Exercise collection – CART

Contents

Lecture exercises

Exercise 1: bagging and correlation

Show that the variance of the bagging prediction depends on the correlation between trees.
Hint: compute $\mathrm{Var}(\frac{1}{B}\sum_{b=1}^{B} f_b)$ when $\mathrm{Var}(f_b) = \sigma^2$ and $\mathrm{Corr}(f_i, f_j) = \rho$, where f_b is a single tree of the ensemble.

Solution 1:

$f(x) = \frac{1}{B}\sum_{b=1}^{B} f_b(x)$ is the bagging estimator based on B bootstrap samples. Then we can easily calculate:

$$\begin{aligned}
\mathrm{Var}(f(x)) &= \frac{1}{B^2}\left(\sum_{b=1}^{B}\mathrm{Var}(f_b(x)) + \sum_{i\neq j}^{B}\mathrm{Cov}(f_i(x), f_j(x))\right) \\
&= \frac{1}{B^2}\left(B\sigma^2 + (B^2 - B)\rho\sigma^2)\right) \\
&= \frac{1}{B}\sigma^2 + \rho\sigma^2 - \frac{1}{B}\rho\sigma^2 \\
&= \rho\sigma^2 + \frac{\sigma^2}{B}(1-\rho)
\end{aligned}$$

In the first line the rules for variance of a non-independent sum of random variables is used. All other steps are trivial.

Exercise 2: split computation

Given are the dataset

x	1	2	7.0	10	20
y	1	1	0.5	10	11

and the same dataset, but with the feature x log-transformed

log(x)	0	0.7	1.9	2.3	3
y	1	1.0	0.5	10.0	11

Either manually compute the first split point that the CART algorithm would find for each dataset or implement your own CART split-point-finding algorithm with a few lines of code.

Solution 2:

- Proceed as follows, when solving manually:
  (a) Split x in two groups using the following split points.
    - (1), $(2, 7, 10, 20)$ (splitpoint 1.5)
    - $(1, 2)$, $(7, 10, 20)$ (splitpoint 4.5)
    - $(1, 2, 7)$, $(10, 20)$ (splitpoint 8.5)
    - $(1, 2, 7, 10)$, (20) (splitpoint 15)

  (b) For each possible split point compute the sum of squares in both groups.

  (c) Use as split point the point that splits both groups best w.r.t. minimizing the sum of squares in both groups.

  Here, we have only one split variable x. A split point t, leads to the following half-spaces:

$$\mathcal{N}_1(t) = \{(x, y) \in \mathcal{N} : x \leq t\} \text{ and } \mathcal{N}_2(t) = \{(x, y) \in \mathcal{N} : x > t\}.$$

  Remember the minimization Problem (here only for one split variable x):

$$\min_t \left(\min_{c_1} \sum_{(x,y)\in\mathcal{N}_1} (y - c_1)^2 + \min_{c_2} \sum_{(x,y)\in\mathcal{N}_2} (y - c_2)^2 \right).$$

  The inner minimization is solved through: $\hat{c}_1 = \bar{y}_1$ and $\hat{c}_2 = \bar{y}_2$

  Which results in:

$$\min_t \left(\sum_{(x,y)\in\mathcal{N}_1} (y - \bar{y}_1)^2 + \sum_{(x,y)\in\mathcal{N}_2} (y - \bar{y}_2)^2 \right).$$

  The sum of squares error of the parent is:

$$Impurity_{parent} = MSE_{parent} = \frac{1}{5} \sum_{i=1}^{5} (y_i - 4.7)^2 = 22.56$$

  Calculate the risk for each split point:

$x \leq 1.5$

$$\begin{aligned}\mathcal{R}(1, 1.5) &= \frac{1}{5}\text{MSE}_{left} + \frac{4}{5}\text{MSE}_{right} = \\ &= \frac{1}{5} \cdot \frac{1}{1}(1-1)^2 + \frac{4}{5} \cdot \frac{1}{4}((1-5.625)^2 + (0.5-5.625)^2 + (10-5.625)^2 + (11-5.625)^2) \\ &= 19.1375\end{aligned}$$

$x \leq 4.5 \;\; \mathcal{R}(1, 4.5) = 13.43$

$x \leq 8.5 \;\; \mathcal{R}(1, 8.5) = 0.13$

$x \leq 15 \;\; \mathcal{R}(1, 15) = 12.64$

Minimal empirical risk is obtained by choosing the split point 8.5.

Doing the same for the log-transformation gives:

$x \leq 0.3 \;\; \mathcal{R}(1, 0.3) = 19.14$

$x \leq 1.3 \;\; \mathcal{R}(1, 1.3) = 13.43$

$x \leq 2.1 \;\; \mathcal{R}(1, 2.1) = 0.13$

$x \leq 2.6 \;\; \mathcal{R}(1, 2.6) = 12.64$

Minimal empirical risk is obtained by choosing the split point 2.1.

- Code example:

```r
x = c(1,2,7,10,20)
y = c(1,1,0.5,10,11)

calculate_mse <- function (y) mean((y - mean(y))^2)
calculate_total_mse <- function (yleft, yright) {
  num_left <- length(yleft)
  num_right <- length(yright)

  w_mse_left <- num_left / (num_left + num_right) * calculate_mse(yleft)
  w_mse_right <- num_right / (num_left + num_right) * calculate_mse(yright)

  return(w_mse_left + w_mse_right)
}

split <- function(x, y) {
  # try out all unique points as potential split points and ...
  unique_sorted_x <- sort(unique(x))
  split_points <- unique_sorted_x[1:(length(unique_sorted_x) - 1)] +
    0.5 * diff(unique_sorted_x)
  node_mses <- lapply(split_points, function(i) {
    y_left <- y[x <= i]
    y_right <- y[x > i]

    # ... compute SS in both groups
    mse_split <- calculate_total_mse(y_left, y_right)
    print(sprintf("Split at %.1f: empirical Risk = %.2f", i, mse_split))

    return(mse_split)
  })
  # select the split point yielding the maximum impurity reduction
  best <- which.min(node_mses)
  split_points[best]
}

x
```

```
## [1]  1  2  7 10 20

split(x, y) # the 3rd observation is the best split point

## [1] "Split at 1.5: empirical Risk = 19.14"
## [1] "Split at 4.5: empirical Risk = 13.43"
## [1] "Split at 8.5: empirical Risk = 0.13"
## [1] "Split at 15.0: empirical Risk = 12.64"
## [1] 8.5

log(x)

## [1] 0.0000000 0.6931472 1.9459101 2.3025851 2.9957323

split(log(x), y) # also here, the 3rd observation is the best split point

## [1] "Split at 0.3: empirical Risk = 19.14"
## [1] "Split at 1.3: empirical Risk = 13.43"
## [1] "Split at 2.1: empirical Risk = 0.13"
## [1] "Split at 2.6: empirical Risk = 12.64"
## [1] 2.124248
```

Exercise 3: lymphography classification

Download the lymphography dataset from moodle.

(a) Download the file lymphography.csv from moodle and read it in using `read.csv()`

(b) Have a short look into the background and structure of the data.

(c) Delete 6 observations from the smallest class, so the resulting problem is binary classification.

Now fit CART (from `rpart`) and a second model of your choice to the data and answer the following questions:

- How "stable" are the resulting trees from the CART model?
- How do the results differ between pruned and unpruned trees?
- Is one of the two methods better suited for the data?

Solution 3:

See R code

Exercise 4: leaf node prediction

The fractions of the classes $k = 1, \ldots, g$ in node $\mathcal{N}$ of a decision tree are $\pi_1^{(\mathcal{N})}, \ldots, \pi_g^{(\mathcal{N})}$. Assume we replace the classification rule in node $\mathcal{N}$

$$\hat{k}|\mathcal{N} = \arg\max_k \pi_k^{(\mathcal{N})}$$

with a randomizing rule, in which we draw the classes in one node from their estimated probabilities.

Compute the expectation of the misclassification rate in node $\mathcal{N}$, for data distributed like the training data, assuming independent observations. What do you notice? (*Hint*: The observations and the predictions using the randomizing rule follow the same distribution.)

Solution 4:

According to the lecture for a target y with target space $\mathcal{Y} = \{1, \ldots, g\}$ the target class proportion $\pi_k^{(\mathcal{N})}$ of class $k \in \mathcal{Y}$ in a node can be computed, s.t.

$$\pi_k^{(\mathcal{N})} = \frac{1}{|\mathcal{N}|} \sum_{(x^{(i)}, y^{(i)}) \in \mathcal{N}} [y^{(i)} = k].$$

Now for any $n \in \mathbb{N}$ let $Y^{(1)}, \ldots, Y^{(n)}, \hat{Y}^{(1)}, \ldots, \hat{Y}^{(n)}$ be i.i.d. random variables, where $Y^{(i)}$ and $\hat{Y}^{(i)}$ are categorically distributed with

$$\mathbb{P}(Y^{(i)} = k|\mathcal{N}) = \mathbb{P}(\hat{Y}^{(i)} = k|\mathcal{N}) = \pi_k^{(\mathcal{N})} \quad \forall i \in \{1, \ldots, n\}, \quad k \in \mathcal{Y}.$$

The random variables $Y^{(1)}, \ldots, Y^{(n)}$ represent data distributed like the training data[1] of size n and the random variables $\hat{Y}^{(1)}, \ldots, \hat{Y}^{(n)}$ the corresponding estimators using the randomizing rule. With these we can define the misclassification rate $\mathsf{err}_{\mathcal{N}}$ of node $\mathcal{N}$ for data distributed like the training data, s.t

$$\mathsf{err}_{\mathcal{N}} = \frac{1}{n} \sum_{i=1}^{n} [Y^{(i)} \neq \hat{Y}^{(i)}].$$

We're interested in the expected misclassification rate $\mathsf{err}_{\mathcal{N}}$ of node $\mathcal{N}$ for data distributed like the training data,

[1] under the independence assumption

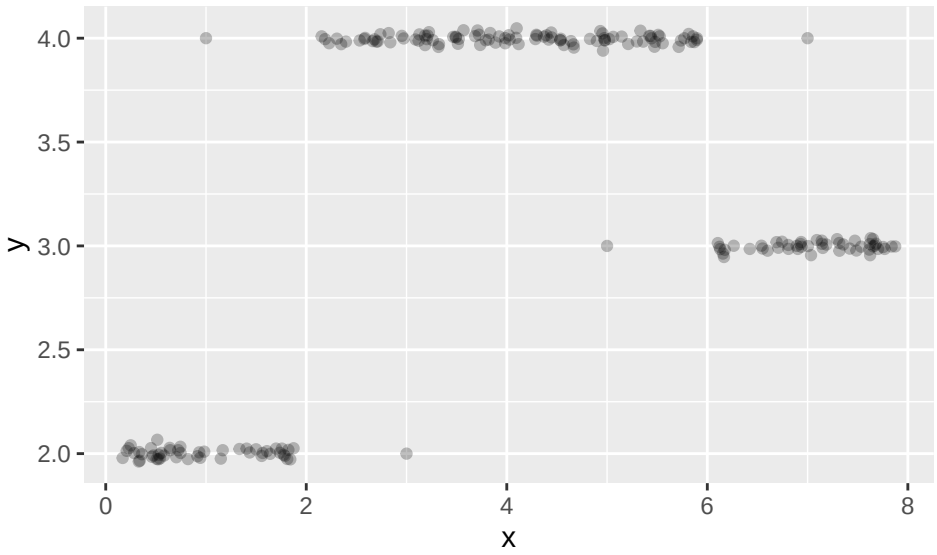
i.e.,

$$
\begin{aligned}
\mathbb{E}_{Y^{(1)},\ldots,Y^{(n)},\hat{Y}^{(1)},\ldots,\hat{Y}^{(n)}}\left(\mathsf{err}_{\mathcal{N}}\right) &= \mathbb{E}_{Y^{(1)},\ldots,Y^{(n)},\hat{Y}^{(1)},\ldots,\hat{Y}^{(n)}}\left(\frac{1}{n}\sum_{i=1}^{n}[Y^{(i)} \neq \hat{Y}^{(i)}]\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{Y^{(i)},\hat{Y}^{(i)}}\left([Y^{(i)} \neq \hat{Y}^{(i)}]\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{Y^{(i)}}\left(\mathbb{E}_{\hat{Y}^{(i)}}\left([Y^{(i)} \neq \hat{Y}^{(i)}]\right)\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{Y^{(i)}}\left(\sum_{k=1}^{g}[Y^{(i)} \neq k]\pi_k^{(\mathcal{N})}\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{Y^{(i)}}\left(\sum_{k\in\mathcal{Y}\setminus\{Y^{(i)}\}}\pi_k^{(\mathcal{N})}\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_{Y^{(i)}}\left(1-\pi_{Y^{(i)}}^{(\mathcal{N})}\right) \\
&= \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{g}(1-\pi_k^{(\mathcal{N})})\pi_k^{(\mathcal{N})} \\
&= \frac{n}{n}\sum_{k=1}^{g}(1-\pi_k^{(\mathcal{N})})\pi_k^{(\mathcal{N})} \\
&= 1-\sum_{k=1}^{g}\left(\pi_k^{(\mathcal{N})}\right)^2.
\end{aligned}
$$

This is exactly the Gini-Index which CART uses for splitting the tree.

Questions from past exams

Exercise 5: WS2020/21, main exam, question 1



The above plot shows $\mathcal{D} = \left(\left(\mathbf{x}^{(1)}, y^{(1)}\right), \ldots, \left(\mathbf{x}^{(n)}, y^{(n)}\right)\right)$, a data set with $n = 200$ observations of a continuous target variable y and a continuous, 1-dimensional feature variable $\mathbf{x}$. In the following, we aim at predicting y with a machine learning model that takes $\mathbf{x}$ as input.

(a) Assume we trained a regression tree with L1 loss $L\left(y, f(\mathbf{x})\right) = |y - f(\mathbf{x})|$. The training resulted in two splits of the feature $\mathbf{x}$ which means that the resulting estimated model is

$$\begin{aligned}\hat{f}(\mathbf{x}) &= \sum_{m=1}^{3} \hat{c}_m \mathbb{I}(\mathbf{x} \in \hat{Q}_m) \\ &= \hat{c}_1 \mathbb{I}(\mathbf{x} \in (-\infty, \hat{q}_1]) + \hat{c}_2 \mathbb{I}(\mathbf{x} \in (\hat{q}_1, \hat{q}_2]) + \hat{c}_3 \mathbb{I}(\mathbf{x} \in (\hat{q}_2, \infty))\end{aligned}$$

(i) Estimate the two split points $\hat{q}_1$ and $\hat{q}_2$ visually from the plot.

(ii) Estimate the three predicted labels $\hat{c}_1$, $\hat{c}_2$ and $\hat{c}_3$ visually from the plot.

(iii) How would the estimated model change if we used L2 loss $L\left(y, f(\mathbf{x})\right) = 0.5(y - f(\mathbf{x}))^2$ instead of L1 loss? State for each split point $\hat{q}_1$ and $\hat{q}_2$ and for each predicted label $\hat{c}_1$, $\hat{c}_2$ and $\hat{c}_3$

- if it changes and
- if it changes, in which direction it changes

and explain your decision thoroughly.

(b) Given are two new observations $\mathbf{x}_{*1} = -10$ and $\mathbf{x}_{*2} = 7$. State the prediction for each of the two models

(i) regression tree

(ii) QDA

and explain how you derived the predictions.

(c) Discuss in 1-2 sentences which of the 2 models (regression tree, QDA) you would prefer for modeling the data and explain your decision.

Solution 5:

(a)

(i) $\hat{q}_1 = 2$, $\hat{q}_2 = 6$

(ii) $\hat{c}_1 = 2$, $\hat{c}_2 = 4$, $\hat{c}_3 = 3$

(iii) $\hat{q}_1$ and $\hat{q}_2$ would not change since including more 'wrong' points is making end notes less pure, regardless of the loss. (Basically no explanation needed for full point, since it does not change.) $\hat{c}_1$ and $\hat{c}_3$ would be higher since the L2 loss gives more weight to extreme observations. $\hat{c}_2$ would be lower since the L2 loss gives more weight to extreme observations.

(b)

(i) $\hat{y}_{*1} = 2$, $\hat{y}_{*2} = 3$,

(ii) $\hat{z}_{*1} = 3$, since the variance of class 3 is higher, the density will overshoot the density of class 1. $\hat{z}_{*2} = 2$, obviously highest posterior here.

(c) E.g.,

- CART better than LM because I do not have to specify those indicator functions manually and estimate the split points manually, CART does this data driven
- For QDA we have to throw away information of y, this favors CART
- QDA predicts the middle class (3) for very extreme observations, this does not seem right. However, we do not know how data behave outside the bounds of x.
- QDA assumes gaussian distributions which is clearly not the case.

Exercise 6: WS2020/21, main exam, question 2

The table below shows $\mathcal{D} = \left(\left(\mathbf{x}^{(1)}, y^{(1)}\right), \ldots, \left(\mathbf{x}^{(n)}, y^{(n)}\right)\right)$, a data set with $n = 5$ observations of a continuous target variable y and a continuous, 1-dimensional feature variable $\mathbf{x}$. In the following, we aim at predicting y with a machine learning model that takes $\mathbf{x}$ as input.

ID	$\mathbf{x}$	y
1	1.0	3.1
2	5.2	0.5
3	2.7	1.7
4	1.1	4.5
5	1.5	2.7

(a) We want to train a regression tree on the above data with the L1 loss $L\left(y, f(\mathbf{x})\right) = |y - f(\mathbf{x})|$. Compute the first split point $\hat{q}_1$ of the regression tree.

(b) Compute the two predicted labels $\hat{c}_1$ and $\hat{c}_2$ corresponding to the two resulting intervals from a).

(c) Predict the label y for a new observation $\mathbf{x}_* = 2$ with this regression tree and explain your calculation.

Solution 6:

(a)

	x	y
1	1.0	3.1
4	1.1	4.5
5	1.5	2.7
3	2.7	1.7
2	5.2	0.5

- Split x in two groups using the following split points.
  - (1), (1.1, 1.5, 2.7, 5.2) (splitpoint 1.05)
  - (1, 1.1), (1.5, 2.7, 5.2) (splitpoint 1.3)
  - (1, 1.1, 1.5), (2.7, 5.2) (splitpoint 2.1)
  - (1, 1.1, 1.5, 2.7), (5.2) (splitpoint 4.0)
- For each possible split point compute the empirical risk.
- Use as split point the point that splits both groups best w.r.t. minimizing the empirical risk.

$q = 1.05$

$$\begin{aligned}\mathcal{R}(1, 1, 1.05) &= 0 + \sum_{i=2}^{5} |y_i - median(y[2:5])| \\ &= 0 + \sum_{i=2}^{5} |y_i - 2.2| \\ &= 0 + 2.3 + 0.5 + 0.5 + 1.7 \\ &= 5\end{aligned}$$

$q = 1.3$

$$\begin{aligned}\mathcal{R}(1, 1, 1.3) &= \sum_{i=1}^{2} |y_i - median(y[1:2])| + \sum_{i=3}^{5} |y_i - median(y[3:5])| \\ &= \sum_{i=1}^{2} |y_i - 3.8| + \sum_{i=3}^{5} |y_i - 1.7| \\ &= 0.7 + 0.7 + 1 + 1.2 \\ &= 3.6\end{aligned}$$

$q = 2.1$

$$\begin{aligned}\mathcal{R}(1, 1, 2.1) &= \sum_{i=1}^{3} |y_i - median(y[1:3])| + \sum_{i=4}^{5} |y_i - median(y[4:5])| \\ &= \sum_{i=1}^{3} |y_i - 3.1| + \sum_{i=4}^{5} |y_i - 1.1| \\ &= 0 + 1.4 + 0.4 + 0.6 + 0.6 \\ &= 3\end{aligned}$$

$q = 4.0$

$$\begin{aligned}\mathcal{R}(1, 1, 4.0) &= \sum_{i=1}^{4} |y_i - median(y[1:4])| + 0 \\ &= \sum_{i=1}^{4} |y_i - 2.9| \\ &= 0.2 + 1.6 + 0.2 + 1.2 \\ &= 3.2\end{aligned}$$

Minimal empirical risk is obtained by choosing the split point $\hat{q} = 2.1$

(b) $\hat{c}_1 = 3.1$ and $\hat{c}_2 = 1.1$

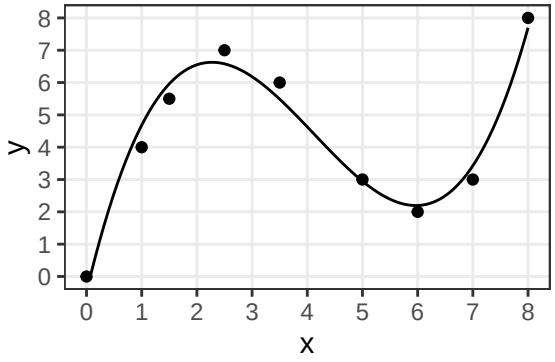
(c) Since $\mathbf{x}_* = 2 < \hat{q} = 2.1$, the observations falls in the left part, i.e., the prediction is $\hat{c}_1 = 3.8$

Exercise 7: WS2020/21, retry exam, question 1

(a) We want to compare a linear regression model with a regression tree. For the linear regression, we use the feature variable $\mathbf{x}$ without any transformations; for the regression tree, we use a fully grown tree, i.e., we set the hyperparameters $cp = 0$, $minsplit = 1$ and $minbucket = 1$ in `rpart`.

(i) State the training loss of the regression tree and explain why the regression tree would yield a better fit (i.e., a smaller training loss) to the data than the linear regression model.

(ii) Given the true underlying model is the cubical polynomial function shown with a solid line in the plot below: Which of the two models (linear regression and regression tree) would extrapolate better in the region $x \in (100, \infty)$? Explain your decision.



Solution 7:

(a) The training loss of the regression tree would be 0 because the fully grown tree would yield a step function that goes through every point. The training loss of the linear model can not be 0 since the training points can not be interpolated with a straight line.

(b) The linear model would extrapolate better because the regression tree extrapolates a constant value of 8 but the true function goes further up. As would go the estimated linear model.

Exercise 8: WS2020/21, retry exam, question 2

The table below shows $\mathcal{D} = \left(\left(\mathbf{x}^{(1)}, y^{(1)}\right), \ldots, \left(\mathbf{x}^{(n)}, y^{(n)}\right)\right)$, a data set with $n = 10$ observations of a binary target variable `PlayTennis` and two binary feature variables `Temperature` and `Weather`. In the following, we aim at predicting `PlayTennis` with a machine learning model that takes `Temperature` and `Weather` as input.

ID	1	2	3	4	5	6	7	8	9	10
Temperature	cool	cool	cool	hot	hot	cool	hot	cool	cool	hot
Weather	rain	rain	sunny	sunny	sunny	rain	rain	sunny	sunny	sunny
PlayTennis	no	no	yes	no	yes	no	yes	yes	yes	yes

(a) We want to train a classification tree on the above data using the Brier score (Gini impurity) as split criterion. Which feature will be chosen for the first split? Calculate all necessary Brier scores.

Solution 8:

(a) Calculate Brier score for both variables:

- **For temperature**:

  N1 = hot: $\hat{\pi}_{\text{no}}^{(\text{hot})} = 1/4$, $\hat{\pi}_{\text{yes}}^{(\text{hot})} = 3/4$, $Brier_{\text{hot}} = 2 \cdot 1/4 \cdot 3/4 = 3/8$

  N2 = cool: $\hat{\pi}_{\text{no}}^{(\text{cool})} = 3/6$, $\hat{\pi}_{\text{yes}}^{(\text{cool})} = 3/6$, $Brier_{\text{cool}} = 2 \cdot 1/2 \cdot 1/2 = 1/2$

  Average node impurity for temperature: $Brier_{\text{temperature}} = 4/10 \cdot 3/8 + 6/10 \cdot 1/2 = .45$

- **For Weather**:

  N1 = sunny: $\hat{\pi}_{\text{no}}^{(\text{sunny})} = 1/6$, $\hat{\pi}_{\text{yes}}^{(\text{sunny})} = 5/6$, $Brier_{\text{sunny}} = 2 \cdot 1/6 \cdot 5/6 = 10/36$

  N2 = rain: $\hat{\pi}_{\text{no}}^{(\text{rain})} = 3/4$, $\hat{\pi}_{\text{yes}}^{(\text{rain})} = 1/4$, $Brier_{\text{rain}} = 2 \cdot 3/4 \cdot 1/4 = 3/8$

  Average node impurity for weather: $Brier_{\text{weather}} = 4/10 \cdot 3/8 + 6/10 \cdot 10/36 = .317$

The split would be made on the feature *weather* because the Brier score of this feature is smaller.

Ideas & exercises from other sources