Kyle Salitrik

kps168

CMPSC448 - HW3

1) Initial Entropy: $E_I = -\frac{9}{16}\log_2\left(\frac{9}{16}\right) - \frac{7}{16}\log_2\left(\frac{7}{16}\right) \approxeq 0.989$

$$E(\text{color, edible}) = \frac{3}{16}\left(-\frac{1}{3}\log_2\left(\frac{1}{3}\right) - \frac{2}{3}\log_2\left(\frac{2}{3}\right)\right) + \frac{13}{16}\left(-\frac{8}{13}\log_2\left(\frac{8}{13}\right) - \frac{5}{13}\log_2\left(\frac{5}{13}\right)\right) \cong 0.953 \qquad \text{Gain} \cong 0.0355$$

$$E(\text{size, edible}) = \frac{8}{16}\left(-\frac{3}{8}\log_2\left(\frac{3}{8}\right) - \frac{5}{8}\log_2\left(\frac{5}{8}\right)\right) + \frac{8}{16}\left(-\frac{6}{8}\log_2\left(\frac{6}{8}\right) - \frac{2}{8}\log_2\left(\frac{2}{8}\right)\right) \cong 0.882 \qquad \text{Gain} \cong 0.1058$$

$$E(\text{shape, edible}) = \frac{12}{16}\left(-\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right)\right) + \frac{4}{16}\left(-\frac{3}{4}\log_2\left(\frac{3}{4}\right) - \frac{1}{4}\log_2\left(\frac{1}{4}\right)\right) \cong 0.953 \qquad \text{Gain} \cong 0.0358$$

Based on the information gain, size should be the root decision node

2) a) Given that the numerical features are treated as discrete classes, encountering an unseen example (or new) class during prediction may cause errors such as being unable to classify the sample. It would also be impossible to create a general solution because there is an infinite set of classes (numbers).

b) Assuming the height of the tree is: $1 \leq h \leq d$
Given d features and n data examples, the runtime should be $O(n \cdot (m-1)\log_2(h))$ for the worst case where each data example reaches the bottom of the tree and requires comparison w/ the last partition.

3) a) $h(\text{yellow}) = \begin{cases} 1 & \text{if } x = 1 \\ -1 & \text{otherwise} \end{cases}$

$$D_1 = \left\{\frac{1}{16}, \frac{1}{16}, \ldots, \frac{1}{16}\right\}_{1\times 16}$$

$$\varepsilon_1 = \frac{1}{16}\left\{(h(x_1) \neq y_1) + (h(x_2) \neq y_2) + \cdots + (h(x_{16}) \neq y_{16})\right\} = \frac{1}{16}\left\{\overset{(1)}{0} + \overset{(2)}{1} + \overset{(3)}{1} + \overset{(4)}{0} + \overset{(5)}{0} + \overset{(6)}{0} + \overset{(7)}{0} + \overset{(8)}{0} + 0^{(9)} + 1^{(10)} + 0^{(11)} + 1^{(12)} + 1^{(13)} + 1^{(14)} + \overset{(15)}{0} + 0^{(16)}\right. = \frac{1}{16}(6)$$

$$\alpha_1 = \frac{1}{2}\ln\left(\frac{1-0.375}{0.375}\right) \approxeq \boxed{0.255}$$

b) $Z_1 = \frac{1}{2}\sqrt{\left(\frac{6}{16}\left(\frac{10}{16}\right)\right)} \approxeq 0.242$

$\frac{D_1}{Z_1} = 1/\sqrt{15} \approx 0.258$

b)
$$D_2 = \frac{1}{\sqrt{15}} \left\{ e^{-\alpha_1 y_1 h_1(x_1)}, e^{-\alpha_2 y_2 h_1(x_2)} \dots, e^{-\alpha_1 y_{16} h(x_{16})} \right\}$$
$$= \left\{ \frac{1}{5}, \frac{1}{3}, \frac{1}{3}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{3}, \frac{1}{5}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{5}, \frac{1}{5} \right\}$$

c)
Ideally you would stop before the error reaches 0. This is because you want to avoid over-fitting.
According to the introductory chapter, we would stop at $0 < \epsilon_t < \frac{1}{2} - \gamma$ where $\gamma > 0$ is a chosen value.

4) For this section, I chose SVM and Random Forest Classifiers to train.

a)
Random Forest Classifiers work by training multiple decision trees on the data and returns the class with the most votes from the forest as the prediction.
The SVM algorithm attempts to fit a decision boundary to the data while maximizing the margin between the data and decision boundary. To do this, the algorithm uses the closest points to the boundary as "support vectors" and attempts to maximize the distance from these vectors to the decision boundary. For non-linearly separable data, a kernel is used to map the data to higher dimensions.

b)
To train each model, I started with a baseline of their default hyperparameters. For SVM, the first parameter I changed was the kernel function to see which performed best.
Next, I tuned the C and gamma parameters.
For the random forest Classifier, I began by varying the depth of the trees followed by the number of trees in the forest. After finding the best performance with these values, I varied the other hyperparameters.

c)
Hyperparameters for SVM:
- Kernel_type - the kernel defines which function should be used to map the data into higher dimensions. The default is RBF, however a linear kernel fit best.
- C - the C value changes the weight of the penalty for an incorrect classification. For this dataset, the default of C=1.0 worked best w/ a linear classifier
- Gamma - the gamma parameter tunes how much influence a point has on those around it. Gamma allows one to trade off bias and variance of the model. I kept gamma at the default of $\frac{1}{\text{n\_features}}$. The parameter had very little impact on the linear kernel.

Hyperparameters for RFC:
- n_estimators: Simply the number of trees in the random forest. Default is 10, the best performance came from 10 to 15 depending on the max depth. For the chosen classifier, the default of 10 worked best.

- max_depth: Determines how tall each tree is. Default is 2, but a max depth of 15 provided the best results.
- min_impurity_decrease: Default is zero. This parameter helps to determine when a node should be split. This value was set to 0.00001.
- min_samples_leaf: Default = 1. This prevents a leaf from being made if the number of training samples is lower than the threshold. This parameter was set to 2.
- Bootstrap: Determines whether or not data samples are drawn with replacement. Default of true was kept.

d & e) See attached files. Note: not every combination was kept for comparison.