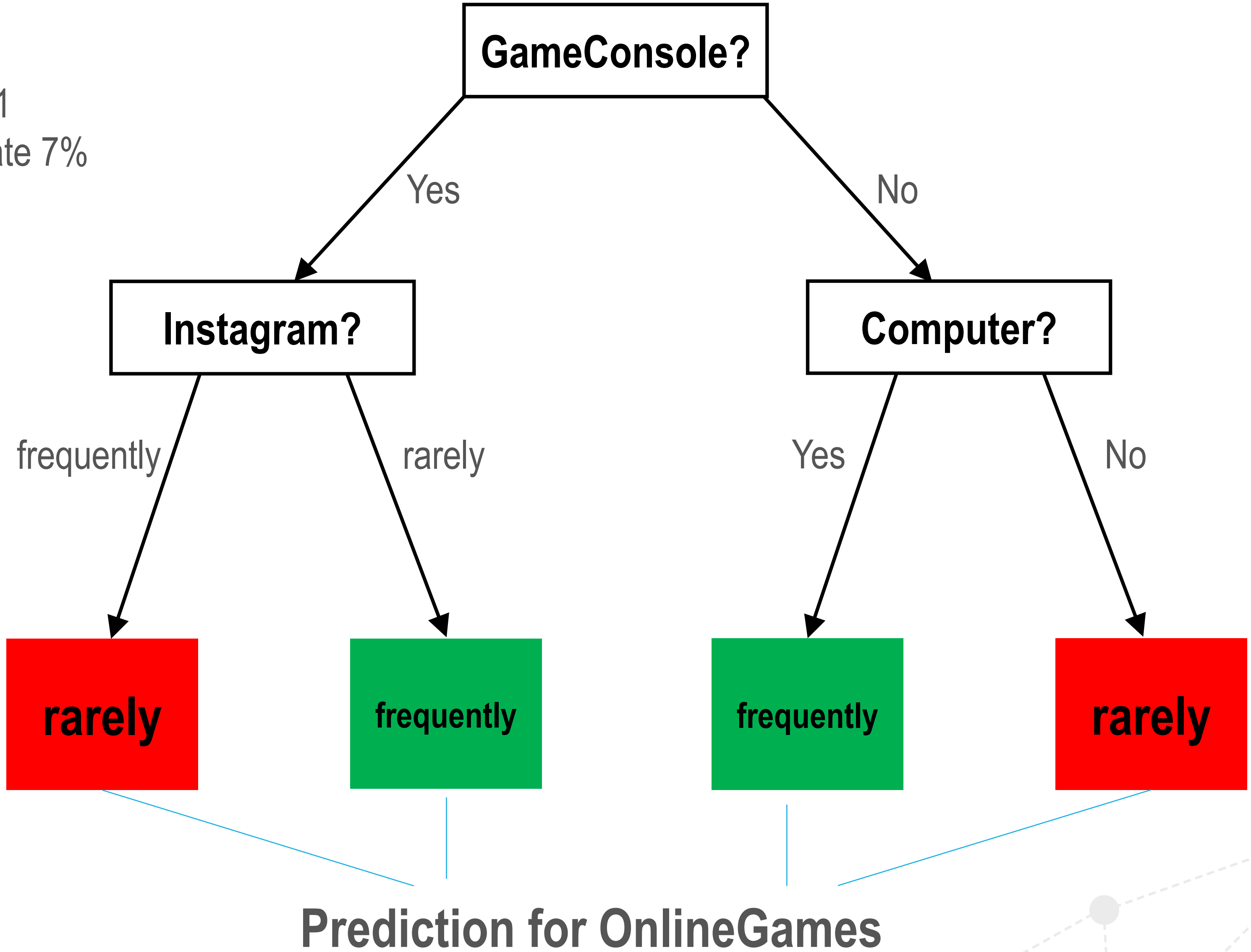
rarely

	OnlineGames	GameConsole	Computer	Instagram
1	frequently	Yes	No	rarely
2	frequently	Yes	No	rarely
5	frequently	Yes	No	rarely
6	frequently	Yes	Yes	rarely
8	frequently	Yes	Yes	rarely
9	frequently	Yes	Yes	rarely

Final rule (to be tested on further data)

Rule 3

Errors 1
MiscRate 7%

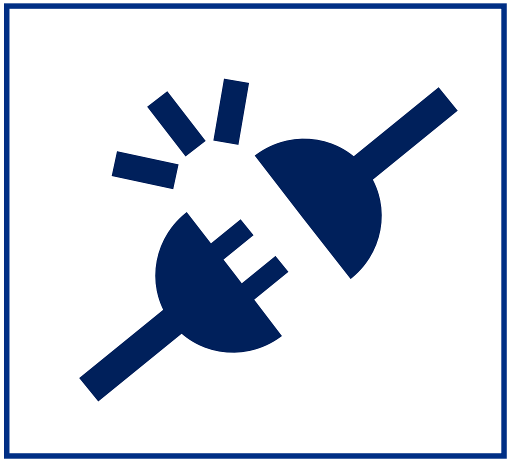


UNIVERSITÄT PADERBORN

ProDaBi
Projekt Data Science und Big Data in der Schule

Deutsche Telekom Stiftung

3. Tools for teaching modeling with decision trees



Manual creation
of decision trees

CODAP

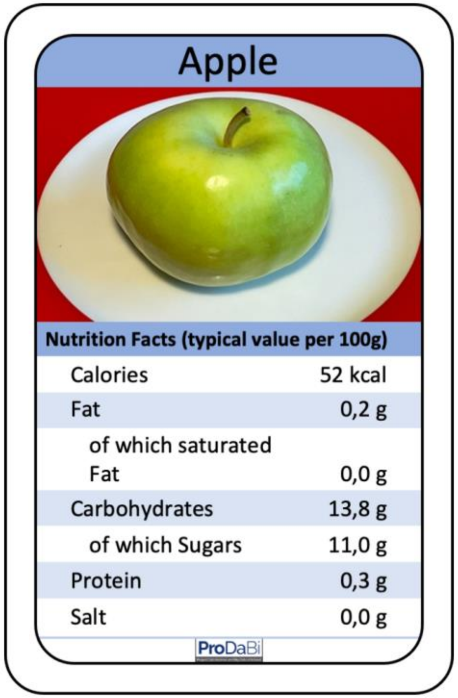
jupyter

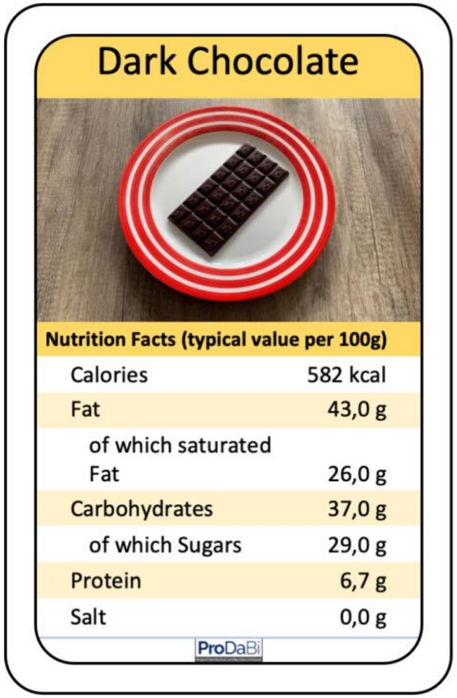
3.1 Recommender system for food - unplugged with data cards

Topic of the series of lessons with data cards

- **Subject**: Manual modeling with decision trees
- **Example**: Recommender system for food items
- **Guiding Questions**:
  - How can we use nutrition information to decide whether a food is **rather recommendable** or **rather not recommendable**?
  - How can a **method of machine learning** help to create a rule system?

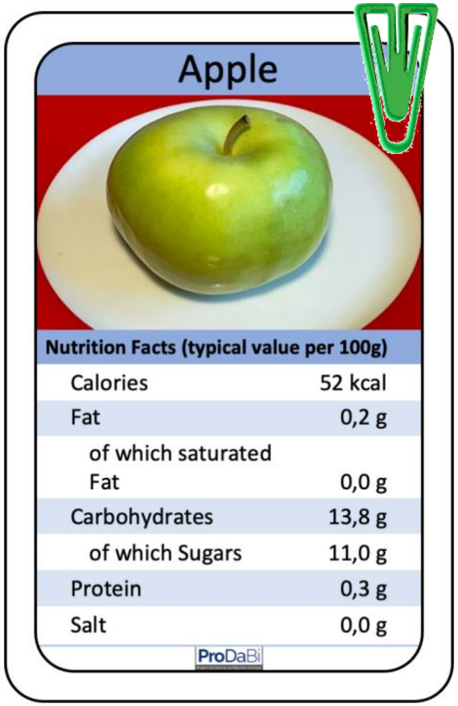
The material

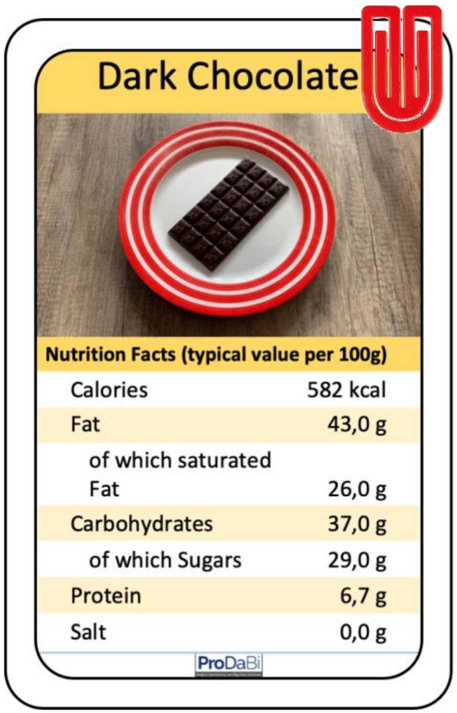




- 55 data cards about food items
    - nutrition facts (typical value per 100g)
- green and red paper clips to label the cards
- worksheets and slides

The material

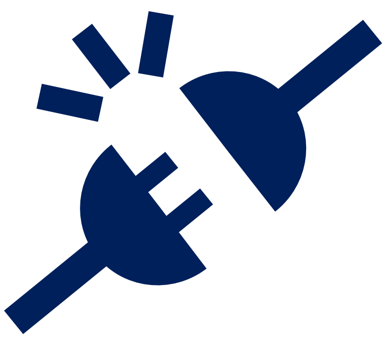




- 55 data cards about food items
  - nutrition facts (typical value per 100g)
- green and red paper clips to label the cards
- worksheets and slides

Demonstration: Defining data based decision rules with data cards

Video_Datacards.mp4

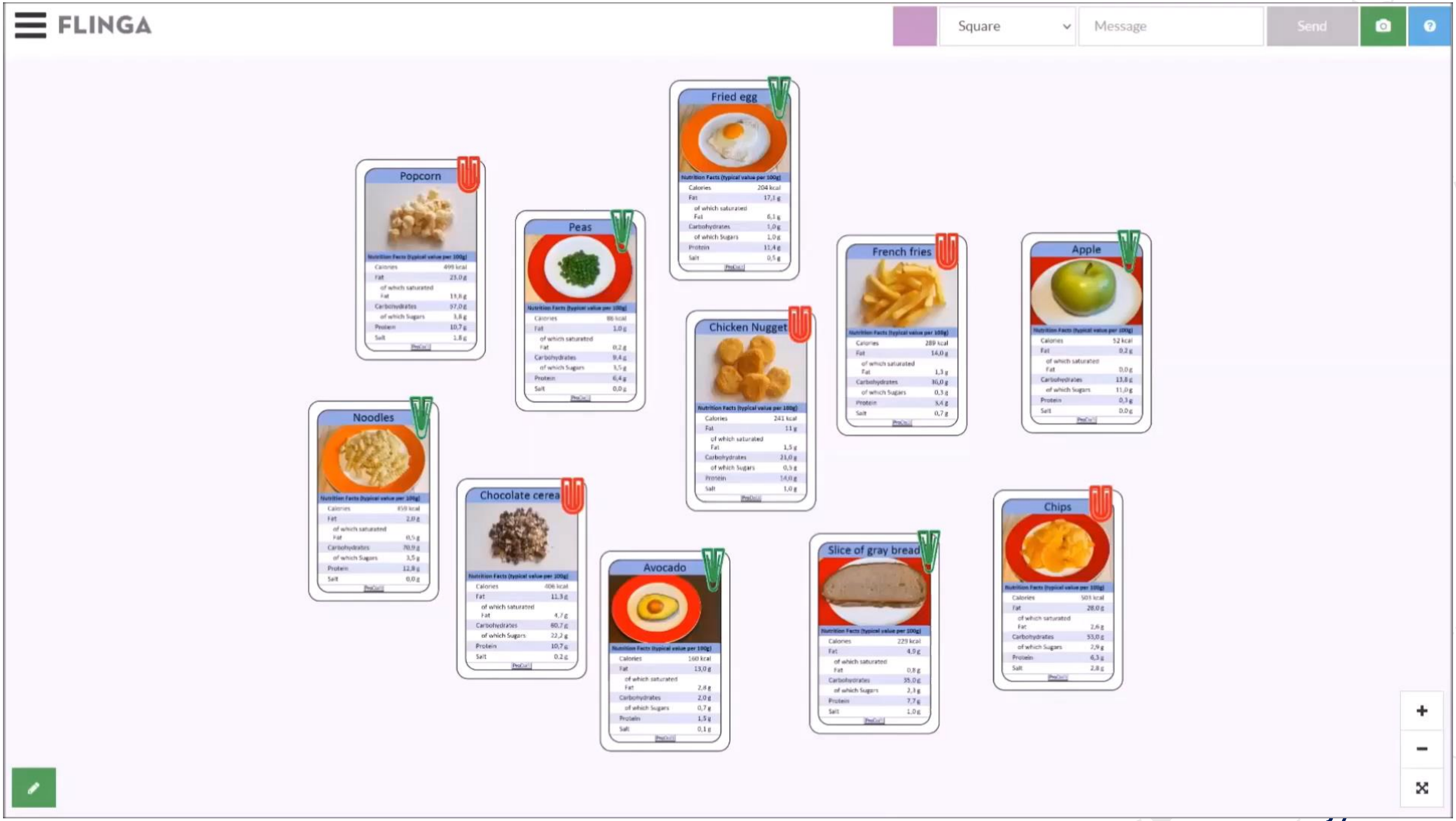


UNIVERSITÄT PADERBORN

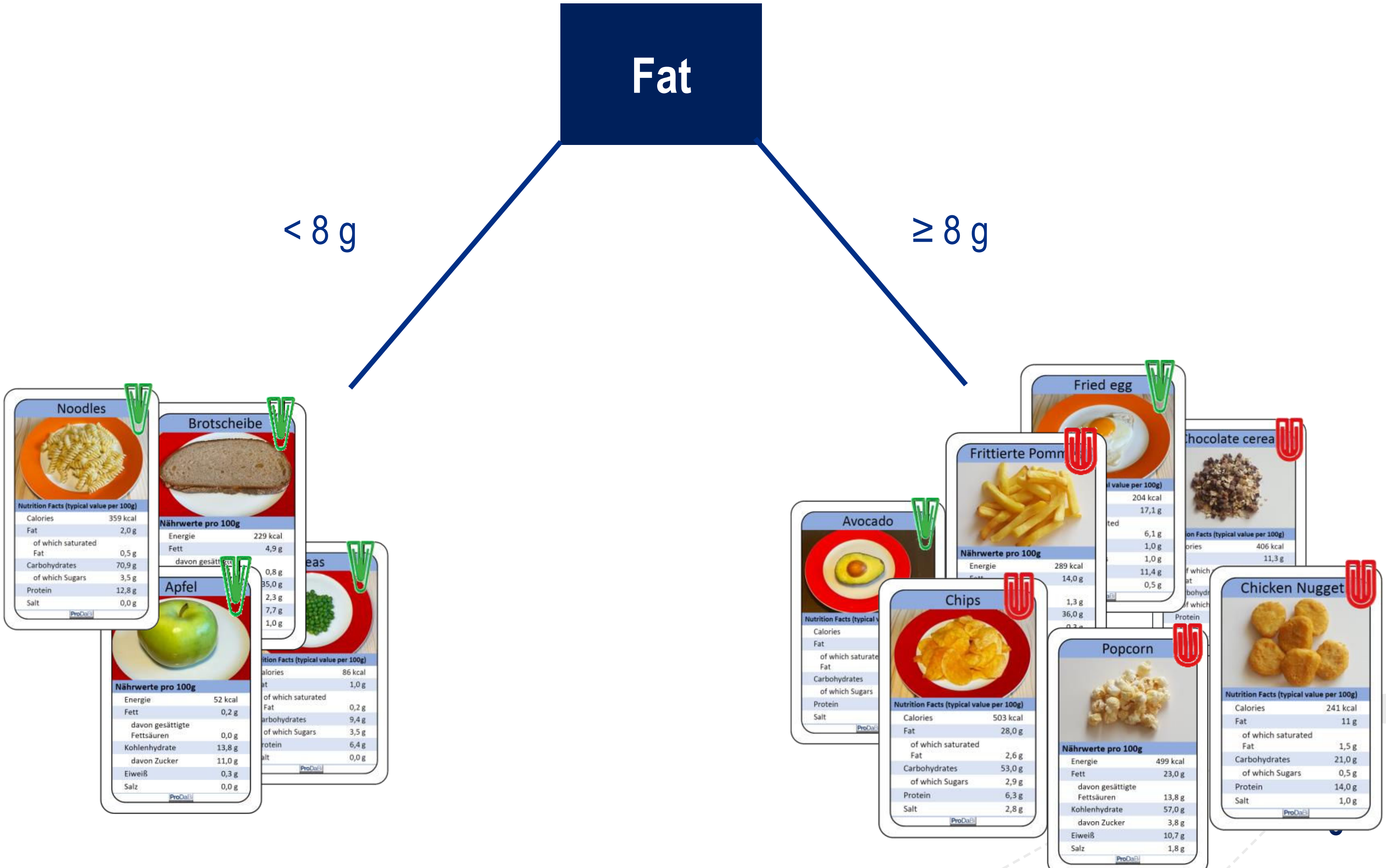
ProDaBi
Projekt Data Science und Big Data in der Schule

Deutsche Telekom Stiftung

How students use the material



Defining decision rules



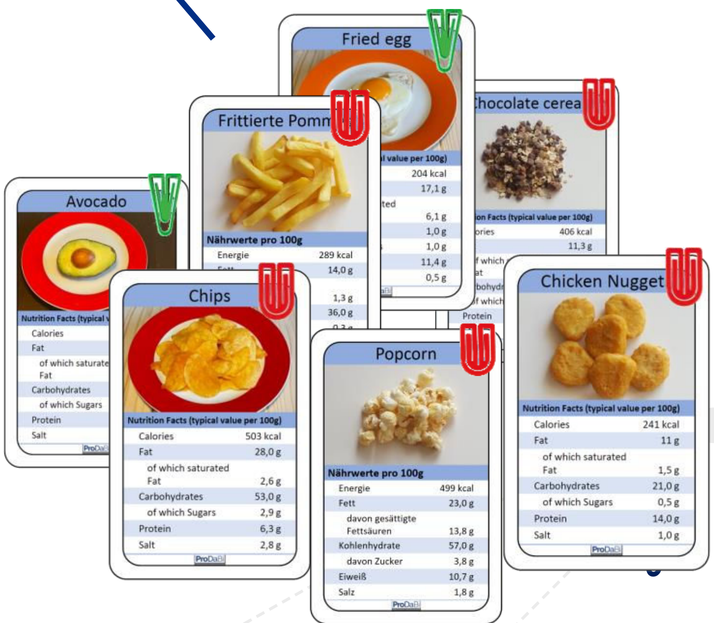
Defining decision rules

Fat

< 8 g

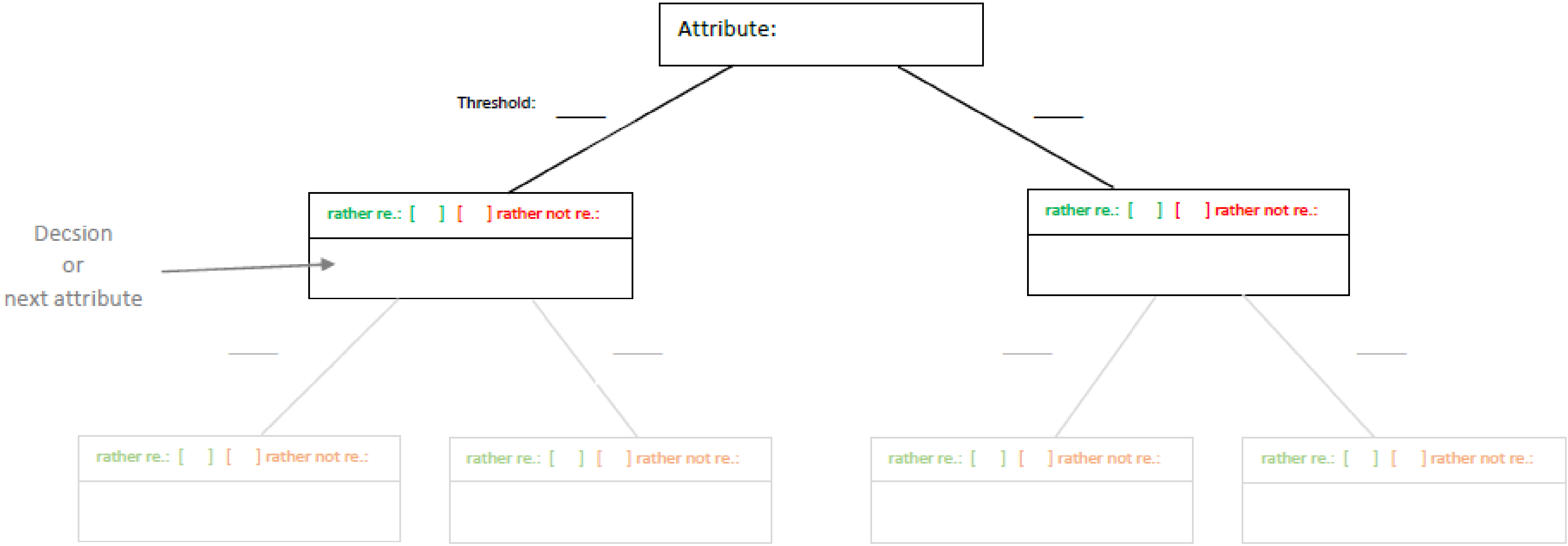
≥ 8 g

rather recommendable



Documentation of decision tree

Tree documentation



UNIVERSITÄT PADERBORN

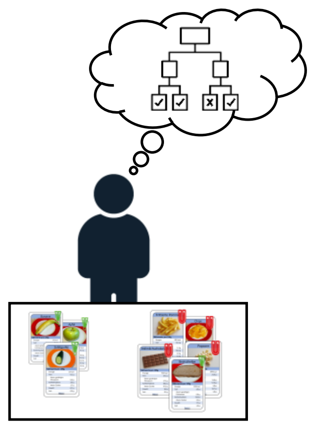
ProDaBi
Projekt Data Science und Big Data in der Schule

Deutsche Telekom Stiftung

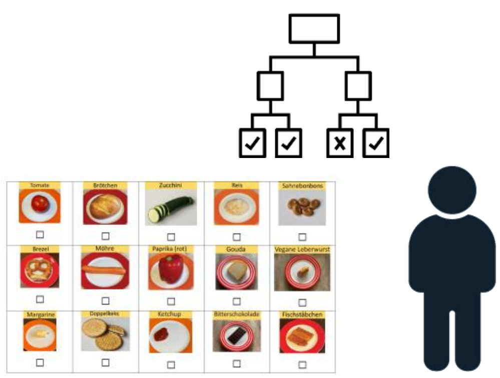
Overview of the series of lessons

Label the data

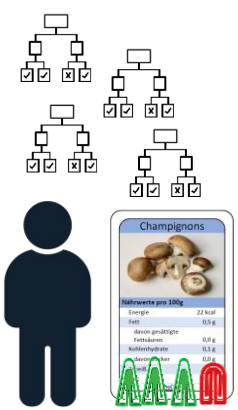
Statistics with embodied activities



Creating decision trees



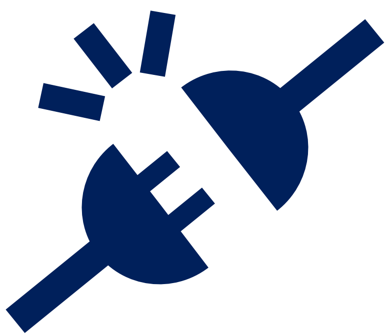
Testing own decision tree with test data



Testing different decision trees with one item

Impressions of students work in class

Tools for teaching modeling with decision trees



CODAP

jupyter

Semi-automatic creation
of decision trees

3.2 Personalized advertisement with JIM data – using CODAP

Topic of the series of lessons with CODAP

- **Subject**: Semi-automatic modeling with decision trees
- **Example**: Personalized advertisement on online platforms (JIM data)
- **Guiding Questions**:
  - How can we use personal data to predict personal interest (e.g. playing onlinegames)?
  - How can we systematically find a good decision tree based on data?

CODAP – A tool for data science

CODAP
Common Online Data Analysis Platform

a product of The Concord Consortium

https://codap.concord.org/

- Easy exploration of multivariate data via drag & drop
- Manual construction of data based decision trees with the plug-in Arbor
  - collaboration with the developer Tim Erickson
  - adding features for teaching machine learning

Decision Trees in CODAP

Demonstration: Basic functionality for free exploration

Video_CODAP1.mp4

UNIVERSITÄT PADERBORN

ProDaBi
Projekt Data Science und Big Data in der Schule

Deutsche Telekom Stiftung

JIM data demo — Version 2.0 (0600) — English

Tables | Graph | Map | Slider | Calc | Text | Plugins — Undo | Redo | Tiles | Options | Help

JIM_53cases

cases (53 cases)

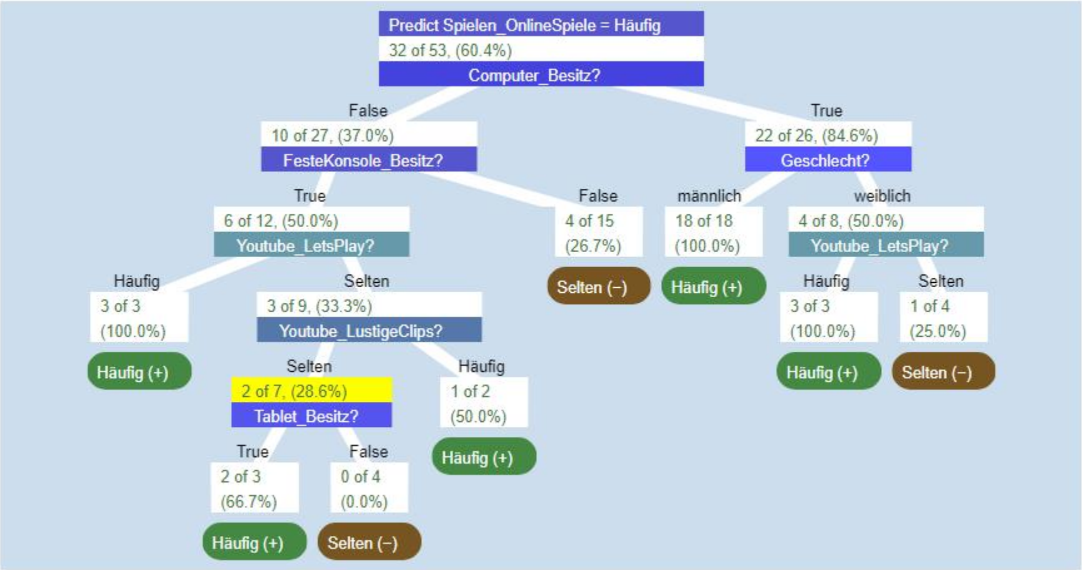
in-dex	Playing OnlineGames	Own Computer	Own GameConsole	Own Tablet	Youtube LetsPlay	Youtube MusicClips	Youtube SportClips	Using Instagram
1	frequently	False	True	True	frequently	frequently	rarely	frequently
2	frequently	True	False	False	frequently	frequently	frequently	frequently
3	frequently	False	True	False	frequently	frequently	frequently	rarely
4	rarely	False	False	False	rarely	rarely	rarely	frequently
5	rarely	False	True	False	rarely	rarely	frequently	frequently
6	rarely	False	False	False	rarely	frequently	rarely	frequently
7	frequently	True	True	False	frequently	frequently	rarely	frequently
8	frequently	False	True	True	rarely	frequently	rarely	frequently
9	rarely	False	False	False	rarely	rarely	rarely	frequently
10	rarely	False	True	False	rarely	rarely	frequently	frequently
11	rarely	False	False	True	rarely	rarely	rarely	frequently
12	rarely	False	True	False	rarely	rarely	rarely	frequently
13	rarely	False	False	False	rarely	rarely	rarely	frequently
14	frequently	True	True	False	frequently	frequently	rarely	frequently
15	frequently	True	True	True	rarely	frequently	frequently	frequently
16	frequently	False	False	False	rarely	rarely	rarely	rarely
17	frequently	False	False	False	rarely	frequently	rarely	rarely
18	frequently	True	True	False	frequently	frequently	frequently	rarely
19	frequently	True	False	False	rarely	rarely	rarely	frequently

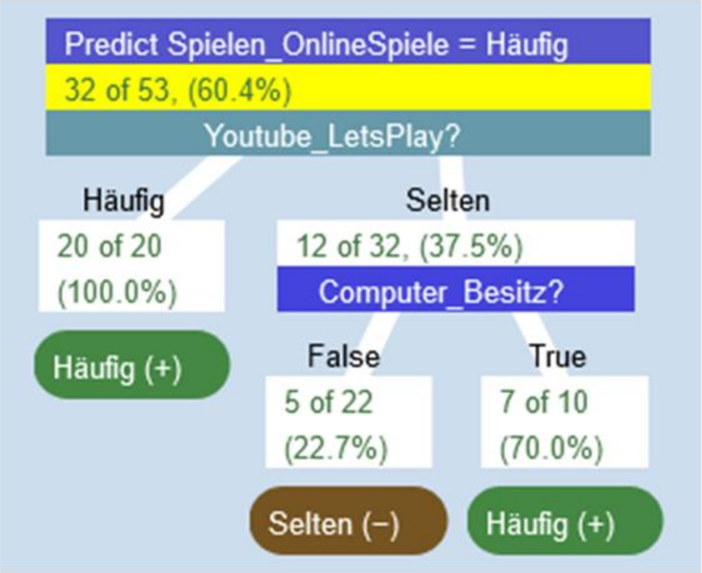
2021s — decision tree

tree | settings | help!

Drag your target attribute here

Student solutions





- Explained contextually why the different variables might be appropriate for predicting the target variables
- he tried different combinations by chance until he could not find any more improvements

Non – Systematic

- tested different variables by "trial&error" as the top decision rule
- searched for partial data sets with relative frequencies of the target value "close to 100% or close to 0%"
- stopped the process very early so that the final partial data sets remain "representative"

Systematic

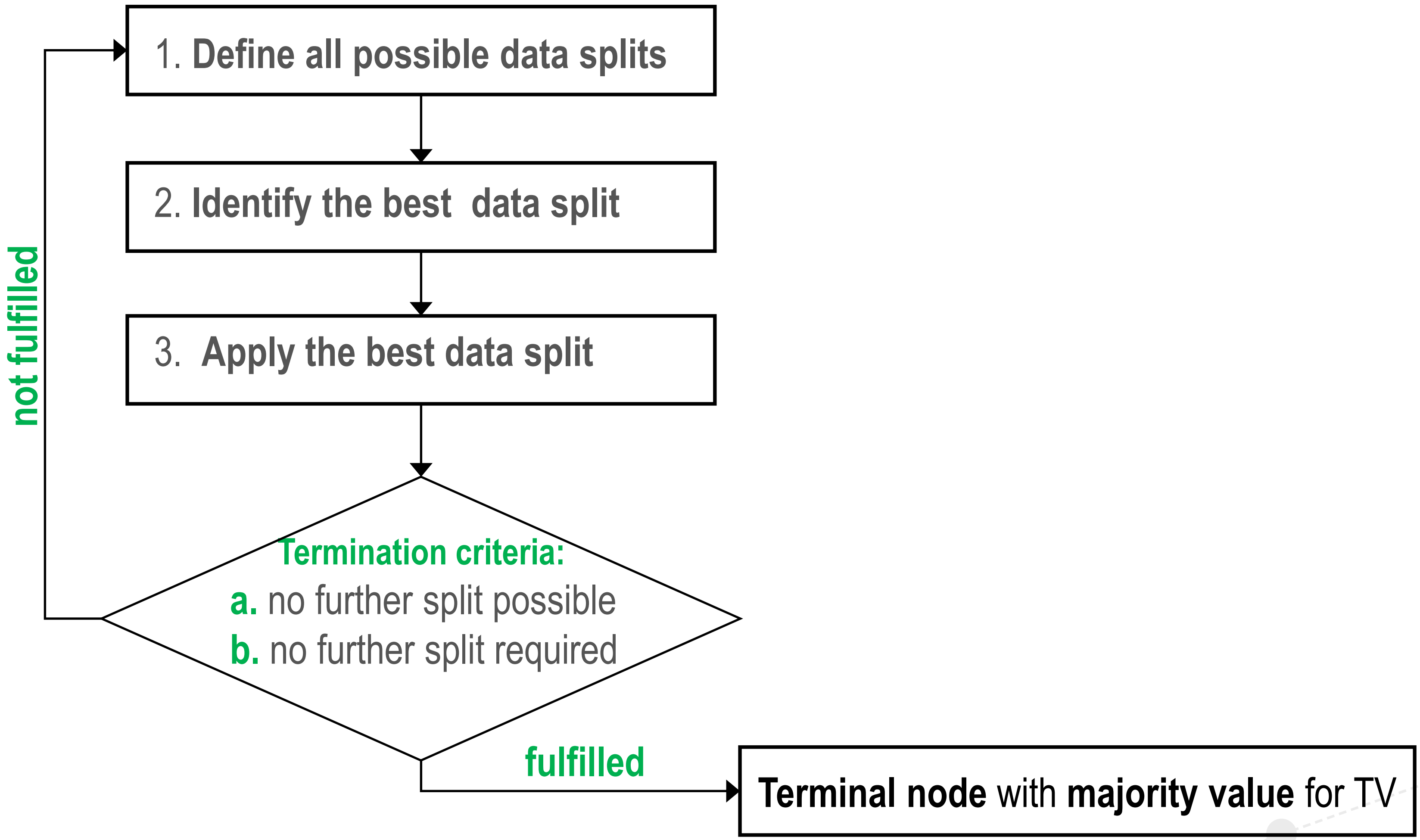
Decision Trees in CODAP

Demonstration: Playing the machine

Video_CODAP2.mp4

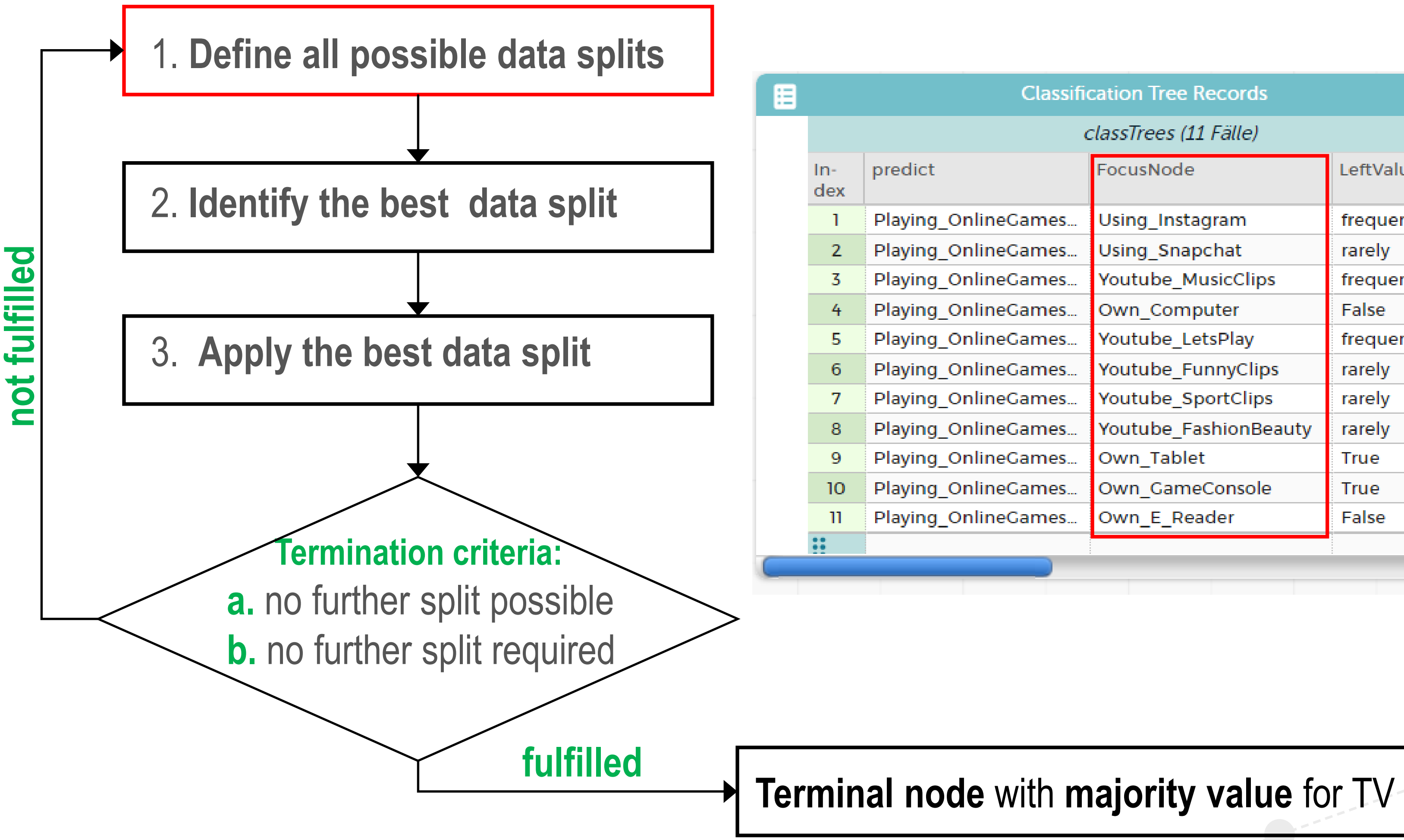
Algorithmic process of creating a decision tree

Input: data , target variable (TV)



Algorithmic process of creating a decision tree

Input: data , target variable (TV)



Classification Tree Records

classTrees (11 Fälle)

Index	predict	FocusNode	LeftValue	MCR
1	Playing_OnlineGames...	Using_Instagram	frequen...	0.396
2	Playing_OnlineGames...	Using_Snapchat	rarely	0.396
3	Playing_OnlineGames...	Youtube_MusicClips	frequen...	0.396
4	Playing_OnlineGames...	Own_Computer	False	0.264
5	Playing_OnlineGames...	Youtube_LetsPlay	frequen...	0.226
6	Playing_OnlineGames...	Youtube_FunnyClips	rarely	0.302
7	Playing_OnlineGames...	Youtube_SportClips	rarely	0.396
8	Playing_OnlineGames...	Youtube_FashionBeauty	rarely	0.377
9	Playing_OnlineGames...	Own_Tablet	True	0.396
10	Playing_OnlineGames...	Own_GameConsole	True	0.34
11	Playing_OnlineGames...	Own_E_Reader	False	0.34

Algorithmic process of creating a decision tree

Input: data , target variable (TV)

1. Define all possible data splits

2. Identify the best data split

3. Apply the best data split

Termination criteria:
a. no further split possible
b. no further split required

not fulfilled

fulfilled

Terminal node with **majority value** for TV

Classification Tree Records

classTrees (11 Fälle)

Index	predict	FocusNode	LeftValue	MCR
1	Playing_OnlineGames...	Youtube_LetsPlay	frequen...	0.226
2	Playing_OnlineGames...	Own_Computer	False	0.264
3	Playing_OnlineGames...	Youtube_FunnyClips	rarely	0.302
4	Playing_OnlineGames...	Own_GameConsole	True	0.34
5	Playing_OnlineGames...	Own_E_Reader	False	0.34
6	Playing_OnlineGames...	Youtube_FashionBeauty	rarely	0.377
7	Playing_OnlineGames...	Using_Instagram	frequen...	0.396
8	Playing_OnlineGames...	Using_Snapchat	rarely	0.396
9	Playing_OnlineGames...	Youtube_MusicClips	frequen...	0.396
10	Playing_OnlineGames...	Youtube_SportClips	rarely	0.396
11	Playing_OnlineGames...	Own_Tablet	True	0.396

Algorithmic process of creating a decision tree

Input: data , target variable (TV)

1. **Define all possible data splits**
2. **Identify the best data split**
3. **Apply the best data split**

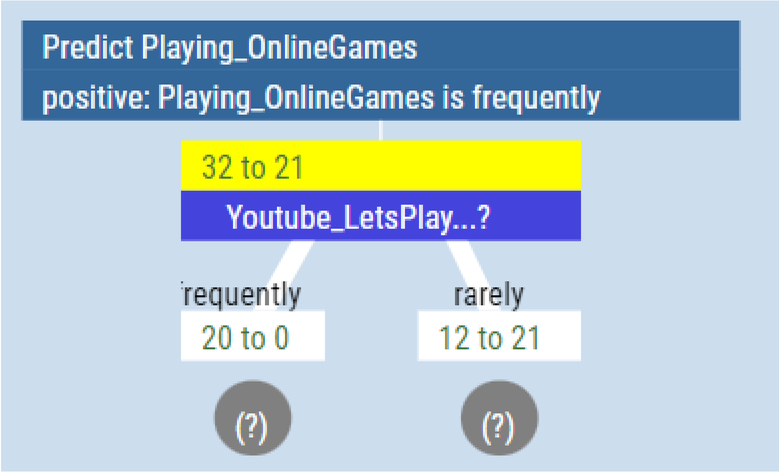
Termination criteria:

a. no further split possible

b. no further split required

not fulfilled → back to 1.

fulfilled → **Terminal node** with **majority value** for TV



Algorithmic process of creating a decision tree

Input: data , target variable (TV)

1. Define all possible data splits

2. Identify the best data split

3. Apply the best data split

Termination criteria:

a. no further split possible

b. no further split required

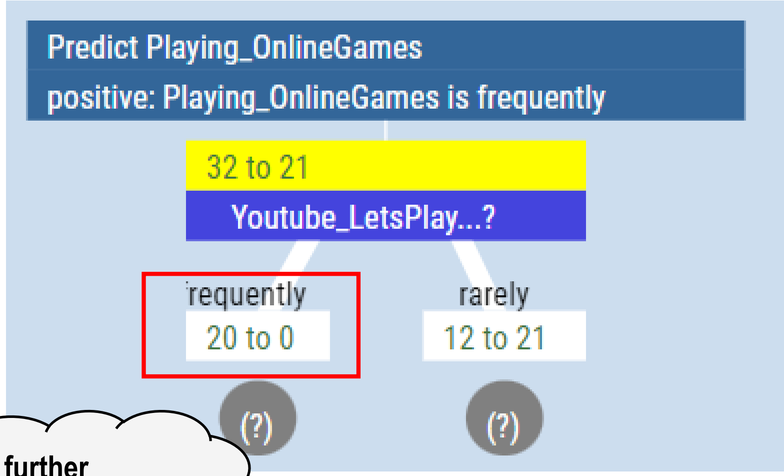
No further predictor variables available?

Is the partial data set pure?

not fulfilled

fulfilled

Terminal node with **majority value** for TV



Algorithmic process of creating a decision tree

Input: data , target variable (TV)

1. Define all possible data splits
2. Identify the best data split
3. Apply the best data split

Termination criteria:

a. no further split possible

b. no further split required

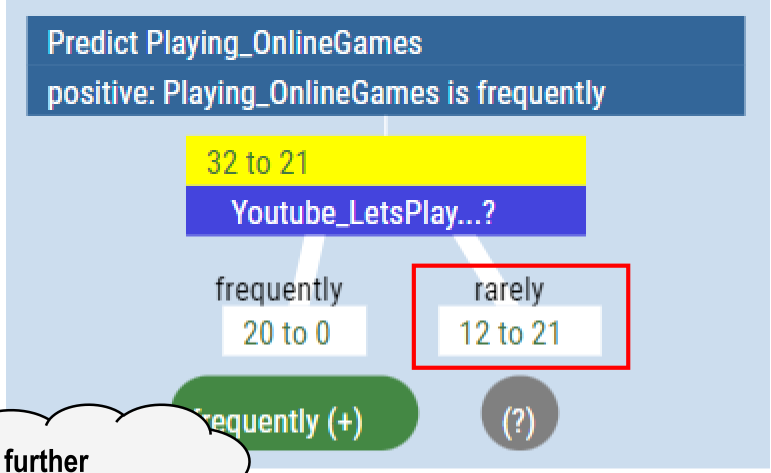
No further predictor variables available?

Is the partial data set pure?

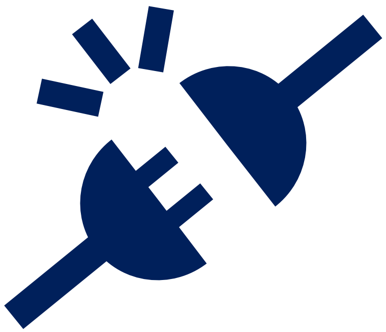
not fulfilled

fulfilled

Terminal node with **majority value** for TV



Tools for teaching modeling with decision trees



CODAP

jupyter

Automatic creation of decision trees

UNIVERSITÄT PADERBORN

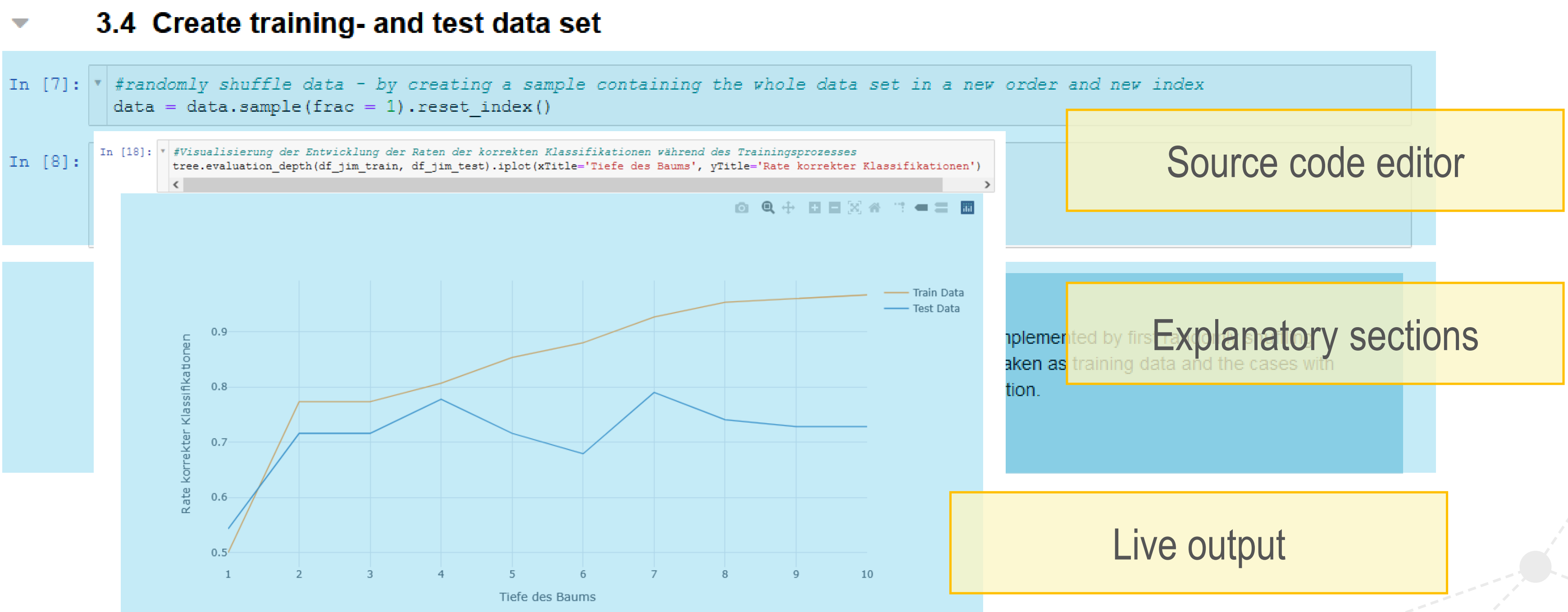
ProDaBi
Projekt Data Science und Big Data in der Schule

Deutsche Telekom Stiftung

3.3. Personalized advertisement with JIM data – using Jupyter Notebooks (grade 8 – 12)

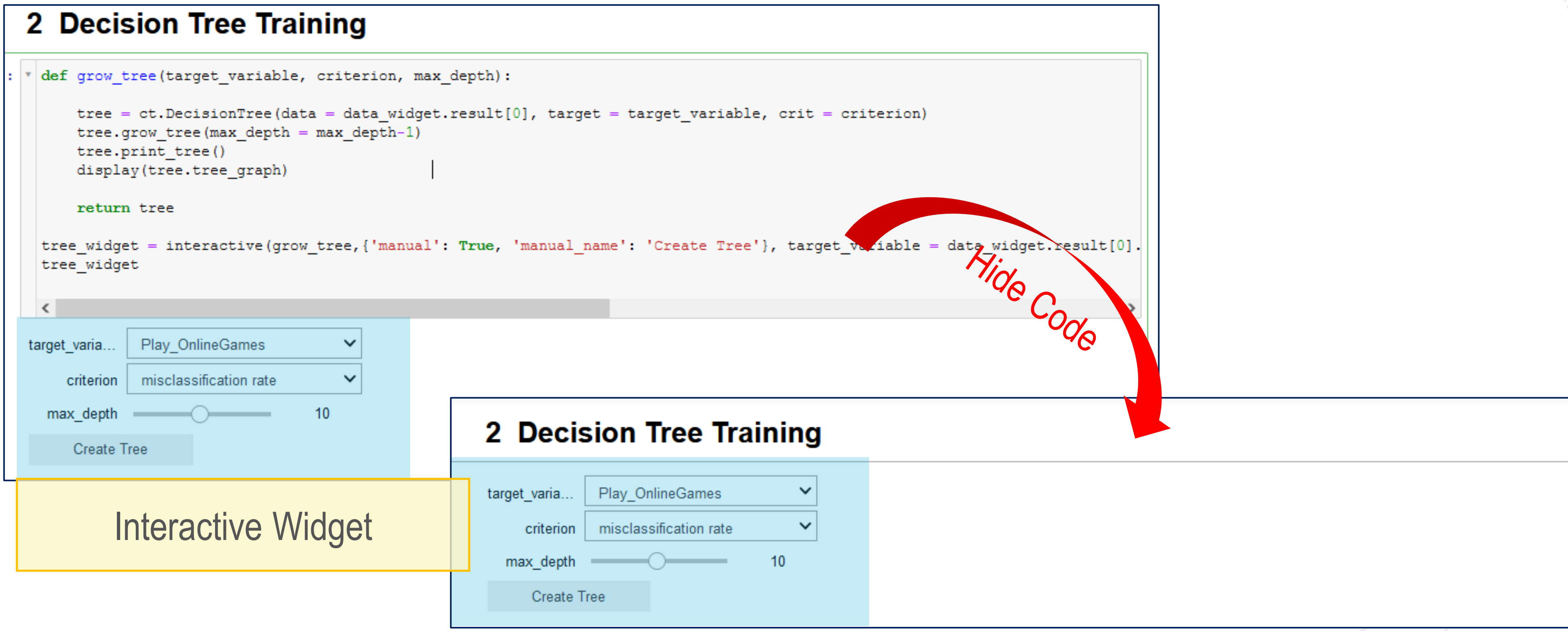
Jupyter Notebook (with Python)

- **Jupyter Notebook is a cell-based environment that can be used versatilly for teaching**
  - Explanatory cells (text an pictures)
  - Code cells (create/ vary python code)
  - Live output (Output of code directly below code cell)



Jupyter Notebook (with Python)

- **Interactive widgets with hidden source code**
  - Jupyter Notebooks as interactive Tools without students noticing python commands



PyTree Library as tool for creating decision trees

- We have developed a **library (PyTree)** of **prepared commands** for students to create **decision trees based on data** and to create **meaningful visualizations**

- **Behind the scenes:**

```python
def grow_tree(self, data = pd.DataFrame(), target = None, crit = 'entropy', max_depth = float('inf'), act_depth = 0, min_gain = 0, min_leaf_cases=0):


    attributes = (data.columns).drop(target)
    #print('loading...')

    if (data[target].nunique() == 1) or (len(attributes) == 0) or (act_depth >= max_depth) or (len(data) < min_leaf_cases):
        # Falls nur ein Wert für die Zielvariable vorliegt, gib ein leaf mit diesem Wert aus
        # Falls Anzahl der Attribute 0 ist, gib ein leaf mit dem Mehrheitswert der Zeilvariable aus
        # Falls maximale Tiefe des Baums erreicht ist, gib leaf mit Mehrheitswert der Zeilvariable aus
        self.return_leaf_node(data, target)

    # Falls vorherige Abfragen nicht zutrafen wird ein weiterer Split gesucht um ihn anzuwenden
    else:
        #Finde alle möglichen Splits
        list_of_splits = find_all_splits(data, target)

        if len(list_of_splits) > 0:
            #Identifiziere den besten Split unter allen Splits
            best_split = identify_best_split(data, target, list_of_splits, self.criterion)


            # Überprüfen: Ist der Split produktiv?
            if information_gain(data, target, best_split) > min_gain: # Ist best_split produktiv?

                #Wende den besten Split auf die Inputdaten an und erstelle somit ein Liste von Teildatensätzen
                list_of_subsets = apply_split(data, best_split)


                #Den erstellten Split als Knoten ausgeben, falls best_split produktiv ist
                current_node = self.return_split_node(best_split, data)


                #Rekursive weitere Anwendung für jeden erstellten Teildatensatz
                for i in range(len(list_of_subsets)):

                    next_node_nr = len(self.tree_nodes) + 1
                    new_input_subset = list_of_subsets[i].drop(best_split.attribute, axis = 1)

                    self.grow_tree(new_input_subset, target, self.criterion, max_depth, act_depth+1)

                    self.new_edge(root = current_node.node_nr, target = next_node_nr, label = best_split.split_values[i])
```

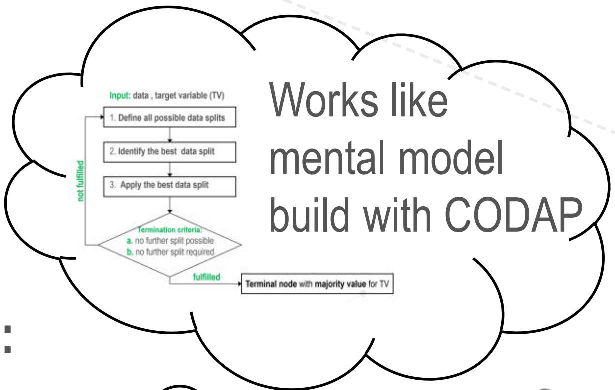
Commands for students:

Automatic creation
- grow_tree()
- validation_pruning()

Manual creation and editing
- manual_split()
- manual_prune()

Evaluation and visualization
- prediction_accuracy()
- evaluation_depth()
- evaluate_fairness()
- …

PyTree Library



- **What students see in code-based Notebooks:**

```
# Import Decision Tree Library
import PyTree
```

```
#Training of Decision Tree
tree.grow_tree(df_input = data_jim, target = 'Playing_OnlineGames')
```

```
#Training of Decision Tree
tree.grow_tree(df_input = data_jim, target = 'Playing_OnlineGames', crit = 'entropy', max_depth = 5, min_leaf_nodes = 10)
```

- **What students see in menu-based Notebooks:**

2 Decision tree training

target_variable: Play_OnlineGames

criterion: misclassification rate

max_depth: 10

Create Tree

Use of Jupyter Notebooks in Class

- **Tool-JNs**
  - menu-based (limited actions)
  - focus one aspect of learning about decision trees (overfitting, pruning, evaluation, …)

- **Worked example JN**
  - code-based (unlimited actions with python code)
  - presents a whole modelling process (from data preparation to evalutation of the final decision tree)
  - Additional narrative enhancements (code explanation, context explanation, reasons for human decisions)

- **Computational Essay JN**
  - code-based (unlimited actions with python code)
  - students document their modelling process adapted from the worked example

Tool Notebook: Jupyter Notebook with hidden Code

Demonstration: Jupyter Notebooks for creating Decision Trees

Video_Jupyter.mp4

UNIVERSITÄT PADERBORN

ProDaBi
Projekt Data Science und Big Data in der Schule

Deutsche Telekom Stiftung

jupyter ProDaBi_Decision_Tree_auto_pruning Last Checkpoint: 17.06.2021 (autosaved)

Logout

File Edit View Insert Cell Kernel Navigate Widgets Help

Trusted | Python 3

Run | Code

1 Import data

Dateiname jim_binaer.csv

```
Trainingsdaten
```

	Play_OnlineGames	Gender	Use_Twitter	Use_Snapchat	Use_Instragam	Youtube_MusicVideos	Youtube_LetsPlay	Youtube_FunnyClips	Youtube_SportVideos
0	frequently	male	rarely	frequently	frequently	frequently	frequently	frequently	rarely
1	frequently	male	rarely	rarely	frequently	frequently	frequently	rarely	rarely
2	rarely	female	rarely	rarely	frequently	rarely	rarely	rarely	rarely
3	frequently	male	rarely	rarely	rarely	frequently	frequently	rarely	rarely
4	frequently	male	rarely	frequently	frequently	frequently	frequently	frequently	frequently
...	...	...	...	...	...	...	...	...	...
145	frequently	male	frequently	rarely	frequently	rarely	rarely	rarely	rarely
146	rarely	female	rarely	rarely	rarely	frequently	rarely	frequently	rarely
147	frequently	male	frequently	frequently	frequently	frequently	frequently	rarely	rarely
148	frequently	male	frequently	frequently	frequently	frequently	frequently	frequently	frequently
149	frequently	male	rarely	rarely	frequently	rarely	frequently	rarely	frequently

150 rows × 15 columns

```
Testdaten
```

	Play_OnlineGames	Gender	Use_Twitter	Use_Snapchat	Use_Instragam	Youtube_MusicVideos	Youtube_LetsPlay	Youtube_FunnyClips	Youtube_SportVideos
0	rarely	female	rarely	rarely	rarely	NaN	rarely	NaN	frequently
1	rarely	female	rarely	rarely	rarely	frequently	NaN	NaN	frequently
2	rarely	female	rarely	frequently	frequently	frequently	rarely	rarely	rarely
3	rarely	male	rarely	frequently	frequently	rarely	rarely	rarely	rarely
4	frequently	male	rarely	frequently	frequently	frequently	frequently	rarely	rarely
...	...	...	...	...	...	...	...	...	...
76	frequently	male	rarely	frequently	frequently	frequently	frequently	frequently	rarely
77	rarely	male	rarely	frequently	frequently	rarely	rarely	frequently	rarely
78	frequently	male	rarely	frequently	frequently	rarely	frequently	rarely	rarely
79	frequently	male	rarely	frequently	frequently	frequently	frequently	frequently	frequently
80	frequently	male	rarely	rarely	rarely	frequently	frequently	frequently	rarely

Worked Example: Jupyter Notebook with explicit code and comments

2 Application context: purpose of the model

Explanations of context and problem to establish a narrative

Application: Personalized Advertising on Online Platforms.

This notebook documents the development process of a decision tree. This tree is to decide for online platforms whether a user receives advertising for online games or not. The users who receive advertising should be those who play online games frequently. A user's data can be used to predict whether he or she plays online games frequently or rarely.

Classification problem

With the present data set, a classification problem with a target variable and different predictor variables can be formulated for the task just described.

- **target variable:**

  Playing_OnlineGames

- **predictor variables:**

  88 Variables about media use

should predict the expression of the target variable Playing_OnlineGames. The prediction should be based on other data of the user (e.g. personal data, Youtube user behavior, use of online platforms, ...).

The target variable currently has 7 values (7 - 1 or daily - never). However, we only want to make a prediction about whether a user plays online games **frequently** or **rarely**. Therefore, it was necessary to recode the target variable (section 3.2).

Worked Example: Jupyter Notebook with explicit code and comments

Explicit code with explanations and reasoning

3.2 Recode variables

```
#recoding the target variable
df_jim['Playing_OnlineGames'].replace([7,6,5],'frequently')
df_jim['Playing_OnlineGames'].replace([4,3,2,1],'rarely')
```

Rationale and explanation: recode target variable.

The target variable is recoded because for our prediction model we only want to know if someone plays online "Frequently" or "Rarely" in order to make a decision about placing ads. How frequent it is in detail (daily, once a week, ...) does not interest us at all for this application. A target variable with two values is also easier to predict. We therefore summarize the original values as follows:

```
7, 6, 5 --> Frequently

4, 3, 2, 1 --> Rarely
```

Worked Example: Jupyter Notebook with explicit code and comments

```
#Evaluation with test data
tree.prediction_accuracy(data_jim_test, row_percentage=True, absolute_no = True)
```

prediction correct	Häufig	Selten
Häufig	77.3%	22.7%
Selten	12.0%	88.0%

prediction correct	Häufig	Selten
Häufig	58	17
Selten	9	66

Explicit code and visualisations with missing comments, so that Students have to find self explanations

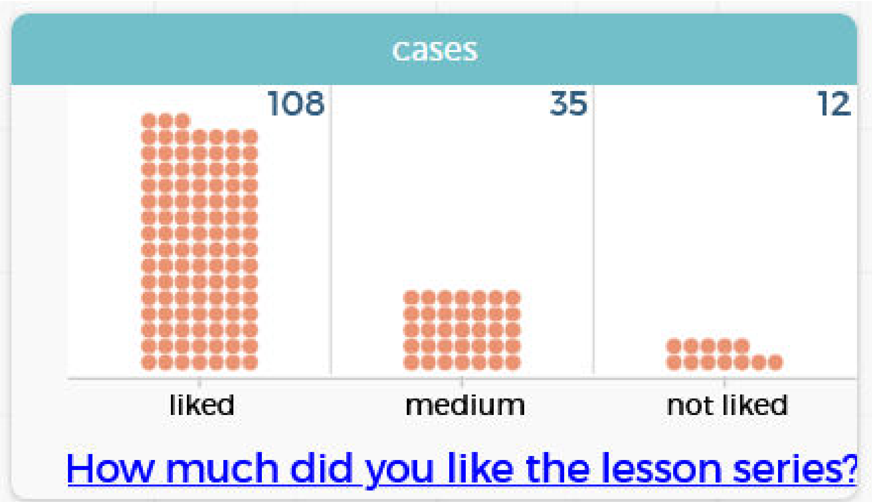
Comment about evaluation

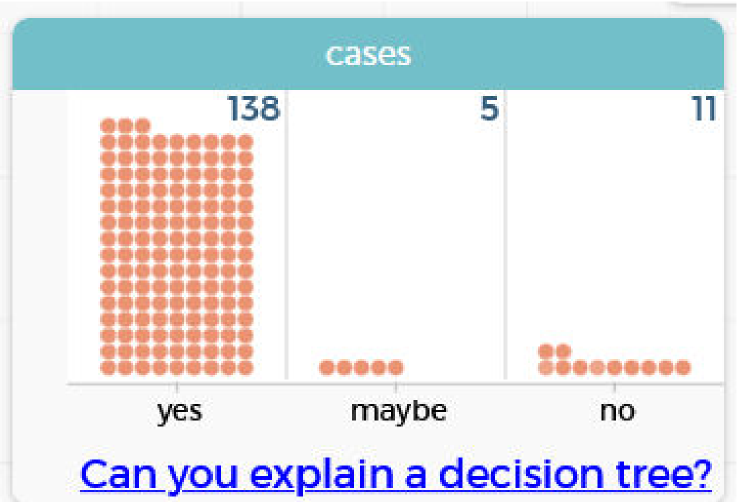
...

4. Some evalutions from students

Brief evaluation of data cards module

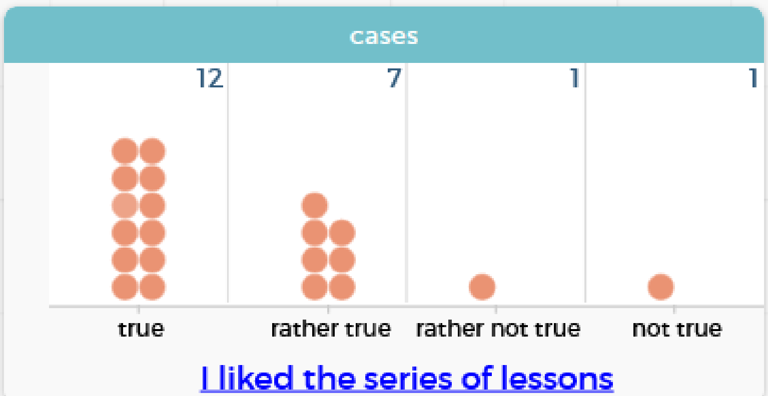
n=156 students, grade 6, age 11-12

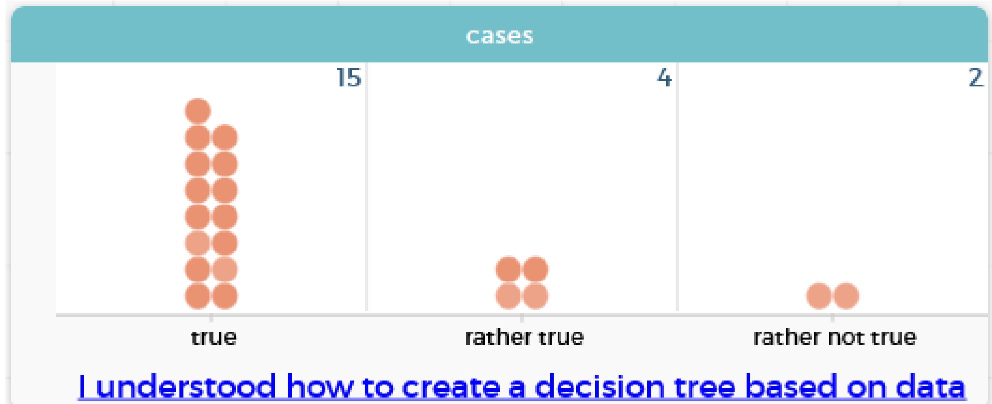


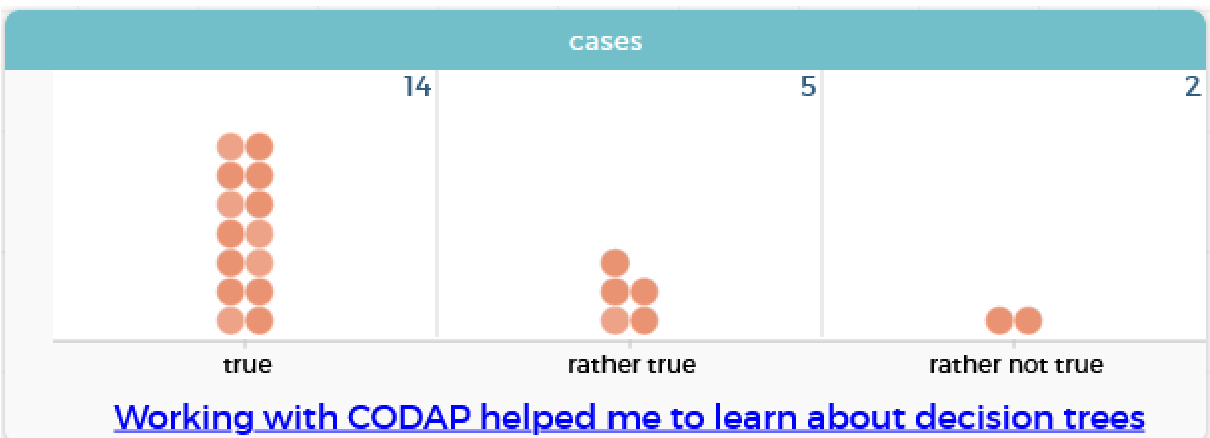


Brief evaluation of CODAP module

with n=21 students, grade 9, age 14-15







5. Looking back: Tools and facets of modelling at different levels

UNIVERSITÄT PADERBORN

ProDaBi
Projekt Data Science und Big Data in der Schule

Deutsche Telekom Stiftung

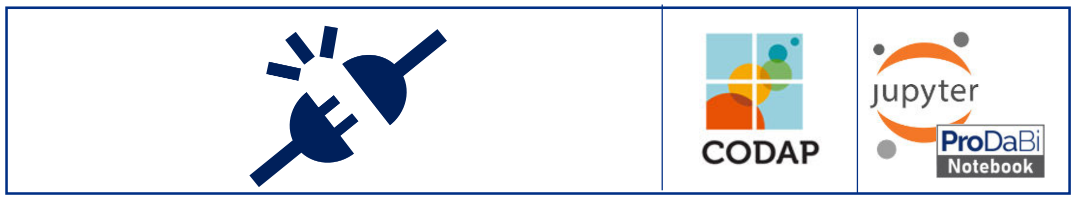
Use of tools for different school levels

Grade 5 - 6: Basic		jupyter ProDaBi Notebook
Grade 8 - 10: Standard	CODAP	jupyter ProDaBi Notebook
Grade 12: Advanced	python Course; CODAP	jupyter ProDaBi Notebook; jupyter python

UNIVERSITÄT PADERBORN
ProDaBi Projekt Data Science und Big Data in der Schule
Deutsche Telekom Stiftung

Use of tools for different school levels

Grade 5 - 6:



Grade 8 - 10:

Grade 12:

Thank you very much for your attention!

www.prodabi.de/en

Literature

Biehler, R., & Fleischer, Y. (2021). Introducing students to machine learning with decision trees using codap and jupyter notebooks. *Teaching Statistics*, *43*(S1). https://doi.org/10.1111/test.12279

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 (Eds.), *Paderborn symposium on data science education at school level 2017: The collected extended abstracts* (pp. 2–14). Universitätsbibliothek Paderborn.

Ridgway, J., Ridgway, R., & Nicholson, J. (2018). Data science for all : A stroll in the foothills. In *Looking back, looking forward : Proceedings of the tenth international conference on teaching statistics (icots10 in, July, 2018), kyoto, Japan* (pp. 1–6).

Sulmont, E., Patitsas, E., & Cooperstock, J. R. (2019a). Can you teach me to machine learn? In E. K. Hawthorne, M. A. Pérez-Quiñones, S. Heckman, & J. Zhang (Eds.), *Proceedings of the 50th acm technical symposium on computer science education* (pp. 948–954). ACM.

Sulmont, E., Patitsas, E., & Cooperstock, J. R. (2019b). What is hard about teaching machine learning to non-majors? Insights from classifying instructors' learning goals. *ACM Transactions on Computing Education*, *19*(4), 1–16.

Andrew Zieffler, Nicola Justice, Robert delMas & Michael D. Huberty (2021): The Use of Algorithmic Models to Develop Secondary Teachers' Understanding of the Statistical Modeling Process, *Journal of Statistics and Data Science Education*, DOI:10.1080/26939169.2021.1900759