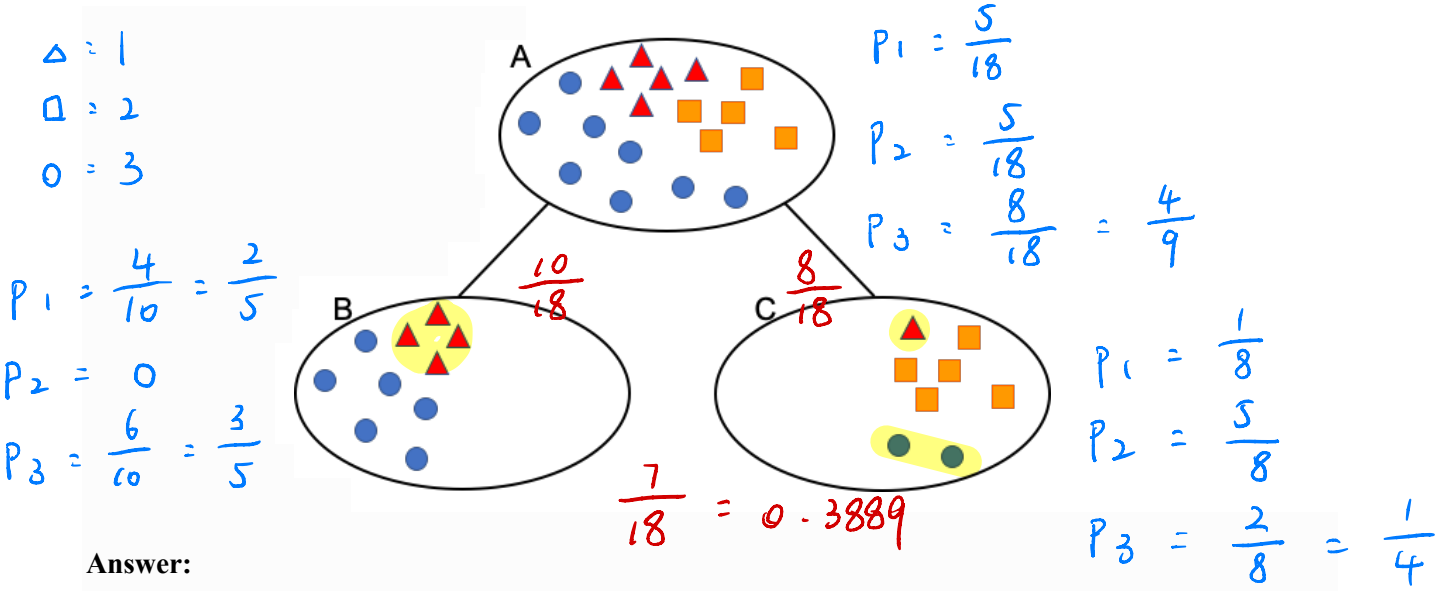
EE2211 Tutorial 9

(Gini impurity, entropy and misclassification rate)

Question 1:

Compute the Gini impurity, entropy, misclassification rate for nodes A, B and C, as well as the overall metrics (Gini impurity, entropy misclassification error) at depth 1 of the decision tree shown below.



Answer:

Let's assume class 1, class 2 and class 3 correspond to red triangles, orange squares and blue circles respectively.

- For node A, $p_1 = \frac{5}{18}, p_2 = \frac{5}{18}, p_3 = \frac{8}{18} = \frac{4}{9}$
- For node B, $p_1 = \frac{4}{10} = \frac{2}{5}, p_2 = \frac{0}{10} = 0, p_3 = \frac{6}{10} = \frac{3}{5}$
- For node C, $p_1 = \frac{1}{8}, p_2 = \frac{5}{8}, p_3 = \frac{2}{8} = \frac{1}{4}$

For **Gini impurity**, recall formula is $1 - \Sigma_{i=1}^{K} p_i^2$

- Node A: $1 - \left(\frac{5}{18}\right)^2 - \left(\frac{5}{18}\right)^2 - \left(\frac{4}{9}\right)^2 = 0.6481$
- Node B: $1 - \left(\frac{2}{5}\right)^2 - (0)^2 - \left(\frac{3}{5}\right)^2 = 0.48$
- Node C: $1 - \left(\frac{1}{8}\right)^2 - \left(\frac{5}{8}\right)^2 - \left(\frac{1}{4}\right)^2 = 0.5312$
- Overall Gini at depth 1: $\left(\frac{10}{18}\right) 0.48 + \left(\frac{8}{18}\right) 0.5312 = 0.5028$

Observe the decrease in Gini impurity from root (0.6481) to depth 1 (0.5028)

For **entropy**, recall formula is $-\sum_i p_i \log_2 p_i$

- Node A: $-\left(\frac{5}{18}\right)\log_2\left(\frac{5}{18}\right) - \left(\frac{5}{18}\right)\log_2\left(\frac{5}{18}\right) - \left(\frac{4}{9}\right)\log_2\left(\frac{4}{9}\right) = 1.5466$
- Node B: $-\left(\frac{2}{5}\right)\log_2\left(\frac{2}{5}\right) - (0)\log_2(0) - \left(\frac{3}{5}\right)\log_2\left(\frac{3}{5}\right) = 0.9710$
- Node C: $-\left(\frac{1}{8}\right)\log_2\left(\frac{1}{8}\right) - \left(\frac{5}{8}\right)\log_2\left(\frac{5}{8}\right) - \left(\frac{1}{4}\right)\log_2\left(\frac{1}{4}\right) = 1.2988$
- Overall entropy at depth 1: $\left(\frac{10}{18}\right) 0.9710 + \left(\frac{8}{18}\right) 1.2988 = 1.1167$

Observe the decrease in entropy from root (1.5466) to depth 1 (1.1167)

For **misclassification rate**, recall formula is $1 - \max_i p_i$

- Node A: $1 - \max\left(\left(\frac{5}{18}\right), \left(\frac{5}{18}\right), \left(\frac{4}{9}\right)\right) = 1 - \left(\frac{4}{9}\right) = \frac{5}{9} = 0.5556$
- Node B: $1 - \max\left(\left(\frac{2}{5}\right), 0, \left(\frac{3}{5}\right)\right) = 1 - \left(\frac{3}{5}\right) = \frac{2}{5}$
- Node C: $1 - \max\left(\left(\frac{1}{8}\right), \left(\frac{5}{8}\right), \left(\frac{1}{4}\right)\right) = 1 - \left(\frac{5}{8}\right) = \frac{3}{8}$
- Overall misclassification error rate at depth 1: $\left(\frac{10}{18}\right)\left(\frac{2}{5}\right) + \left(\frac{8}{18}\right)\left(\frac{3}{8}\right) = 0.3889$
- We can also double check that at depth 1, the 4 red triangles will be classified wrongly for node B and the 1 red triangle + 2 blue circles will be classified wrongly for node C. So in total, there will be 7 wrong classifications out of 18 datapoints, which corresponds to $\left(\frac{7}{18}\right) = 0.3889$
- Observe the decrease in misclassification rate from root (0.5556) to depth 1 (0.3889)

1. $x \leq 5$

2. $x > 5$

(MSE of regression trees)

Question 2:

Calculate the overall MSE for the following data at depth 1 of a regression tree assuming a decision threshold is taken at $x = 5.0$. How does it compare with the MSE at the root?

$\{x, y\}$: {1, 2}, {0.8, 3}, {2, 2.5}, {2.5, 1}, {3, 2.3}, {4, 2.8}, {4.2, 1.5}, {6, 2.6}, {6.3, 3.5}, {7, 4}, {8, 3.5}, {8.2, 5}, {9, 4.5}

Answer:

At depth 1, when $x > 5$

- $y = \{2.6, 3.5, 4, 3.5, 5, 4.5\} => \bar{y} = 3.85$
- MSE = $\frac{1}{6}((2.6 - \bar{y})^2 + (3.5 - \bar{y})^2 + (4 - \bar{y})^2 + (3.5 - \bar{y})^2 + (5 - \bar{y})^2 + (4.5 - \bar{y})^2) = 0.5958$

At depth 1, when $x \leq 5$

- $y = \{2, 3, 2.5, 1, 2.3, 2.8, 1.5\} => \bar{y} = 2.1571$
- MSE = $\frac{1}{7}((2 - \bar{y})^2 + (3 - \bar{y})^2 + (2.5 - \bar{y})^2 + (1 - \bar{y})^2 + (2.3 - \bar{y})^2 + (2.8 - \bar{y})^2 + (1.5 - \bar{y})^2) = 0.4367$

Overall MSE at depth 1: $\frac{6}{13} \times 0.5958 + \frac{7}{13} \times 0.4367 = 0.5102$

At the root:

- $y = \{2, 3, 2.5, 1, 2.3, 2.8, 1.5, 2.6, 3.5, 4, 3.5, 5, 4.5\} => \bar{y} = 2.9385$
- MSE = $\frac{1}{13}((2.6 - \bar{y})^2 + (3.5 - \bar{y})^2 + (4 - \bar{y})^2 + (3.5 - \bar{y})^2 + (5 - \bar{y})^2 + (4.5 - \bar{y})^2 + (2 - \bar{y})^2 + (3 - \bar{y})^2 + (2.5 - \bar{y})^2 + (1 - \bar{y})^2 + (2.3 - \bar{y})^2 + (2.8 - \bar{y})^2 + (1.5 - \bar{y})^2) = 1.2224$

Therefore, MSE has decreased from 1.2224 at the root to 0.5102 at depth 1

(Regression tree, Python)

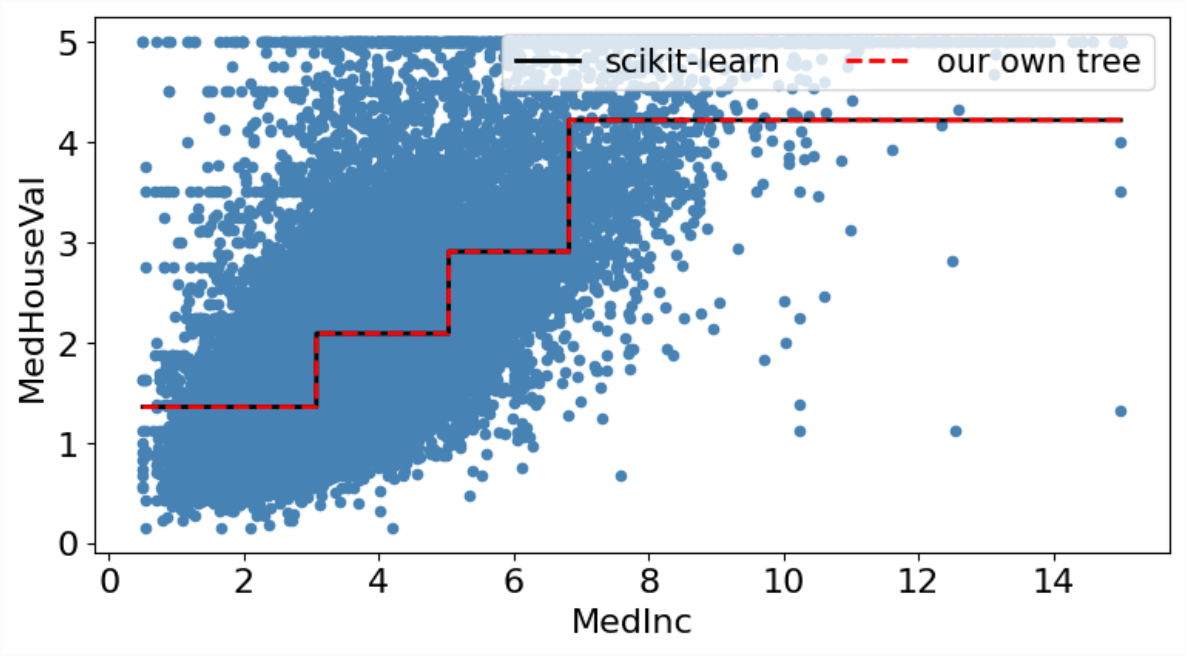
Question 3:

Import the California Housing dataset "`from sklearn.datasets import fetch_california_housing`" and "`housing = fetch_california_housing()`". This data set contains 8 features and 1 target variable listed below. Use "`MedInc`" as the input feature and "`MedHouseVal`" as the target output. Fit a regression tree to depth 2 and compare your results with results generated by "`from sklearn.tree import DecisionTreeRegressor`" using the "squared error" criterion.

```
Target: ['MedHouseVal']
Features:['MedInc', 'HouseAge', 'AveRooms', 'AveBedrms', 'Population',
'AveOccup', 'Latitude', 'Longitude']
```

Answer:

Please refer to Tut9_Q3_zhou.py. We can exactly replicate the results from scikit-learn. Note that in the plot below, the blue dots are the training datapoints. The curves from scikit-learn (black line) and our own tree (red dashed line) are on top of each other, so they might be hard to tell apart.



(Classification tree, Python)

Question 4:

Get the data set "`from sklearn.datasets import load_iris`". Perform the following tasks.

(a) Split the database into two sets: 80% of samples for training, and 20% of samples for testing using `random_state=0`

(b) Train a decision tree classifier (i.e., "`tree.DecisionTreeClassifier`" from sklearn) using the training set with a maximum depth of 4 based on the "entropy" criterion.

(c) Compute the training and test accuracies. You can use `accuracy_score` from `sklearn.metrics` for accuracy computation
(d) Plot the tree using "`tree.plot_tree`".

Answer:

Please refer to Tut9_Q4_yeo.py.

Training accuracy: 0.9917
Test accuracy: 1.0

The resulting tree looks like this: