Assignment 5:

Patrick Schneefuss 2951267, Jonas Lammert 3149269, Alexander Tiessen 2965198

Task 1:

Level 0:

Root (+ : 115, - : 125) Entropy = - (115/240 * log2(115/240) + 125/240 * log2(125/240)) = 0.9987

pick F1: F1 = 0: (+ : 50, - : 70) Entropy = - (50/120 * log2(50/120) + 70/120 * log2(70/120)) = 0.9799

F1 = 1: (+ : 65, - : 55) Entropy = - (65/120 * log2(65/120) + 55/120 * log2(55/120)) = 0.995

IG = 0.9987 - (1/2 * 0.9799 + 1/2 * 0.995) = 0.01125

pick F2: F2 = 0: (+ : 70, - : 50) Entropy = - (70/120 * log2(70/120) + 50/120 * log2(50/120)) = 0.9799

F2 = 1: (+ : 45, - : 75) Entropy = - (45/120 * log2(45/120) + 75/120 * log2(75/120)) = 0.9544

IG = 0.9987 - (1/2 * 0.9799 + 1/2 * 0.9544) = 0.03155

pick F3: F3 = 0: (+ : 15, - : 65) Entropy = - (15/80 * log2(15/80) + 65/80 * log2(65/80)) = 0.6962

F3 = 1: (+ : 30, - : 50) Entropy = - (30/80 * log2(30/80) + 50/80 * log2(50/80)) = 0.9544

F3 = 2: (+ : 70, - : 10) Entropy = - (70/80 * log2(70/80) + 10/80 * log2(10/80)) = 0.5436

IG = 0.9987 - (1/3 * 0.6962 + 1/3 * 0.9544 + 1/3 * 0.5436) = 0.2673

pick F4: F4 = 0: (+ : 50, - : 70) Entropy = - (50/120 * log2(70/120) + 70/120 * log2(50/120)) = 0.9799

F4 = 1: (+ : 70, - : 50) Entropy = - (50/120 * log2(70/120) + 70/120 * log2(50/120)) = 0.9799

IG = 0.9987 - (0.9799) = 0.0188

So we pick F3 as attribute for Root, highest IG.

Level 1:

Now let child with F3 = 0 be child 1, same procedure:

child 1: (+ : 15, - : 65) Entropy = - (15/80 * log2(15/80) + 65/80 * log2(65/80)) = 0.6962

pick F1: F1 = 0: (+ : 5, - : 35) Entropy = - (5/40 * log2(5/40) + 35/40 * log2(35/40)) = 0.5436

F1 = 1: (+ : 10, - : 30) Entropy = - (10/40 * log2(10/40) + 30/40 * log2(30/40)) = 0.8113

IG = 0.6962 - (1/2 * 0.5436 + 1/2 * 0.8113) = 0.01875

pick F2: F2 = 0: (+ : 15, - : 25) Entropy = - (15/40 * log2(15/40) + 25/40 * log2(25/40)) = 0.9544

F2 = 1: (+ : 0, - : 40) Entropy = 0

IG = 0.6962 - (1/2 * 0 + 1/2 * 0.9544) = 0.219

pick F4: F4 = 0: (+ : 10, - : 30) Entropy = - (10/40 * log2(10/40) + 30/40 * log2(30/40)) = 0.8113

F4 = 1: (+ : 5, - : 35) Entropy = - (5/40 * log2(5/40) + 35/40 * log2(35/40)) = 0.5436

IG = 0.6962 - (1/2 * 0.8113 + 1/2 * 0.5436) = 0.01875

So for child 1 we pick F2

Now let child with F3 = 1 be child 2 same procedure:

child 2: (+ : 30, - : 50) Entropy = - (30/80 * log2(30/80) + 50/80 * log2(50/80)) = 0.9544

pick F1: F1 = 0: (+ : 15, - : 25) Entropy = - (15/40 * log2(15/40) + 25/40 * log2(25/40)) = 0.9544

F1 = 1: (+ : 15, - : 25) Entropy = - (15/40 * log2(15/40) + 25/40 * log2(25/40)) = 0.9544

IG = 0.9544 - 0.9544 = 0

pick F2: F2 = 0: (+ : 15, - : 25) Entropy = - (15/40 * log2(15/40) + 25/40 * log2(25/40)) = 0.9544

F2 = 1: (+ : 15, - : 25) Entropy = - (15/40 * log2(15/40) + 25/40 * log2(25/40)) = 0.9544

IG = 0.9544 - 0.9544 = 0

pick F4: F4 = 0: (+ : 10, - : 30) Entropy = - (10/40 * log2(10/40) + 30/40 * log2(30/40)) = 0.8113

F4 = 1: (+ : 20, - : 20) Entropy = - (20/40 * log2(20/40) + 20/40 * log2(20/40)) = 1

IG = 0.9544 - (1/2 * 0.8113 + 1/2 * 1) = 0.04875

So for child 2 we pick F4

Now child with F3 = 2 be child 3 same procedure:

child 3: (+ : 70, - : 10) Entropy = - (70/80 * log2(70/80) + 10/80 * log2(10/80)) = 0.5436

pick F1: F1 = 0: (+ : 30, - : 10) Entropy = - (30/40 * log2(30/40) + 10/40 * log2(10/40)) = 0.8113

F1 = 1: (+ : 40, - : 0) Entropy = 0

IG = 0.5436 - (1/2 * 0.8113) = 0.13795

pick F2: F2 = 0: (+ : 40, - : 0) Entropy = 0

F2 = 1: (+ : 30, - : 10) Entropy = - (30/40 * log2(30/40) + 10/40 * log2(10/40)) = 0.8113

IG = 0.5436 - (1/2 * 0.8113) = 0.13795

pick F4: F4 = 0: (+ : 30, - : 10) Entropy = - (30/40 * log2(30/40) + 10/40 * log2(10/40)) = 0.8113

F4 = 1: (+ : 40, - : 0) Entropy = 0

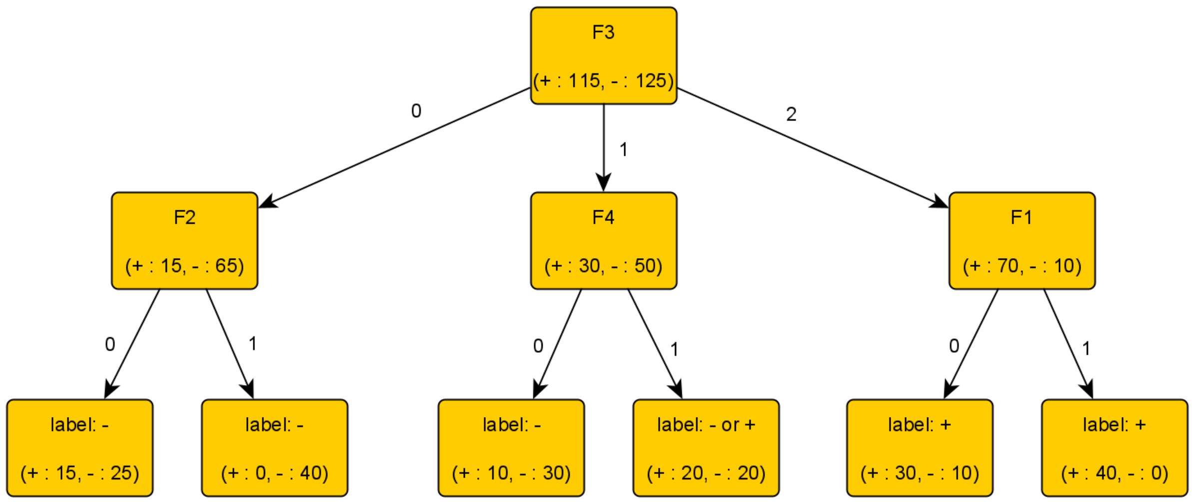
IG = 0.5436 - (1/2 * 0.8113) = 0.13795

11

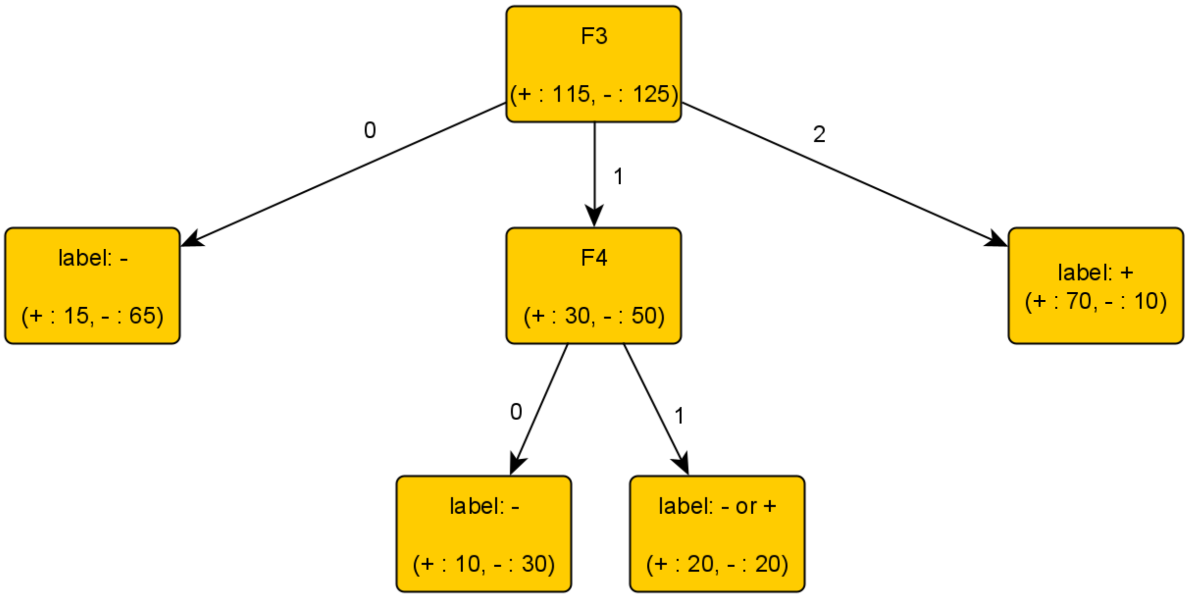
So for child 3 it does not matter, we just pick F1 here

Level 2:

Each of the thre nodes from level 1 has two child nodes. Their label and the number of samples for this node can be seen in the Calculations for level 1 and in the picture of the tree. For the right child of node F4 either – or + can be chosen, both choices yield the same prediction error.



Error = ((15 + 10 + 20 + 10)/240) = 55/240 = 11/48 = 0,229. The tree can be simplified to the tree seen below without any impact on the prediction error.


F3
(+ : 115, - : 125)
0
1
2
label: -
(+ : 15, - : 65)
F4
(+ : 30, - : 50)
label: +
(+ : 70, - : 10)
0
1
label: -
(+ : 10, - : 30)
label: - or +
(+ : 20, - : 20)

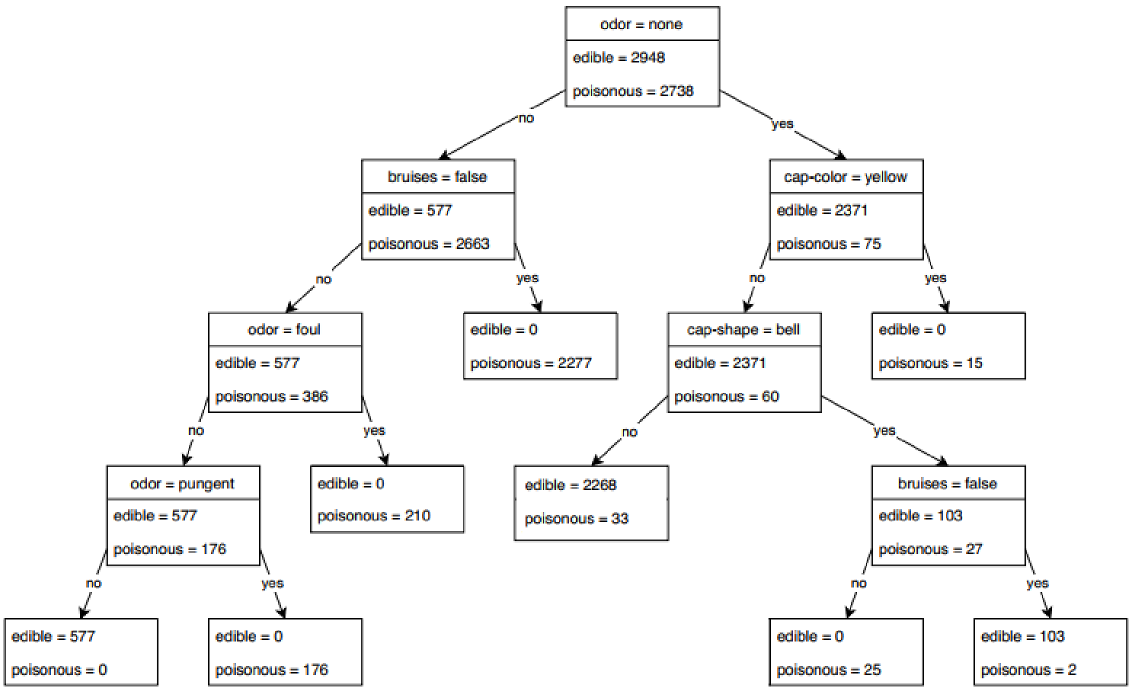
Task 2:

Total error for the tree is : (0 + 0 + 24 + 9 + 0 + 2 + 60 + 27)/(2948+2738) = 122/5686 = 0.021

Viable Nodes for step 1 are "odor = pungent", "cap-color" = pink and "bruises = false"

- "odor = pungent" : (122 + 176) / 5686 = 0,0524
- "cap-color = pink" : 122 / 5686 = 0.021
- "bruises = false" : (122 + 25)/5686 = 0.026

  - ➔ Remove Cap-color = pink, lowest prediction error of the three (stays unchanged to original tree in this case)

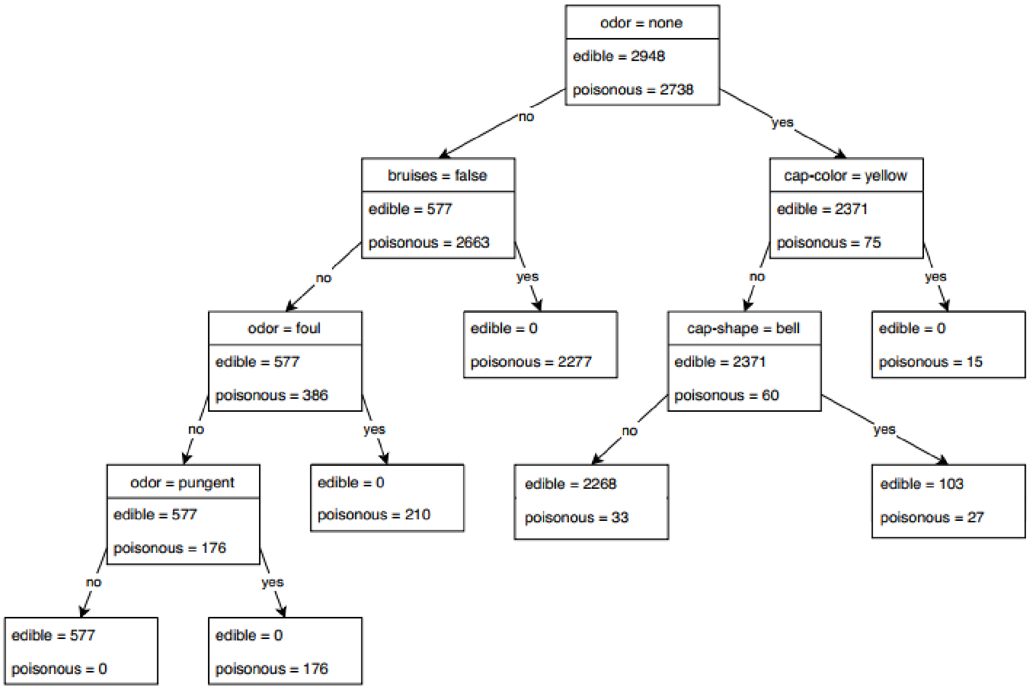


Viable Nodes for step 2 are "odor = pungent" and "bruises = false"

Since error remained the same, error for pruned trees also remains the same

- "odor = pungent" : (122 + 176) / 5686 = 0,0524
- "bruises = false" : (122 + 25)/5686 = 0.026

  - ➔ Remove bruises = false, lowest prediction error



odor = none
edible = 2948
poisonous = 2738
no
yes
bruises = false
edible = 577
poisonous = 2663
cap-color = yellow
edible = 2371
poisonous = 75
no
yes
no
yes
odor = foul
edible = 577
poisonous = 386
edible = 0
poisonous = 2277
cap-shape = bell
edible = 2371
poisonous = 60
edible = 0
poisonous = 15
no
yes
no
yes
odor = pungent
edible = 577
poisonous = 176
edible = 0
poisonous = 210
edible = 2268
poisonous = 33
edible = 103
poisonous = 27
no
yes
edible = 577
poisonous = 0
edible = 0
poisonous = 176

Task3:

1. How does the construction of regression trees differ to classification trees? How is a prediction computed in a regression tree?

In a regression tree, each leaf node represents a numeric value. This numeric value is generally the average of the subgroup of datapoints it is assigned to.
In contrast, classification trees have a binary variable or a discreet class representation in there leaves.
The decision threshold of a node is selected by the lowest sum of squared residuals.

A prediction in a regression tree is done like in a classification tree: The tree is traversed down to a leaf node based on the used features. The difference is that the leaf node is assigned to a continuous value and not to a class.

Sources:

Yisehac Yohannes & Patrick Webb: "Classification and Regression Trees, Cart: A User Manual for Identifying Indicators of Vulnerability to Famine and Chronic Food Insecurity" Intl Food Policy Research Ins, Dezember 1998

StatQuest with Josh Starmer: "Regression Trees, Clearly Explained!!!", 2019,
Url: https://www.youtube.com/watch?v=g9c66TUylZ4

University of Cincinnati Business Analytics R Programming Guide: "Regression Trees"
Url: https://uc-r.github.io/regression_trees

2. How can kNN be used for regression?

In kNN for regression a prediction is made via the nearest neighbors of the training sample in respect to a line in the feature space. This line represents the feature values at which a prediction should be made. The average of the kNN is taken to determine a predicted value. This results in a simple way to predict even more complex (nonlinear) functions. A in kNN for classification, this doesn't require a dedicated training step.

Max Miller: "The Basics: KNN for classification and regression: Building an intuition for how KNN models work", towards datascience, October 2018
Url: https://towardsdatascience.com/the-basics-knn-for-classification-and-regression-c1e8a6c955