Decision Trees (Udemy)

16 July 2024 14:25

Decision Tree Classifier

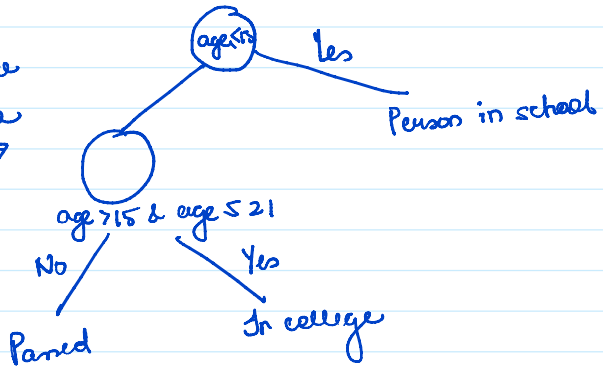
There are 2 types of algorithms to implement Decision Tree

- ID3 – We can have multiple splits at each level
- CART – We have 2 splits at each level

Example age = 14

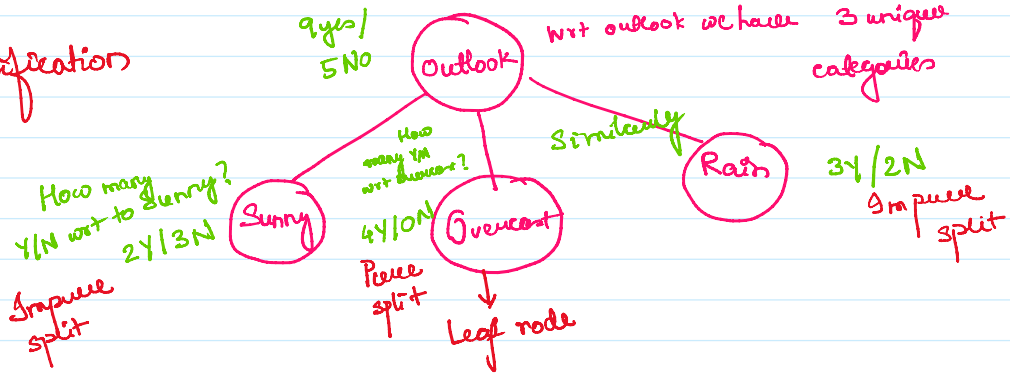
```
if (age <= 15):
    print('In school')
elif (age > 15 and age <= 21)
    print('Person maybe in college')
else:
    person has passed
```

Decision tree of this code



Dataset → Binary Classification

Day	Outlook	Temp	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



This splitting continues until we reach a leaf node.

With blunt eyes we can easily see impure and pure splits but how to determine mathematically

↓

Entropy & Information Gain

① Purity → Pure / Impure Split
- Entropy
- Gini Impurity

② What feature to select for splitting?

Feature that gives leaf nodes faster should be selected

↳ Information Gain used.

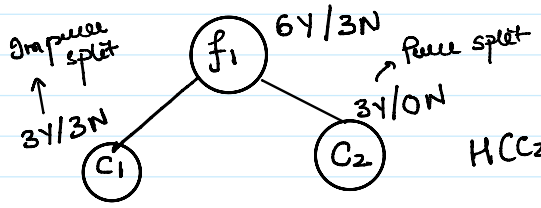
↳ Entropy
↳ Gini Impurity

feature ... selected
↳ Information Gain used.

① Entropy

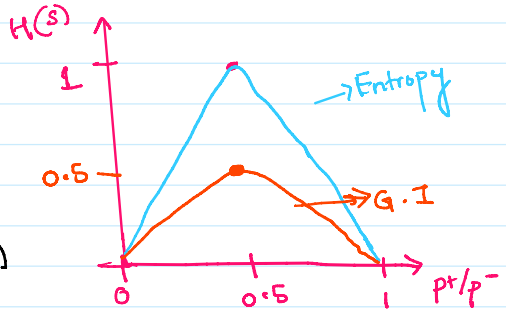
$$H(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

for binary → + → category 1
− → category 2



$$H(C_1) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$
$$= -\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6}$$
$$= -2\times\frac{3}{6}\log_2\frac{3}{6}$$
$$= 1 \Rightarrow \text{maximum impure value}$$

$$H(C_2) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$
$$= -\frac{3}{3}\log_2\frac{3}{3} - 0\log_2 0$$
$$= -\log_2 1 - 0$$
$$= 0 \Rightarrow \text{minimum impure value}$$



② Gini Impurity

$$G.I = 1 - \sum_{i=1}^{n} (p)^2$$

For C_1
$$= 1 - \left((p_+)^2 + (p_-)^2\right)$$
$$= 1 - \left(\left(\frac{3}{6}\right)^2 + \left(\frac{3}{6}\right)^2\right)$$
$$= 1 - \frac{1}{2} = \frac{1}{2} \rightarrow \text{max value for G.I}$$

For C_2 $= 1 - \left(\left(\frac{3}{3}\right)^2 + \left(\frac{0}{3}\right)^2\right)$
$$= 1 - \left((1)^2 - 0\right)$$
$$= 1 - 1 = 0$$
↳ min value

For knowing the split purity/impurity we use the Entropy as well as Gini Impurity. But to decide which feature to choose we use Information Gain.

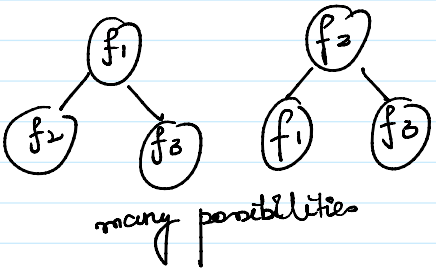
Now how to decide the feature to split with?

↳ We use Information gain

f_1 f_2 f_3 o/p

entropy of root node

f1 f2 f3 o/p

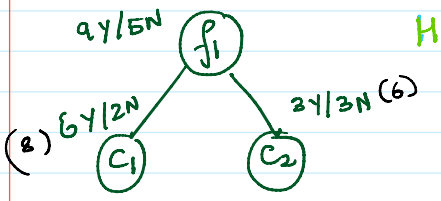


$$Gain(S, f_1) = H(S) - \sum_{v \in Val} \frac{|S_v|}{|S|} H(S_v)$$

entropy "Dil mein"

$$H(S) = -p_+ \log_2 p_+ \; - p_- \log_2 p_-$$

Let's take a scenario



$$H(S) = -\frac{9}{14}\log\frac{9}{14} - \frac{5}{14}\log\frac{5}{14}$$

≈ 0.94

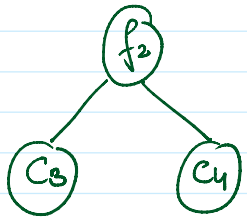
$$H(C_1) = -\frac{6}{8}\log_2\frac{6}{8} - \frac{2}{8}\log_2\frac{2}{8}$$

$= 0.81$

$H(C_2) = 1$ (very impure split)

$$Gain(S, f_1) = 0.94 - \left[\frac{8}{14} \times 0.81 + \frac{6}{14} \times 1\right]$$

$$\boxed{Gain(S, f_1) = 0.049}$$



$Gain(S, f_2) = 0.051$

$\therefore Gain(S, f_1) < Gain(S, f_2)$

$\therefore$ Split with feature f_2

Q) When to use entropy and when to use Gini Impurity?

Whenever dataset is small → Entropy → It involves a log term and hence it will take more computational time for larger datasets

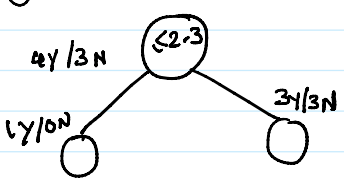
large → Gini Impurity

Decision Tree split for numerical feature

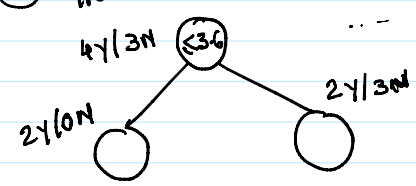
f1	o/p
2.3	Yes
3.6	Yes
4	No
5.2	No
6.7	Yes
8.9	No

① Sort the values first

① Threshold = 2.3



② Threshold = 3.6



This way keep creating multiple decision trees

(A) choose the split having the best information gain

6.7	Yes
8.9	No
10.5	Yes

We choose the split having the best information gain

- The disadvantage is it is very high time complexity as we compare a lot of decision trees

Post Pruning and Pre-Pruning

- When we try to train our decision tree to extreme depth, we tend to overfitting
- To reduce overfitting we have 2 methods, post & pre-pruning.

Post-pruning → 1) Construct the DT completely
2) Prune it with respect to depth
3) Used for smaller datasets → This because if dataset is large, making complete DT will be computationally very expensive

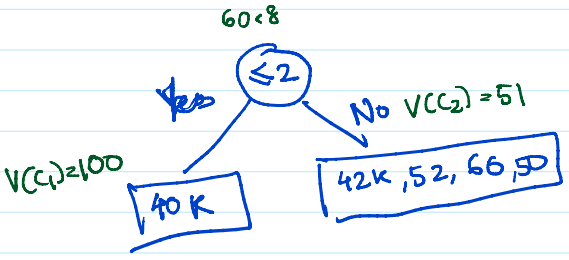
Pre-pruning → 1) Play with hyperparameters (Hyperparameter tuning while constructing DT)

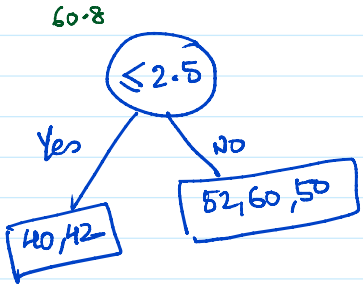
max-depth, max-features ... so on

Decision Tree Regressor

Dataset

Exp	Gap	Salary
2	Yes	40 K
2.5	Yes	42 K
3	No	52 K
4	No	60 K
4.5	Yes	50 K
		$\bar{y}=50$





How to select split? Variance Reduction

Variance Reduction

① Variance = $\frac{1}{n}\sum_{i=1}^{n}(y-\bar{y})^2$ → Average — Mean Squared Error

$$\text{Variance (Root)} = \frac{1}{5}\left[(40-50)^2 + (42-50)^2 + (62-50)^2 + (60-50)^2 + (50-50)^2\right]$$

$$= 60.8$$

For split 1

$$\text{Variance}(C_1) = \frac{1}{n}\sum_{i=1}^{n}(y-\bar{y})^2$$

$$= \frac{1}{1}(40-50)^2 = 100$$

$$\text{Variance}(C_2) = \frac{1}{n}\sum_{i=1}^{n}(y-\bar{y})^2$$

$$= \frac{1}{4}\left[(42-50)^2 + (62-50)^2 + (50-50)^2 + (60-50)^2\right]$$

$$= 61$$

$$\text{Variance reduction} = \text{Var(root)} - \sum w_i \, V(\text{child})$$

(w_i = ratio of elements on left to elements before split)

$$= 60.8 - \left[\frac{1}{5}\times 100 + \frac{4}{5}\times 61\right]$$

$$= 60.8 - 20 - 40.8$$

$$= 0$$

For split 2

$$\text{Variance}(C_1) = \frac{1}{2}\left[(40-50)^2 + (42-50)^2\right]$$

$$= 82$$

$$\text{Variance}(C_2) = \frac{1}{3}\left[(52-50)^2 + (50-50)^2 + (60-50)^2\right]$$

$$= \frac{140}{3} = 46.66$$

$$\text{Variance reduction} = \text{Var(root)} - \sum w_i \, \text{Var(child)}$$

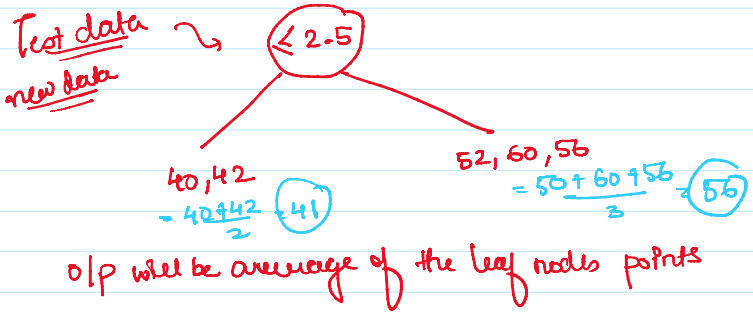
$$= 60.8 - \left[\frac{2}{5}\times 82 + \frac{3}{5}\times 4666\right]$$

$$= 60.8 - [32.8 + 27.96]$$

$$= 0.004$$

Variance Reduction (Left split) < Variance Reduction (Right split)

∴ We choose the right / 2nd split option



o/p will be average of the leaf nodes points