Lecture 16: Decision Trees

COMP90049

Semester 2, 2022

Joe West, CIS

Acknowledgement: Jeremy Nicholson, Tim Baldwin & Karin Verspoor

Roadmap

So far:

- Naive Bayes: Features are conditionally independent.
- Logistic Regression: Linear classifier
- Neural Network: Lack of interpretability
- KNN: time-consuming during testing

Today:

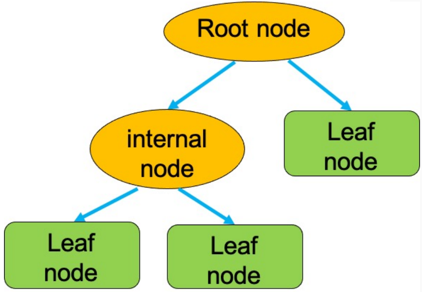
- Decision Trees
- ID3 Algorithm
  - Information Gain
  - Gain Ratio
- Properties of Decision Trees

THE UNIVERSITY OF MELBOURNE

Decision Trees

Tree-like graphical representation:

- Root/Internal Node: a test on an attribute
- Branch: outcome of the test
- Leaf: class

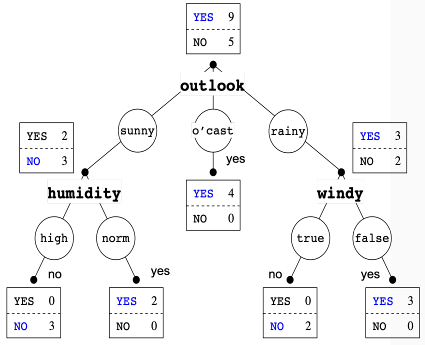


Example

	Outlook	Temperature	Humidity	Windy	Play
a	sunny	hot	high	FALSE	no
b	sunny	hot	high	TRUE	no
c	overcast	hot	high	FALSE	yes
d	rainy	mild	high	FALSE	yes
e	rainy	cool	normal	FALSE	yes
f	rainy	cool	normal	TRUE	no
g	overcast	cool	normal	TRUE	yes
h	sunny	mild	high	FALSE	no
i	sunny	cool	normal	FALSE	yes
j	rainy	mild	normal	FALSE	yes
k	sunny	mild	normal	TRUE	yes
l	overcast	mild	high	TRUE	yes
m	overcast	hot	normal	FALSE	yes
n	rainy	mild	high	TRUE	no

Example

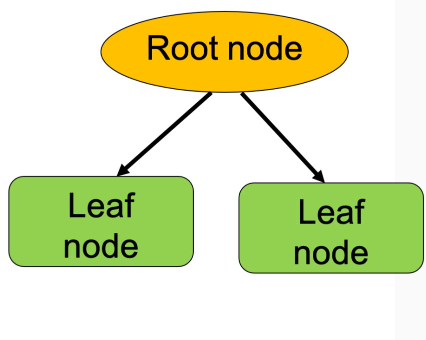
	Outlook	Temperature	Humidity	Windy	Play
a	sunny	hot	high	FALSE	no
b	sunny	hot	high	TRUE	no
c	overcast	hot	high	FALSE	yes
d	rainy	mild	high	FALSE	yes
e	rainy	cool	normal	FALSE	yes
f	rainy	cool	normal	TRUE	no
g	overcast	cool	normal	TRUE	yes
h	sunny	mild	high	FALSE	no
i	sunny	cool	normal	FALSE	yes
j	rainy	mild	normal	FALSE	yes
k	sunny	mild	normal	TRUE	yes
l	overcast	mild	high	TRUE	yes
m	overcast	hot	normal	FALSE	yes
n	rainy	mild	high	TRUE	no



Majority voting to assign class

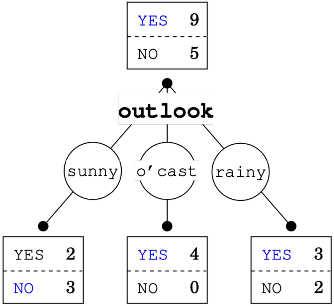
Decision Stump

One-level decision tree: leaves immediately connect to the root



Example

	Outlook	Play
a	sunny	no
b	sunny	no
c	overcast	yes
d	rainy	yes
e	rainy	yes
f	rainy	no
g	overcast	yes
h	sunny	no
i	sunny	yes
j	rainy	yes
k	sunny	yes
l	overcast	yes
m	overcast	yes
n	rainy	no

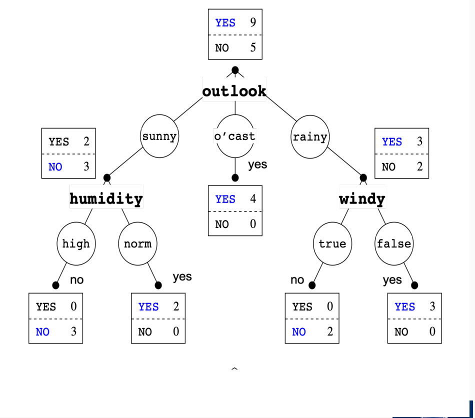


Traverse from root to leaf. Example:

- (sunny, hot, normal, false)
- (rainy, hot, low, false)

Missing values: try all attribute values, then majority voting.
Example:

- (?, cool, high, true)
- (?, hot, norm, false)



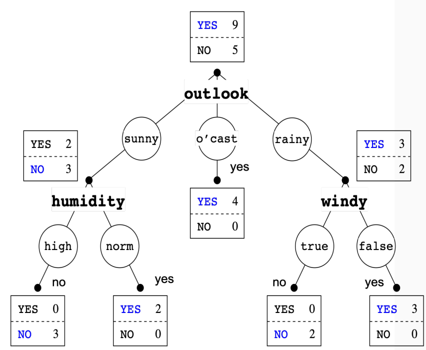
Applications

Examples:

- Business management: predict customer's intention to use e-commerce (Lee et al. 2007)
- Engineering: predict electricity energy consumption (Tso et al. 2007)
- Healthcare Management: predict breast cancer survivability. (Delen et al. 2005)

ID3 Algorithm

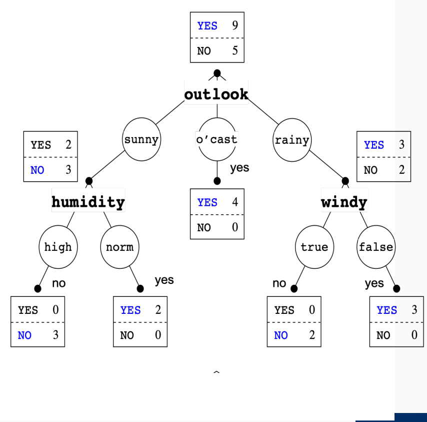
1. Select a new attribute.
2. Create a branch for each attribute value to partition the node instances.
3. Check if all instances at sub-nodes have same class or all attributes are run out:
   - if so, stop.
   - If not, go back to step 1 and repeat the process.



1. Select a new attribute.
2. Create a branch for each attribute value to partition the node instances.
3. Check if all instances at sub-nodes have same class or all attributes are run out:
   - if so, stop.
   - If not, go back to step 1 and repeat the process.

How to select new attribute?

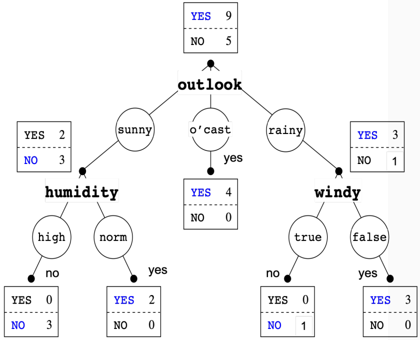
Choose attributes that can lead to a small tree



Smaller Trees vs Larger Trees

A smaller tree that fits the data generalizes better.

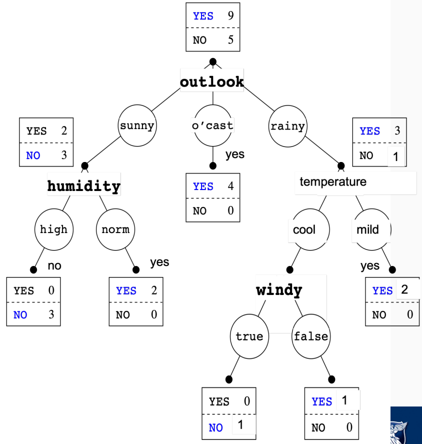
Train/ test	label	Outlook	Temperature	Humidity	Windy	Play
train	a	sunny	hot	high	FALSE	no
train	b	sunny	hot	high	TRUE	no
train	c	overcast	hot	high	FALSE	yes
train	d	rainy	mild	high	FALSE	yes
train	e	rainy	cool	normal	FALSE	yes
train	f	rainy	cool	normal	TRUE	no
train	g	overcast	cool	normal	TRUE	yes
train	h	sunny	mild	high	FALSE	no
train	i	sunny	cool	normal	FALSE	yes
train	j	rainy	mild	normal	FALSE	yes
train	k	sunny	mild	normal	TRUE	yes
train	l	overcast	mild	high	TRUE	yes
train	m	overcast	hot	normal	FALSE	yes
test	n	rainy	mild	high	TRUE	no



Smaller Trees vs Larger Trees

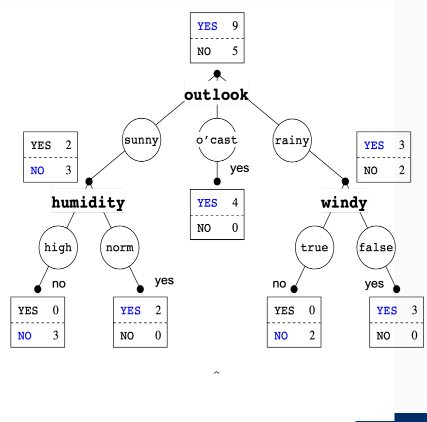
A smaller tree that fits the data generalizes better.

Train/ test	label	Outlook	Temperature	Humidity	Windy	Play
train	a	sunny	hot	high	FALSE	no
train	b	sunny	hot	high	TRUE	no
train	c	overcast	hot	high	FALSE	yes
train	d	rainy	mild	high	FALSE	yes
train	e	rainy	cool	normal	FALSE	yes
train	f	rainy	cool	normal	TRUE	no
train	g	overcast	cool	normal	TRUE	yes
train	h	sunny	mild	high	FALSE	no
train	i	sunny	cool	normal	FALSE	yes
train	j	rainy	mild	normal	FALSE	yes
train	k	sunny	mild	normal	TRUE	yes
train	l	overcast	mild	high	TRUE	yes
train	m	overcast	hot	normal	FALSE	yes
test	n	rainy	mild	high	TRUE	no



Criterion for Attribute Selection

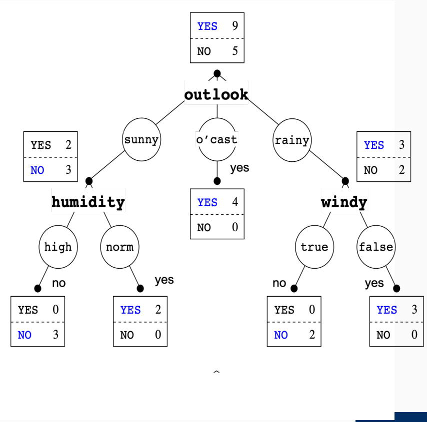
Choose an attribute to partition the data at the node such that each partition is as homogeneous (least impure) as possible to build small trees.



Choose an attribute to partition the data at the node such that each partition is as homogeneous (least impure) as possible to build small trees.

How to measure the purity of classes?

Use Entropy!



Entropy

Entropy is a measure of unpredictability:

- pure class distribution: low entropy
- evenly distributed distribution: high entropy

$$
\begin{align}
H &= \sum_{i}^{n} P(i) self\_information(i) \tag{1}\\
&= \sum_{i}^{n} P(i) log_2 \frac{1}{P(i)} \tag{2}\\
&= -\sum_{i}^{n} P(i) log_2 P(i) \tag{3}\\
& \tag{4}
\end{align}
$$

Example:

- Flip a normal coin 100 times, 52 heads, 48 tails: $H \approx 1$
- Flip a two-headed coin 100 times, $H = 0$

Criterion for Attribute Selection: Information Gain

- Select the attribute that has largest information gain
- Information Gain: Reduction of entropy before and after the data is partitioned using an attribute.

$$H(R_A|R) = H(R) - MeanInfo(x_1, \cdots, x_m)$$

$$MeanInfo(x_1, \cdots, x_m) = \sum_{i}^{m} w(x_i)H(x_i)$$

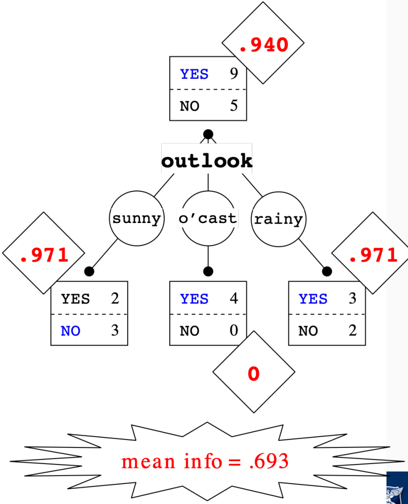
- $H(R)$: entropy of the instances in node R.
- MeanInfo: weighted entropy of the sub-nodes.
- $H(x_i)$: entropy of the instances in sub-node x_i.
- w_i: weight (proportion ratio) of each sub-node

Example

$$H(root) = -\left(\frac{9}{14}\log_2\frac{9}{14} + \frac{5}{14}\log_2\frac{5}{14}\right)$$

$$\begin{aligned}
& MeanInfo(outlook) && (5)\\
=& \frac{5}{14}\times 0.971 + \frac{4}{14}\times 0 + \frac{5}{14}\times 0.971 && (6)
\end{aligned}$$

$$\begin{aligned}
& IG(outlook|R) && (7)\\
=& H(root) - MeanInfo(outlook) && (8)\\
=& 0.940 - 0.693 && (9)\\
=& 0.247 && (10)
\end{aligned}$$



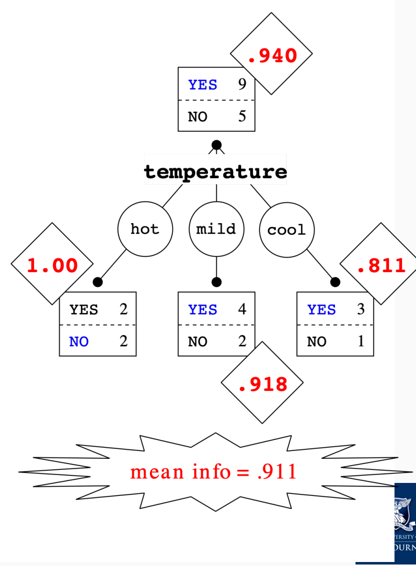
Example

$$IG(temperature|R) \tag{11}$$

$$= H(root) - MeanInfo(temperature) \tag{12}$$

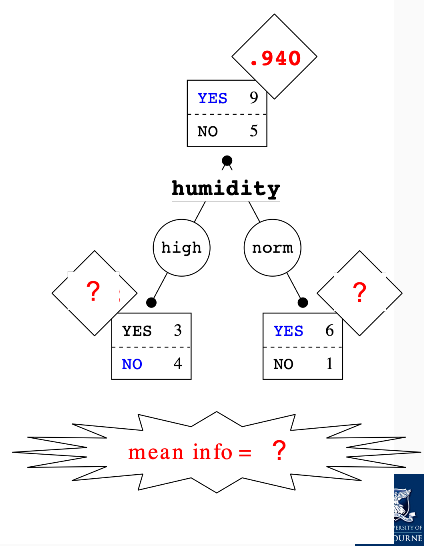
$$= 0.940 - 0.911 \tag{13}$$

$$= 0.029 \tag{14}$$



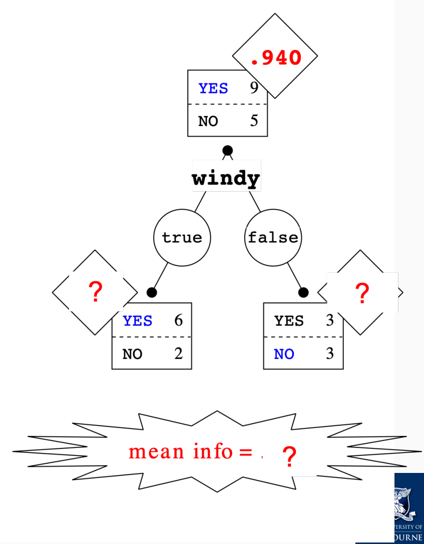
Example

$$IG(humidity|R) \tag{15}$$
$$=H(root) - MeanInfo(humidity) \tag{16}$$
$$=? \tag{17}$$



Example

$$IG(windy|R) \tag{18}$$
$$= H(root) - MeanInfo(windy) \tag{19}$$
$$= ? \tag{20}$$



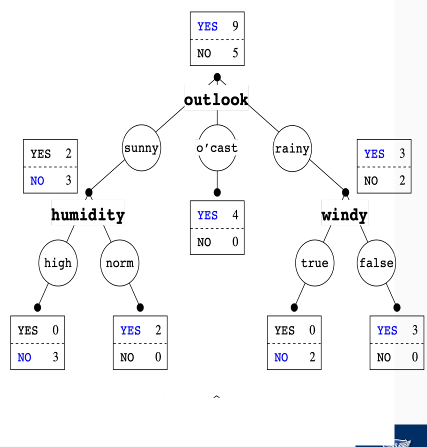
Example

$$H(sunny) = 0.971$$

$$IG(humidity|sunny) = 0.971$$

$$IG(temperature|sunny) = 0.571$$

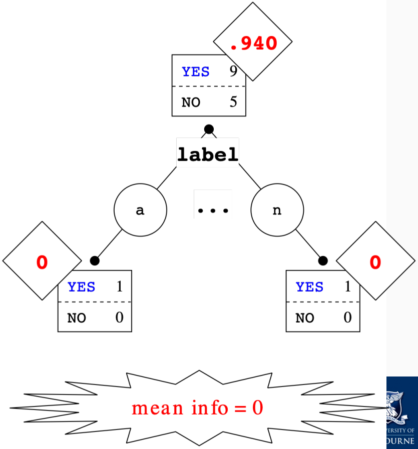
$$IG(windy|sunny) = 0.020$$



Shortcoming of Information Gain

- Information Gain tends to prefer highly-branching attributes
- It may result in overfitting problem.

	Outlook	Temperature	Humidity	Windy	Play
a	sunny	hot	high	FALSE	no
b	sunny	hot	high	TRUE	no
c	overcast	hot	high	FALSE	yes
d	rainy	mild	high	FALSE	yes
e	rainy	cool	normal	FALSE	yes
f	rainy	cool	normal	TRUE	no
g	overcast	cool	normal	TRUE	yes
h	sunny	mild	high	FALSE	no
i	sunny	cool	normal	FALSE	yes
j	rainy	mild	normal	FALSE	yes
k	sunny	mild	normal	TRUE	yes
l	overcast	mild	high	TRUE	yes
m	overcast	hot	normal	FALSE	yes
n	rainy	mild	high	TRUE	no



Solution: Gain Ratio

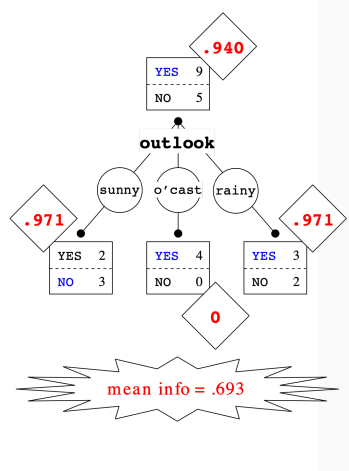
$$GR(R_A|R) = IG(R_A|R)/SI(R_A|R)$$

$$SI(R_A|R) = -\sum_{i}^{m} w(x_i) log_2 w(x_i)$$

Split info (SI): entropy of a given split (evenness of split).

$w(x_i)$: weight (proportion ratio) of each sub-node.

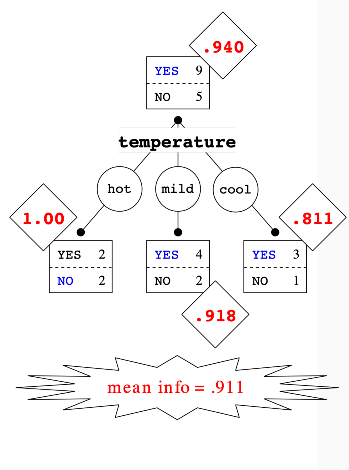
Example



$$SI(outlook|R) = -(\frac{5}{14} \times log_2 \frac{5}{14} + \frac{4}{14} \times log_2 \frac{4}{14} + \frac{5}{14} \times log_2 \frac{5}{14})$$

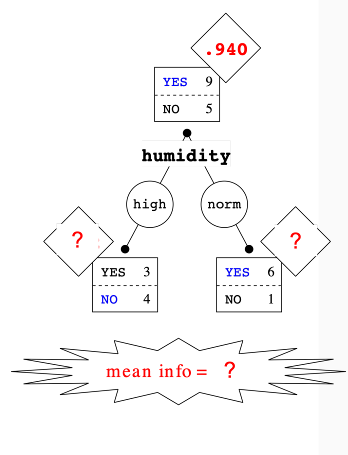
$$GR(outlook|R) = IG(outlook|R)/SI(outlook|R) = 0.247/1.577 = 0.157$$

Example



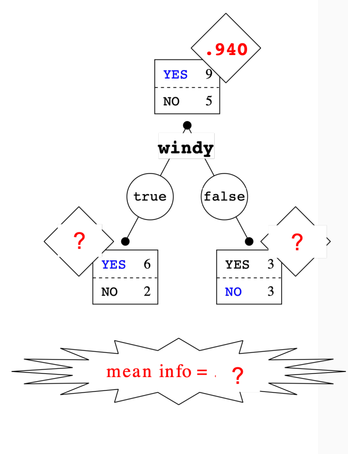
$$GR(temperature|R) = ?$$

Example



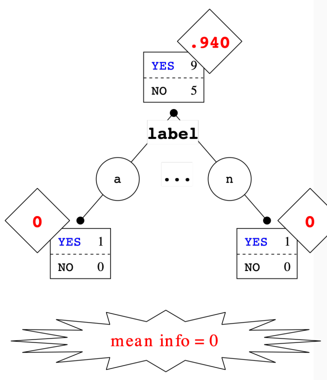
$$GR(humidity|R) = ?$$

Example



$$GR(windy|R) = ?$$

Example



$$GR(windy|R) = ?$$

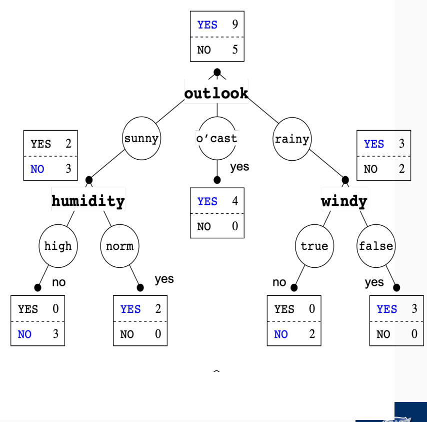
Properties of Decision Trees

Advantages

- Easy to read and understand.
- Requires less data preparation.
- Can handle feature with missing values.
- Can be expressed as disjunctive descriptions.
  i.e., $(\cdots \wedge \cdots \wedge \ldots) \vee (\cdots \wedge \cdots \wedge \ldots) \vee \ldots$

Disjunctive descriptions

disjunctive descriptions - yes: (outlook=sunny ∧ humidity=normal) ∨ (outlook=overcast) ∨ (outlook=rainy ∧ windy=false)



Disadvantages

- Loss of information for continuous variables.
- No guarantee to return the globally optimal decision:
  - Simple-to-complex, hill-climbing search through hypothesis space to construct trees.
  - It does not do back tracking on selected attributes.
- Information gain has bias for attributes with greater no. of values.
- Overfitting
  - New stopping criteria: choose best attribute only if IG/GR is greater than some threshold
  - Pruning: post-process the tree to remove undesirable branches (with few instances, or small IG/GR improvements)

```
#https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifi
from sklearn.model_selection import train_test_split
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier

iris = load_iris()
X = iris.data
y = iris.target
X_train, X_test, y_train, y_test = train_test_split(X, y)

estimator = DecisionTreeClassifier(criterion='entropy',max_depth=1) #decision stump
estimator.fit(X_train, y_train)
estimator.score(X_test,y_test)
```

```
0.5526315789473685
```

```
estimator = DecisionTreeClassifier(criterion='entropy')
estimator.fit(X_train, y_train)
estimator.score(X_test,y_test)
```

```
0.9736842105263158
```

https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

Summary

- How to build decision trees using IG/GR?
- How to test decision trees?
- What are the advantages/disadvantages of decision trees?

References

- Mitchell, Tom (1997). Machine Learning. Chapter 3: Decision Tree Learning.
- Tan et al (2006) Introduction to Data Mining. Section 4.3, pp 150-171.
- Quinlan, J. R. 1986. Induction of Decision Trees. Mach. Learn. 1, 1 (Mar. 1986), 81–106
- Lee, S., Lee, S., & Park, Y. (2007). A prediction model for success of services in e-commerce using decision tree: E-customer's attitude towards online service. Expert Systems with Applications, 33(3), 572-581.
- Delen, D., Walker, G., & Kadam, A. (2005). Predicting breast cancer survivability: a comparison of three data mining methods. Artificial intelligence in medicine, 34(2), 113-127.
- Tso, G. K., & Yau, K. K. (2007). Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks. Energy, 32(9), 1761-1768.

THE UNIVERSITY OF MELBOURNE