Machine Learning (SS 23)
Assignment 06: Decision Tree

Team Members:

- Likhit Jain, 3678905, M.Sc. Computer Science
- Tareq Abu El Komboz, 3405686, M.Sc. Informatik
- Serge Kotchourko, 3309449, M.Sc. Informatik

1. Decision Tree

1. Question Compute the root split of a decision tree for this data set. Make your split decision based on maximizing information gain.

Answer First, some definitions from the lectures that we are going to use to compute the required values:

$$IG = H(P_t) - \sum_{n \in c(t)} p(n) H(P_n)$$
$$H(P) = -\sum_{x \in X} p(x) \log_2(p(x))$$

Where t represents the top node, $c(\cdot)$ is a function returning all possible children of t and $x \in X$ is a class.

First, we calculate the entropy of the root:

$$\begin{aligned} H(P_t) &= -\sum_{x \in X} p(x) \log_2(p(x)) \\ &= -\left[\frac{5}{10}\log_2\left(\frac{5}{10}\right) + \frac{5}{10}\log_2\left(\frac{5}{10}\right)\right] \\ &= -\left[\frac{1}{2}\log_2\left(\frac{1}{2}\right) + \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right] \\ &= -\left[(-1) \times \frac{1}{2} + (-1) \times \frac{1}{2}\right] \\ &= 1 \end{aligned}$$

We first have to calculate the information gain for each class.

Value	Poisonous	Edible	Total	Entropy
Brown	2	1	3	$H(P_{\text{CC-Brown}}) = \frac{14}{15}$
Orange	1	1	2	$H(P_{\text{CC-Orange}}) = 1$
Red	1	1	2	$H(P_{\text{CC-Red}}) = 1$
Yellow	1	2	3	$H(P_{\text{CC-Yellow}}) = \frac{14}{15}$

Table 1: Cap Color

The resulting information gain for *Cap Color* is then:

$$\begin{aligned} IG &= H(P_t) - \sum_{n \in c(t)} p(n) H(P_n) \\ &= 1 - \left[\frac{3}{10} \times \frac{14}{15} + \frac{2}{10} \times 1 + \frac{2}{10} \times 1 + \frac{3}{10} \times \frac{14}{15} \right] \\ &= 1 - \left[\frac{42}{150} + \frac{30}{150} + \frac{30}{150} + \frac{42}{150} \right] \\ &= 1 - \frac{144}{150} = \frac{6}{150} = \frac{1}{25} = 0.04 \end{aligned}$$

Value	Poisonous	Edible	Total	Entropy
White	4	3	7	$H(P_{\text{GC-White}}) = \frac{34}{35}$
Yellow	1	2	3	$H(P_{\text{GC-Yellow}}) = \frac{14}{15}$

Table 2: Gill Color

The resulting information gain for *Gill Color* is then:

$$\begin{aligned} IG &= H(P_t) - \sum_{n \in c(t)} p(n) H(P_n) \\ &= 1 - \left[\frac{7}{10} \times \frac{34}{35} + \frac{3}{10} \times \frac{14}{15} \right] \\ &= 1 - \left[\frac{238}{350} + \frac{42}{150} \right] \\ &= 0.04 \end{aligned}$$

Value	Poisonous	Edible	Total	Entropy
Yes	2	4	6	$H(P_{\text{ST-Yes}}) = \frac{14}{15}$
No	3	1	4	$H(P_{\text{ST-No}}) = \frac{4}{5}$

Table 3: Sticky Texture

The resulting information gain for *Sticky Texture* is then:

$$\begin{aligned} IG &= H(P_t) - \sum_{n \in c(t)} p(n) H(P_n) \\ &= 1 - \left[\frac{6}{10} \times \frac{14}{15} + \frac{4}{10} \times \frac{4}{5} \right] \\ &= 1 - \left[\frac{84}{150} + \frac{16}{50} \right] \\ &= 1 - \left[\frac{28}{50} + \frac{16}{50} \right] \\ &= 1 - \frac{44}{50} = \frac{3}{25} = 0.12 \end{aligned}$$

Here the corresponding entropy calculations referenced in Table 1:

$$\begin{aligned} H(P_{\text{CC-Brown}}) &= -\left[\frac{2}{3}\log_2\left(\frac{2}{3}\right) + \frac{1}{3}\log_2\left(\frac{1}{3}\right)\right] \\ &\approx -\left[(-0.6)\times\frac{2}{3} + (-1.6)\times\frac{1}{3}\right] \\ &= -\left[-\frac{3}{5}\times\frac{2}{3} - \frac{1}{3}\times\frac{1}{3}\right] \\ &= -\left[-\frac{6}{15} - \frac{8}{15}\right] \\ &= \frac{14}{15} \end{aligned}$$

$$\begin{aligned} H(P_{\text{CC-Orange}}) &= -\left[\frac{1}{2}\log_2\left(\frac{1}{2}\right) + \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right] \\ &= -\left[(-1)\times\frac{1}{2} + (-1)\times\frac{1}{2}\right] \\ &= 1 \end{aligned}$$

$$H(P_{\text{CC-Red}}) = -\left[\frac{1}{2}\log_2\left(\frac{1}{2}\right) + \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right] = H(P_{\text{CC-Orange}})$$

$$H(P_{\text{CC-Yellow}}) = -\left[\frac{1}{3}\log_2\left(\frac{1}{3}\right) + \frac{2}{3}\log_2\left(\frac{2}{3}\right)\right] = H(P_{\text{CC-Brown}})$$

Here the corresponding entropy calculations referenced in Table 2:

$$\begin{aligned} H(P_{\text{GC-White}}) &= -\left[\frac{4}{7}\log_2\left(\frac{4}{7}\right) + \frac{3}{7}\log_2\left(\frac{3}{7}\right)\right] \\ &\approx -\left[(-0.8)\times\frac{4}{7} + (-1.2)\times\frac{3}{7}\right] \\ &= -\left[-\frac{4}{5}\times\frac{4}{7} - \frac{3}{7}\times\frac{6}{5}\right] \\ &= \frac{16}{35} + \frac{18}{35} \\ &= \frac{34}{35} \end{aligned}$$

$$H(P_{\text{GC-Yellow}}) = -\left[\frac{1}{3}\log_2\left(\frac{1}{3}\right) + \frac{2}{3}\log_2\left(\frac{2}{3}\right)\right] = H(P_{\text{CC-Yellow}})$$

Here the corresponding entropy calculations referenced in Table 3:

$$\begin{aligned} H(P_{\text{ST-Yes}}) &= -\left[\frac{2}{6}\log_2\left(\frac{2}{6}\right) + \frac{4}{6}\log_2\left(\frac{4}{6}\right)\right] \\ &= -\left[\frac{1}{3}\log_2\left(\frac{1}{3}\right) + \frac{2}{3}\log_2\left(\frac{2}{3}\right)\right] = H(P_{\text{CC-Brown}}) \end{aligned}$$

$$\begin{aligned} H(P_{\text{ST-No}}) &= -\left[\frac{3}{4}\log_2\left(\frac{3}{4}\right) + \frac{1}{4}\log_2\left(\frac{1}{4}\right)\right] \\ &\approx -\left[(-0.4)\times\frac{3}{4} + (-2.0)\times\frac{3}{4}\right] \\ &= -\left[-\frac{2}{5}\times\frac{3}{4} - \frac{10}{5}\times\frac{3}{4}\right] \\ &= \frac{6}{20} + \frac{10}{20} = \frac{16}{20} = \frac{4}{5} \end{aligned}$$

2. Question Plot the resulting decision stump after the execution of the optimal root-node split. The stump should clearly indicate the classification decision associated with each leaf node.

Answer The following tree shows the resulting decisions tree after a root-node split and treating the resulting children as leafs, where the classification is based on majority voting of the present class.

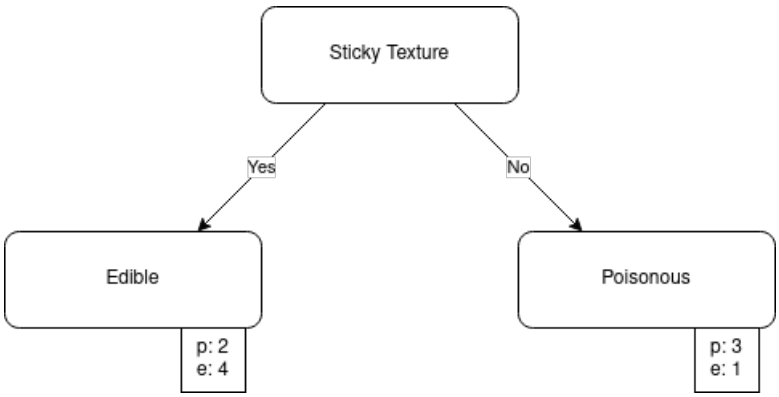

Figure 1: Decision Tree after choosing the maximizing *Information Gain* attribute *Sticky Texture*

3. Question Provide the confusion matrix of your decision stump on the training data. Compute precision, recall and F1-score for class "Yes".

Answer Confusion matrix:

		ground truth Poisonous (Yes)	ground truth Edible (No)	$\sum_{\text{row}}$
classification	Poisonous (Yes)	3	1	4
	Edible (No)	2	4	6
$\sum_{\text{col}}$		5	5	10

Resulting scores (short notation taken from Slide Deck 2):

$$p(\text{"Yes"}) = \frac{3}{4}, \quad r(\text{"Yes"}) = \frac{3}{5}, \quad F_1(\text{"Yes"}) = \frac{2 \times \frac{3}{5} \times \frac{3}{4}}{\frac{3}{5} + \frac{3}{4}} = \frac{2}{3}$$

2. Decision Trees and Information Gain

1. Question What is the information gain of the split made at the root node?

Answer First, we calculate the entropy for the different nodes (T = root node, L = left node, R = right node):

$$\begin{aligned}
H(P_T) &= -\sum_{x \in X} p(x) \log_2(p(x)) \\
&= -\left[\frac{1}{2} + \frac{1}{2}\right] = 1 \\
H(P_L) &= -\left[\frac{13}{18} \log_2\left(\frac{13}{18}\right) + \frac{5}{18} \log_2\left(\frac{5}{18}\right)\right] \approx 0.852 \\
H(P_R) &= -\left[\frac{5}{8} \log_2\left(\frac{5}{8}\right) + \frac{3}{18} \log_2\left(\frac{3}{18}\right)\right] \approx 0.852
\end{aligned}$$

And the resulting information gain is then:

$$\begin{aligned}
IG &= H(P_t) - \sum_{n \in c(t)} p(n) H(P_n) \\
&= 1 - \left[\frac{18}{36} H(P_L) + \frac{18}{36} H(P_R)\right] \\
&= 1 - \left[\frac{18}{36} H(P_L) + \frac{18}{36} H(P_L)\right] \\
&= 1 - H(P_L) = 0.148
\end{aligned}$$

2. Question Which values of a and b will result in the minimal information gain at this split, and what is the corresponding value?

Answer As entropy is maximized if a distribution is evenly spread, i.e. uniformly distributed events, the information gain will be minimized. Assume for example $a = b$ and $n = a + b$:

$$\begin{aligned}
H(P_{L'}) &= -\left[\frac{a}{n} \log_2\left(\frac{a}{n}\right) + \frac{b}{n} \log_2\left(\frac{b}{n}\right)\right] \\
&= -\left[\frac{a}{2a} \log_2\left(\frac{a}{2a}\right) + \frac{b}{2b} \log_2\left(\frac{b}{2b}\right)\right] \\
&= -\left[\frac{1}{2} + \frac{1}{2}\right] = 1
\end{aligned}$$

Also assume that we have an even distribution in the right node, then:

$$\begin{aligned} IG &= H(P_{T'}) - \sum_{n \in c(t)} p(n) H(P_n) \\ &= H(P_{T'}) - \left[\frac{18-n}{18} \times H(P_{L'}) + \frac{n}{18} \times H(P_{R'}) \right] \\ &= H(P_{T'}) - \left[\frac{18-n}{18} \times 1 + \frac{n}{18} \times 1 \right] \\ &= H(P_{T'}) - 1 = 0.852 - 1 = -0.148 \end{aligned}$$

We would even lose information. Unfortunately, in this example we are unable to uniformly distribute the classes over the nodes, hence for this the closest possible solution to minimizing information gain, is to uniformly as possible distribute the classes over both child nodes, i.e. $a = 7$ and $b = 3$ (and in the other node $6, 2$):

$$\begin{aligned} \rightarrow H(P_{L'}) &= \left[\frac{7}{10} \log_2 \left(\frac{7}{10} \right) + \frac{3}{10} \log_2 \left(\frac{3}{10} \right) \right] \approx 0.881 \\ H(P_{R'}) &= \left[\frac{6}{8} \log_2 \left(\frac{6}{8} \right) + \frac{2}{8} \log_2 \left(\frac{2}{8} \right) \right] \approx 0.811 \end{aligned}$$

$$\begin{aligned} \Rightarrow IG &= H(P_{T'}) - \left[\frac{10}{18} \times H(P_{L'}) + \frac{8}{18} \times H(P_{R'}) \right] \\ &= 0.852 - \left[\frac{10}{18} \times 0.881 + \frac{8}{18} \times 0.811 \right] \\ &\approx 0.852 - 0.852 = 0.002 \end{aligned}$$

$\Rightarrow$ The information gain is almost 0, i.e. almost no information was introduced by the split for the classification.

3. Question Which values of c and d will yield the maximum information gain at this split, and what is the resulting value?

Answer We have the inverse case to question 2.2, where we minimize the entropy to maximize the gain. This happens, when a distribution is skewed towards exactly one value. Assume in our case $c = 5$ and $d = 0$ (analogously $c = 0$ and $d = 18$):

$$\begin{aligned} H(P_{L''}) &= \left[\frac{5}{5} \log_2 \left(\frac{5}{5} \right) + \frac{0}{5} \log_2 \left(\frac{0}{5} \right) \right] = 0 \\ H(P_{R''}) &= 0 \\ IG &= 0.852 - \left[\frac{5}{18} \times 0 + \frac{13}{18} \times 0 \right] = 0.852 \end{aligned}$$

Hence, maximal information gain due to perfect classification.

3. Regression with Decision Trees and kNN

Note, we used the book "The Elements of Statistical Learning: Data Mining, Inference, and Prediction." by Trevor Hastie, Robert Tibshirani, Jerome Friedman as resource for these questions.

1. Question What are the distinctions between constructing regression trees and classification trees? How is a prediction calculated in a regression tree?

Answer Construction for regression trees, as its name suggest, is done in a similar way as linear regression, by fitting a constant function to minimize the residual. To keep the explanation short, we only restrict the explanation to the construction (e.g. not going to explain why a simple sum of squares is infeasible to use for construction). On a highly abstractional level, imagine a 2-D plane, on which the classification of a point is defined by its position. Regression trees recursively cut this plane in two halves, such that the residual in each half is minimized . This cutting in half is closely related to fitting a constant function to the points, i.e like regression. Prediction for a new point on this plane is fairly intuitive, i.e. by traversing its cutting criterion's till a leaf node is reached and using its fitted constant. Let's take a closer look on how this cut is found (Hastie et al, p 307):

The halves are defined by:

$$R_1(j,s) = \{X|X_j \leq s\} \quad \text{and} \quad R_2(j,s) = \{X|X_j \geq s\}$$

These two classes split the plane cleanly in two, where each predictor variable where its j-th value is smaller or equal than s is on the R_1 half and subsequently all predictor variables with j-th values greater than s are on the R_2 half. Note, that the halves are not restricted to be equal in size!

$$\min_{j,s} \Big[\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2 \Big]$$

Similar to linear regression, the inner term fits a constant function to its corresponding half (i.e. c_1 to R_1 and c_2 to R_2), such that the sum of squared residuals is minimized in each half. Note, that the actual parameters fitted (outer term) are the s and j values, i.e. the values responsible on how a plane is cut. Also note, that due to recursion, only predictor variables in the cut plane are examined and the fitted function is always the average of all y_i in the plane).

A classification tree works on the exact same basis, only with a slightly modified inner term for optimization of parameters s and j. Instead of using the residual, the miss-classification in each region has to be minimized. Hence, classification is similar to prediction on regression trees, by traversing.

2. Question How can k-nearest-neighbors (kNN) be utilized for regression purposes?

Answer (Hastie et al, p 14 and following) The k Nearest Neighbor approach for regression, i.e. where the predictor and response variable(s) is(are) continuous, does not change significantly. For any predictor variable(s) value(s) it chooses the k nearest neighbour of its (discrete) training data to calculate (based on some metric, like average with or without some weighting of distances) the continuous predictor variable. Hence, it is possible to calculate the continuous value from these discrete values.