## Foundations of Artificial Intelligence

Prof. Dr. J. Boedecker, Prof. Dr. W. Burgard, Prof. Dr. F. Hutter, Prof. Dr. B. Nebel, Dr. rer. nat. M. Tangermann
T. Schulte, M. Krawez, R. Rajan, S. Adriaensen, K. Sirohi
Summer Term 2020

University of Freiburg
Department of Computer Science

# Exercise Sheet 10 — Solutions

**Exercise 10.1** (Decision Trees)

| No | Age | Engine power [kW] | **Risk** |
|---|---|---|---|
| 1 | $< 25$ | $< 100$ | low |
| 2 | $< 25$ | $> 200$ | high |
| 3 | $\geq 25$ | $> 200$ | high |
| 4 | $\geq 25$ | $100 - 200$ | low |
| 5 | $< 25$ | $100 - 200$ | high |
| 6 | $\geq 25$ | $< 100$ | low |

Consider the data on car insurance risk in the table above. Produce a decision tree, which correctly classifies the insurance risk for the examples given, using the attributes *Age* and *Engine Power* in order of decreasing *information gain*. Give detailed calculations that justify the order in which the attributes are tested.
You can make use of the following values:
$\log_2(\frac{1}{3}) \approx -\frac{3}{2}$, $\log_2(\frac{2}{3}) \approx -\frac{1}{2}$, $\log_2(\frac{1}{2}) = -1$, $\log_2(1) = 0$.

**Solution:**

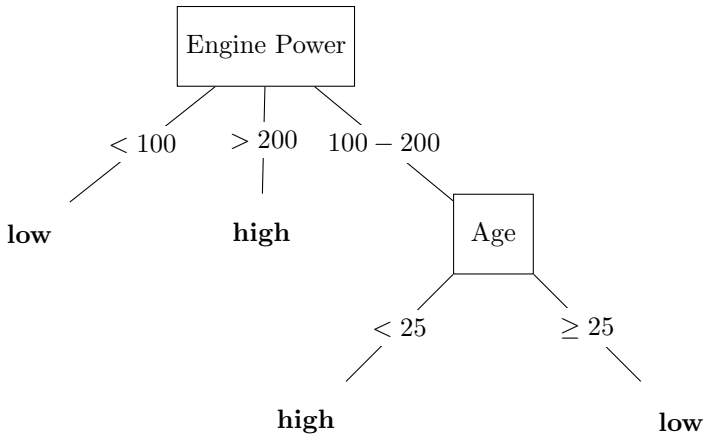
Entropy of the root node: $I(Risk) = I(\frac{1}{2}, \frac{1}{2}) = 1$
Remaining uncertainties after splitting on the different attributes:

$$R(EnginePower) = \frac{1}{3} \cdot I(1,0) + \frac{1}{3} \cdot I(0,1) + \frac{1}{3} \cdot I(\frac{1}{2}, \frac{1}{2}) = \frac{1}{3}$$
$$R(Age) = \frac{1}{2} \cdot I(\frac{1}{3}, \frac{2}{3}) + \frac{1}{2} \cdot I(\frac{2}{3}, \frac{1}{3}) = I(\frac{1}{3}, \frac{2}{3}) = -\frac{1}{3}\log_2(\frac{1}{3}) - \frac{2}{3}\log_2(\frac{2}{3}) = \frac{5}{6}$$

Gains are then:

$$Gain(EnginePower) = 1 - \frac{1}{3} = \frac{2}{3}$$
$$Gain(Age) = 1 - \frac{5}{6} = \frac{1}{6}$$

So the first split should be on attribute *Engine Power*. After that, splitting on *Age* will result in a clean split with no entropy left.




**Exercise 10.2** (Best practices in ML)

When doing machine learning, it is good practice to split the dataset into a training/validation/test set.

- Which subset(s) should you use for the following tasks:
  (a) training models (R & D)[1]
  (b) guard against overfitting (R & D)
  (c) model selection (R & D)
  (d) progress reports (R & D)
  (e) train the model (product)[2]
  (f) evaluating the model (product)
- Which of these subsets should always be fixed a priori (before even looking at the data)?

**Solution:**

- 

| **task** | **phase** | **training set** | **validation set** | **test set** |
|---|---|---|---|---|
| training models | R & D | yes | no | no |
| guard against over-fitting | R & D | no | yes | no |
| model selection | R & D | no | yes | no |
| progress reports | R & D | no | yes | no |
| training the model | product | yes | maybe | no |
| evaluating the model | product | no | no | yes |

- The test set should always be fixed a priori (and used only once, to evaluate the final model). Instances in validation/training sets may vary during R & D. However, having a sparsely-used, fixed subset that acts as a 'pseudo' test-set can be useful (e.g. for internal progress reports). Also, if examples in the validation set are frequently used (e.g during training, hyper-parameter tuning, etc.) failing to detect overfitting is a real risk.

[1] R & D: During research and development
[2] product: For the final product/publication