



**Faculty of Engineering**  
Cairo University

# **Assignment 3**

**Mohamed Sameh Mohamed Zaki**

**1170556**

## Manually Built Tree:

### Assignment-3 Decision Tree

Samples = 14  
A = 8  
not A = 6

$$\text{Entropy of dataset} = - \left[ \frac{8}{14} \log_2 \left( \frac{8}{14} \right) + \frac{6}{14} \log_2 \left( \frac{6}{14} \right) \right] = 0.985$$

Early reg

$$\begin{aligned} \text{yes} &\rightarrow - \frac{4}{6} \log_2 \frac{4}{6} - \frac{2}{6} \log_2 \frac{2}{6} = 0.92 \\ \text{no} &\rightarrow - \frac{4}{8} \log_2 \frac{4}{8} - \frac{4}{8} \log_2 \frac{4}{8} = 1 \end{aligned}$$

$$IG(\text{Early reg}) = 0.985 - \frac{6}{14} \times 0.92 - \frac{8}{14} \times 1 = 0.019$$

Finished HW

$$\begin{aligned} \text{yes} &\rightarrow - \frac{5}{7} \log_2 \frac{5}{7} - \frac{2}{7} \log_2 \frac{2}{7} = 0.86 \\ \text{no} &\rightarrow - \frac{4}{7} \log_2 \frac{4}{7} - \frac{3}{7} \log_2 \frac{3}{7} = 0.985 \end{aligned}$$

$$IG(\text{Finished HW}) = 0.985 - \frac{7}{14} \times 0.86 - \frac{7}{14} \times 0.985 = 0.0625$$

Senior

$$\begin{aligned} \text{yes} &\rightarrow - \frac{5}{8} \log_2 \frac{5}{8} - \frac{3}{8} \log_2 \frac{3}{8} = 0.954 \\ \text{no} &\rightarrow - \frac{3}{6} \log_2 \frac{3}{6} - \frac{3}{6} \log_2 \frac{3}{6} = 1 \end{aligned}$$

$$IG(\text{Senior}) = 0.985 - \frac{8}{14} \times 0.954 - \frac{6}{14} \times 1 = 0.0113$$

Likes coffee

$$\begin{aligned} \text{yes} &\rightarrow - \frac{3}{4} \log_2 \frac{3}{4} - \frac{1}{4} \log_2 \frac{1}{4} = 0.811 \\ \text{no} &\rightarrow - \frac{5}{10} \log_2 \frac{5}{10} - \frac{5}{10} \log_2 \frac{5}{10} = 1 \end{aligned}$$

$$IG(\text{coffee}) = 0.985 - \frac{4}{14} \times 0.811 - \frac{10}{14} \times 1 = 0.039$$

last HW

$$\begin{aligned} \text{yes} &\rightarrow - \frac{5}{9} \log_2 \frac{5}{9} - \frac{4}{9} \log_2 \frac{4}{9} = 0.99 \\ \text{no} &\rightarrow - \frac{2}{5} \log_2 \frac{2}{5} - \frac{3}{5} \log_2 \frac{3}{5} = 0.778 \end{aligned}$$

$$IG(\text{last HW}) = 0.985 - \frac{9}{14} \times 0.99 - \frac{5}{14} \times 0.778 = 0.0657$$

• largest IG  $\rightarrow$  Liked last HW = 0.0657

Samples = 9  
A = 5  
not A = 4

Liked last HW

Samples = 5  
A = 3  
not A = 2

$$E(\text{data}) = - \left( \frac{5}{9} \log_2 \frac{5}{9} + \frac{4}{9} \log_2 \frac{4}{9} \right) = 0.99$$

$$E(\text{data}) = - \left( \frac{3}{5} \log_2 \frac{3}{5} + \frac{2}{5} \log_2 \frac{2}{5} \right) = 0.778$$

Early veg	Finished HW	Senior	Likes coffee	A
1	1	0	0	1
1	1	1	0	1
0	1	1	0	0
0	0	1	1	1
1	0	0	0	0
0	1	0	1	1
0	0	1	0	1
0	1	1	1	0
0	0	0	0	0

$$E(\text{data}) = 0.99$$

$$\text{early veg} \begin{cases} \text{yes} \rightarrow -\frac{2}{3} \log \frac{1}{3} - \frac{1}{3} \log \frac{1}{3} = 0.912 \\ \text{no} \rightarrow -\frac{3}{6} \log \frac{3}{6} - \frac{3}{6} \log \frac{3}{6} = 1 \end{cases}$$

$$IG = 0.99 \cdot \frac{2}{9} (0.912) - \frac{6}{9} (1) = 0.017$$

$$\text{Finished HW} \begin{cases} \text{yes} \rightarrow -\frac{4}{5} \log \frac{4}{5} - \frac{1}{5} \log \frac{1}{5} = 0.72 \\ \text{no} \rightarrow -\frac{1}{4} \log \frac{1}{4} - \frac{3}{4} \log \frac{3}{4} = 1 \end{cases}$$

$$IG = 0.99 \cdot \frac{5}{9} (0.72) - \frac{4}{9} = 0.15$$

$$\text{Senior} \begin{cases} \text{yes} \rightarrow -\frac{3}{5} \log \frac{3}{5} - \frac{2}{5} \log \frac{2}{5} = 0.97 \\ \text{no} \rightarrow -\frac{2}{4} \log \frac{2}{4} - \frac{2}{4} \log \frac{2}{4} = 1 \end{cases}$$

$$IG = 0.99 \cdot \frac{5}{9} (0.97) - \frac{4}{9} = 0.0061$$

$$\text{Likes coffee} \begin{cases} \text{yes} \rightarrow -\frac{2}{3} \log \frac{2}{3} - \frac{1}{3} \log \frac{1}{3} = 0.918 \\ \text{no} \rightarrow -\frac{3}{6} \log \frac{3}{6} - \frac{3}{6} \log \frac{3}{6} = 1 \end{cases}$$

$$IG = 0.99 \cdot \frac{3}{9} (0.918) - \frac{6}{9} = 0.017$$

→ Finished HW has the highest IG

early veg	Finished HW	Senior	Likes coffee	A
0	0	1	0	0
0	1	1	0	1
1	0	0	0	0
1	1	1	0	1
1	0	0	1	1

$$E(\text{data}) = 0.778$$

$$\text{early veg} \begin{cases} \text{yes} \rightarrow -\frac{2}{3} \log \frac{2}{3} - \frac{1}{3} \log \frac{1}{3} = 0.918 \\ \text{no} \rightarrow -\frac{1}{2} \log \frac{1}{2} - \frac{1}{2} \log \frac{1}{2} = 1 \end{cases}$$

$$IG = 0.778 \cdot \frac{3}{5} (0.918) - \frac{2}{5} = -0.1728$$

$$\text{Finished HW} \begin{cases} \text{yes} \rightarrow -\frac{2}{2} \log \frac{2}{2} - \frac{0}{2} \log \frac{0}{2} = 0 \\ \text{no} \rightarrow -\frac{1}{2} \log \frac{1}{2} - \frac{1}{2} \log \frac{1}{2} = 1 \end{cases}$$

$$IG = 0.778 \cdot \frac{2}{5} (1) - \frac{3}{5} = 0.022$$

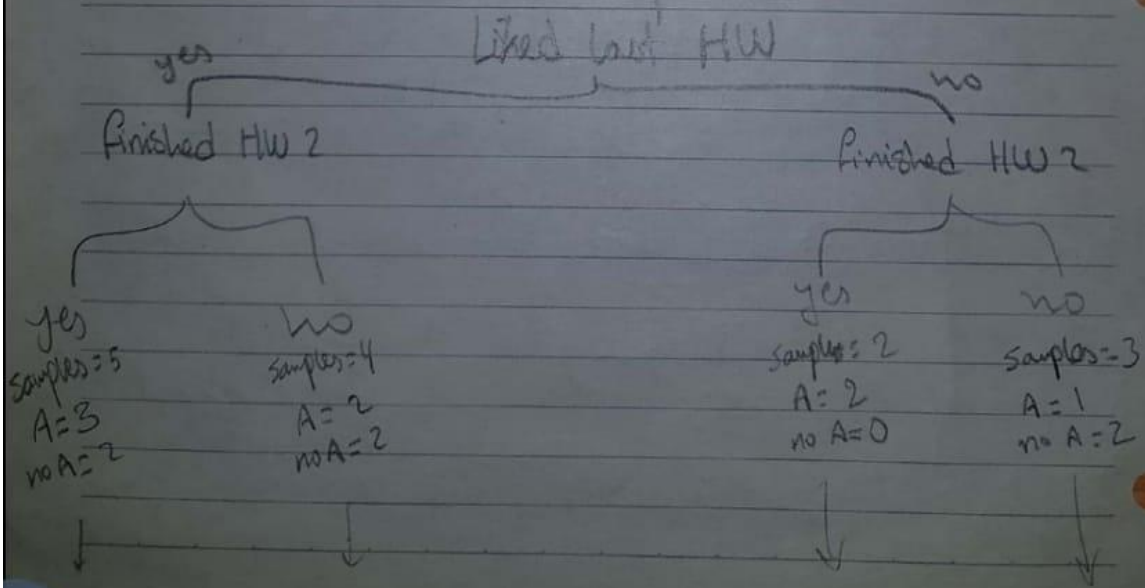
$$\text{Senior} \begin{cases} \text{yes} \rightarrow -\frac{2}{3} \log \frac{2}{3} - \frac{1}{3} \log \frac{1}{3} = 0.918 \\ \text{no} \rightarrow -\frac{1}{2} \log \frac{1}{2} - \frac{1}{2} \log \frac{1}{2} = 1 \end{cases}$$

$$IG = 0.778 \cdot \frac{3}{5} (0.918) - \frac{2}{5} = -0.1728$$

$$\text{Likes coffee} \begin{cases} \text{yes} \rightarrow -\frac{1}{1} \log 1 = 0 \\ \text{no} \rightarrow -\frac{2}{4} \log \frac{2}{4} - \frac{2}{4} \log \frac{2}{4} = 1 \end{cases}$$

$$IG = 0.778 \cdot \frac{4}{5} (1) - \frac{1}{5} = 0.022$$

→ Finished HW has the highest IG



## **Scikit-Learn Tree:**

- Split the dataset into X\_train, X\_test, y\_train, y\_test.
- Create a decision tree object from sklearn.tree.DecisionTreeClassifier.
- Use the decision tree object to fit the X\_train and y\_train.
- Use the test data to evaluate the model.
- To visualize the tree, set a max depth for the tree as the data is big and will cause problems while visualizing or will take a lot of time.

## **Decision Tree Built from Scratch:**

Created class DecisionTree which takes in the data, target column, and the value of the output we're looking for (positive).

The class contains 3 functions: getEntropy, getGain, and updateTree.

- getEntropy calculates the entropy using the data given by getting the negative of the sum of the positive ratio \* log2 the positive ratio and the negative ratio \* log2 the negative ratio.
- getGain calculates the information gain by subtracting the entropy of the data by the calculated mutual information.
- updateTree calls both functions and splits the tree according to the feature with the highest gain. Then, it recursively does the same for the children's trees.

To use:

Create object of the DecisionTree class given the needed info. and call the updateTree function.