Lecture 3: 12 January, 2023

Pranabendu Misra
Slides by Madhavan Mukund

Data Mining and Machine Learning
January–April 2023

Supervised learning

- A set of items
  - Each item is characterized by attributes $(a_1, a_2, \ldots, a_k)$
  - Each item is assigned a class or category $c \leftarrow$ label
- Given a set of examples, build a ML model.
  Then predict category c' for a new item with attributes $(a'_1, a'_2, \ldots, a'_k)$

Supervised learning

- A set of items
  - Each item is characterized by attributes $(a_1, a_2, \ldots, a_k)$
  - Each item is assigned a class or category $c \leftarrow$ label
- Given a set of examples, build a ML model.
  Then predict category c' for a new item with attributes $(a'_1, a'_2, \ldots, a'_k)$
- Examples provided are called training data
- Aim is to learn a mathematical model that generalizes the training data
  - Model built from training data should extend to previously unseen inputs

Supervised learning

- A set of items
  - Each item is characterized by attributes $(a_1, a_2, \ldots, a_k)$
  - Each item is assigned a class or category c ← label
- Given a set of examples, build a ML model.
  Then predict category c' for a new item with attributes $(a'_1, a'_2, \ldots, a'_k)$
- Examples provided are called training data
- Aim is to learn a mathematical model that generalizes the training data
  - Model built from training data should extend to previously unseen inputs
- Classification problem
  - Usually assumed to binary — two classes
- Supervised learning as each training example has a category, i.e. it is *labeled*.

Example: Loan application data set

ID	Age	Has_job	Own_house	Credit_rating	Class
1	young	false	false	fair	**No**
2	young	false	false	good	**No**
3	young	true	false	good	**Yes**
4	young	true	true	fair	**Yes**
5	young	false	false	fair	**No**
6	middle	false	false	fair	**No**
7	middle	false	false	good	**No**
8	middle	true	true	good	**Yes**
9	middle	false	true	excellent	**Yes**
10	middle	false	true	excellent	**Yes**
11	old	false	true	excellent	**Yes**
12	old	false	true	good	**Yes**
13	old	true	false	good	**Yes**
14	old	true	false	excellent	**Yes**
15	old	false	false	fair	**No**

Basic assumptions

Fundamental assumption of machine learning

- Distribution of training examples is identical to distribution of unseen data

Basic assumptions

Fundamental assumption of machine learning

- Distribution of training examples is identical to distribution of unseen data

What does it mean to learn from the data?

- Build a model that does better than random guessing
  - In the loan data set, always saying Yes would be correct about $9/15$ of the time
- Performance should ideally improve with more training data

Basic assumptions

Fundamental assumption of machine learning

- Distribution of training examples is identical to distribution of unseen data

What does it mean to learn from the data?

- Build a model that does better than random guessing
  - In the loan data set, always saying Yes would be correct about $9/15$ of the time
- Performance should ideally improve with more training data

How do we evaluate the performance of a model?

- Model is optimized for the training data. How well does it work for unseen data?
- Don't know the correct answers in advance to compare — different from normal software verification

The road ahead

Many different models

- Decision trees
- Probabilistic models — naïve Bayes classifiers
- Models based on geometric separators
  - Support vector machines (SVM)
  - Neural networks

The road ahead

Many different models

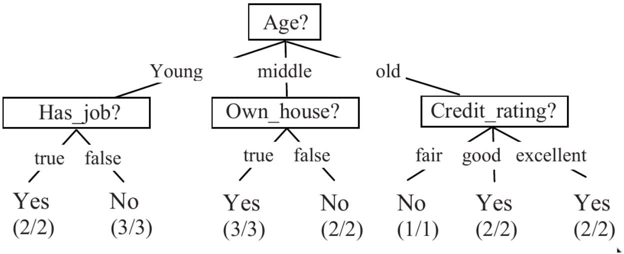
- Decision trees
- Probabilistic models — naïve Bayes classifiers
- Models based on geometric separators
  - Support vector machines (SVM)
  - Neural networks

Important issues related to supervised learning

- Evaluating models
- Ensuring that models generalize well to unseen data
  - A theoretical framework to provide some guarantees
- Strategies to deal with the training data bottleneck

Decision trees

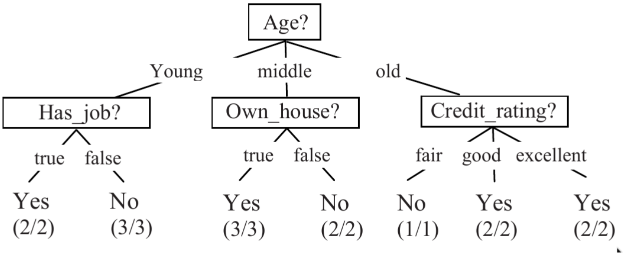
- Play "20 Questions" with the training data



ID	Age	Has_job	Own_house	Credit_rating	Class
1	young	false	false	fair	**No**
2	young	false	false	good	**No**
3	young	true	false	good	**Yes**
4	young	true	true	fair	**Yes**
5	young	false	false	fair	**No**
6	middle	false	false	fair	**No**
7	middle	false	false	good	**No**
8	middle	true	true	good	**Yes**
9	middle	false	true	excellent	**Yes**
10	middle	false	true	excellent	**Yes**
11	old	false	true	excellent	**Yes**
12	old	false	true	good	**Yes**
13	old	true	false	good	**Yes**
14	old	true	false	excellent	**Yes**
15	old	false	false	fair	**No**

Decision trees

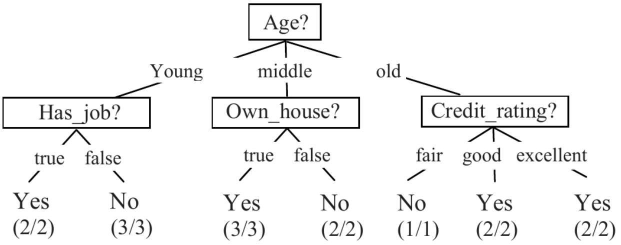
- Play "20 Questions" with the training data
- Query an attribute
  - Partition the training data based on the answer



ID	Age	Has_job	Own_house	Credit_rating	Class
1	young	false	false	fair	**No**
2	young	false	false	good	**No**
3	young	true	false	good	**Yes**
4	young	true	true	fair	**Yes**
5	young	false	false	fair	**No**
6	middle	false	false	fair	**No**
7	middle	false	false	good	**No**
8	middle	true	true	good	**Yes**
9	middle	false	true	excellent	**Yes**
10	middle	false	true	excellent	**Yes**
11	old	false	true	excellent	**Yes**
12	old	false	true	good	**Yes**
13	old	true	false	good	**Yes**
14	old	true	false	excellent	**Yes**
15	old	false	false	fair	**No**

Decision trees

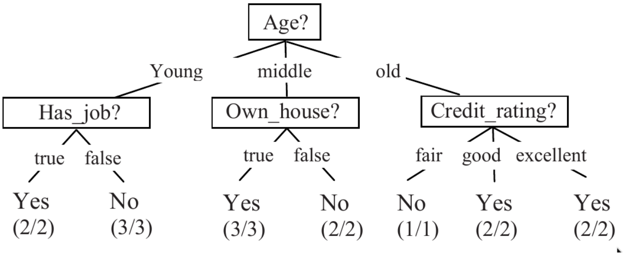
- Play "20 Questions" with the training data
- Query an attribute
  - Partition the training data based on the answer
- Repeat until you reach a partition with a uniform category



ID	Age	Has_job	Own_house	Credit_rating	Class
1	young	false	false	fair	**No**
2	young	false	false	good	**No**
3	young	true	false	good	**Yes**
4	young	true	true	fair	**Yes**
5	young	false	false	fair	**No**
6	middle	false	false	fair	**No**
7	middle	false	false	good	**No**
8	middle	true	true	good	**Yes**
9	middle	false	true	excellent	**Yes**
10	middle	false	true	excellent	**Yes**
11	old	false	true	excellent	**Yes**
12	old	false	true	good	**Yes**
13	old	true	false	good	**Yes**
14	old	true	false	excellent	**Yes**
15	old	false	false	fair	**No**

Decision trees

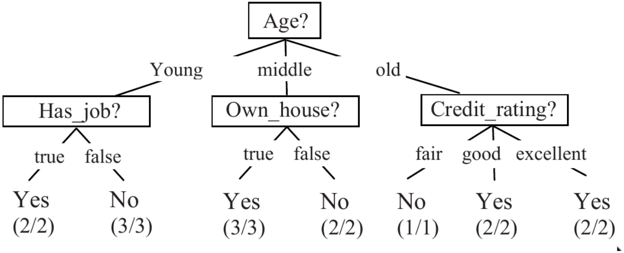
- Play "20 Questions" with the training data
- Query an attribute
  - Partition the training data based on the answer
- Repeat until you reach a partition with a uniform category
- Queries are adaptive
  - Different along each path, depends on history



ID	Age	Has_job	Own_house	Credit_rating	Class
1	young	false	false	fair	**No**
2	young	false	false	good	**No**
3	young	true	false	good	**Yes**
4	young	true	true	fair	**Yes**
5	young	false	false	fair	**No**
6	middle	false	false	fair	**No**
7	middle	false	false	good	**No**
8	middle	true	true	good	**Yes**
9	middle	false	true	excellent	**Yes**
10	middle	false	true	excellent	**Yes**
11	old	false	true	excellent	**Yes**
12	old	false	true	good	**Yes**
13	old	true	false	good	**Yes**
14	old	true	false	excellent	**Yes**
15	old	false	false	fair	**No**

Decision tree algorithm

A : current set of attributes



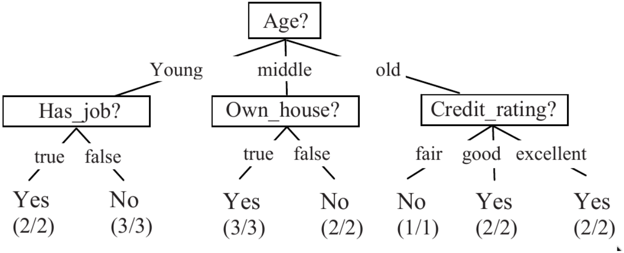
Decision tree algorithm

A : current set of attributes

Pick $a \in A$, create children corresponding to resulting partition with attributes $A \setminus \{a\}$

Stopping criterion:

- Current node has uniform class label
- A is empty — no more attributes to query



Decision tree algorithm

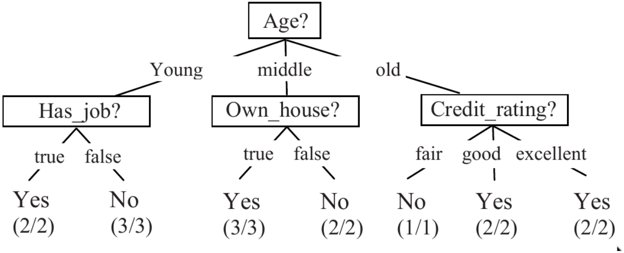
A : current set of attributes

Pick $a \in A$, create children corresponding to resulting partition with attributes $A \setminus \{a\}$

Stopping criterion:

- Current node has uniform class label
- A is empty — no more attributes to query

If a leaf node is not uniform, use majority class as prediction



Decision tree algorithm

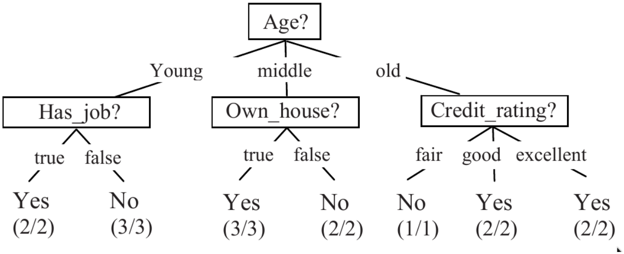
A : current set of attributes

Pick $a \in A$, create children corresponding to resulting partition with attributes $A \setminus \{a\}$

Stopping criterion:

- Current node has uniform class label
- A is empty — no more attributes to query

If a leaf node is not uniform, use majority class as prediction



- Non-uniform node — identical combination of attributes, but different classes

Decision tree algorithm

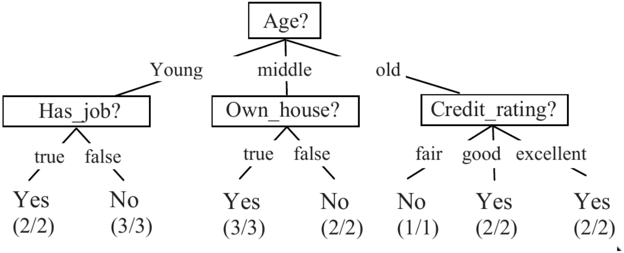
A : current set of attributes

Pick $a \in A$, create children corresponding to resulting partition with attributes $A \setminus \{a\}$

Stopping criterion:

- Current node has uniform class label
- A is empty — no more attributes to query

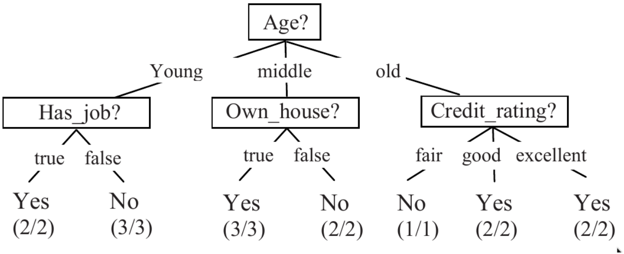
If a leaf node is not uniform, use majority class as prediction

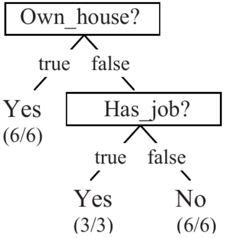


- Non-uniform node — identical combination of attributes, but different classes
- Given attributes may not capture all the criteria for classification. So two identical rows in the training data may have different labels!

Decision trees

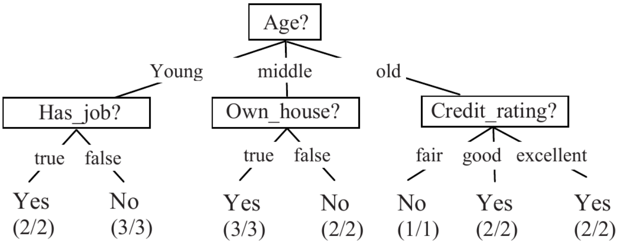
- Tree is not unique

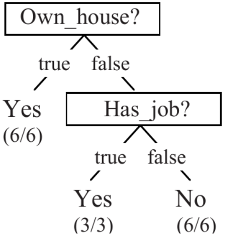




Decision trees

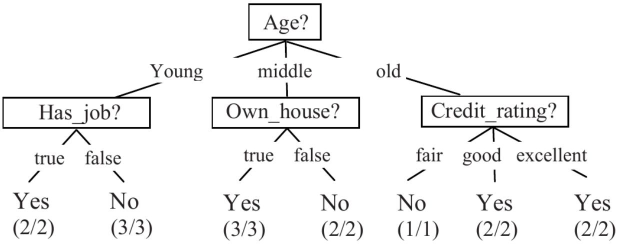
- Tree is not unique
- Which tree is better?

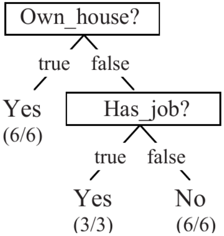




Decision trees

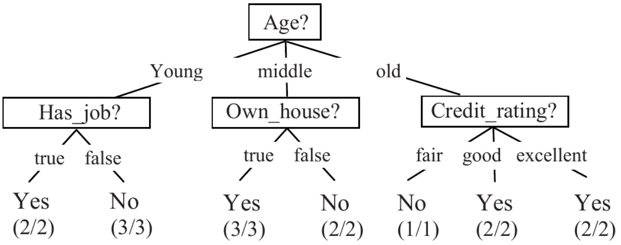
- Tree is not unique
- Which tree is better?
- Prefer small trees

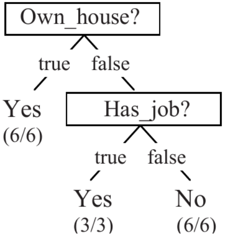




Decision trees

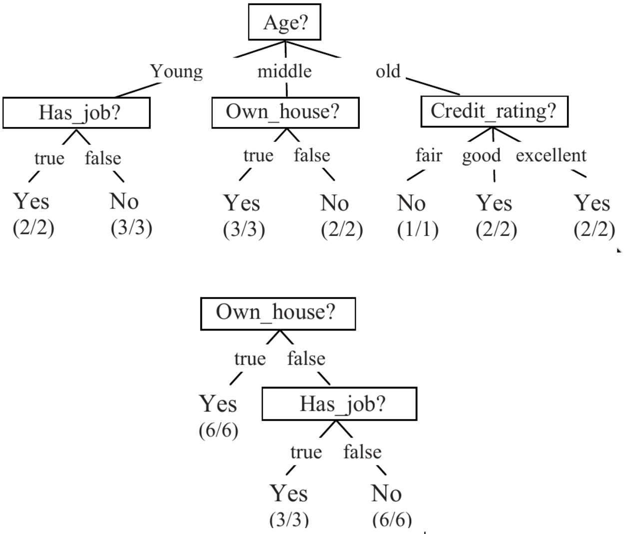
- Tree is not unique
- Which tree is better?
- Prefer small trees
  - Explainability





Decision trees

- Tree is not unique
- Which tree is better?
- Prefer small trees
  - Explainability
  - Generalize better (see later)

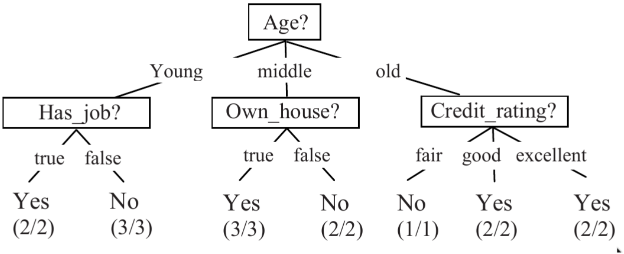


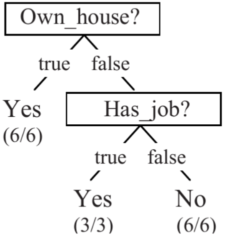
Decision trees

- Tree is not unique
- Which tree is better?
- Prefer small trees
  - Explainability
  - Generalize better (see later)

Unfortunately

- Finding smallest tree is NP-complete — for any definition of “smallest”



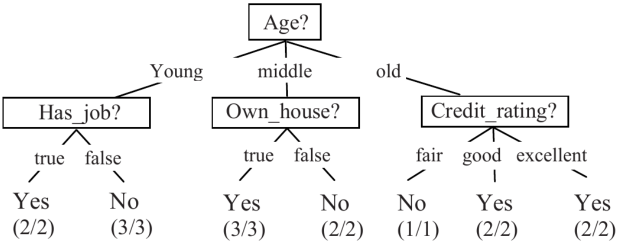


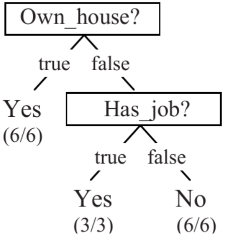
Decision trees

- Tree is not unique
- Which tree is better?
- Prefer small trees
  - Explainability
  - Generalize better (see later)

Unfortunately

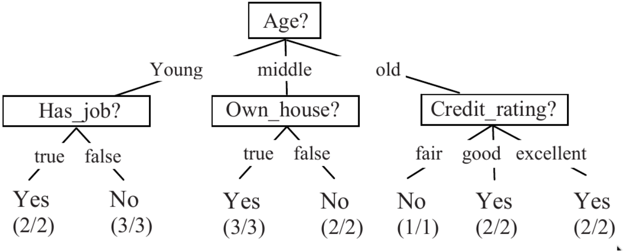
- Finding smallest tree is NP-complete — for any definition of “smallest”
- Instead, greedy heuristic

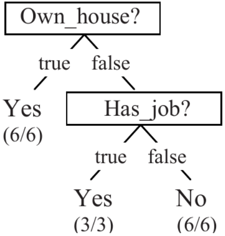




Greedy heuristic

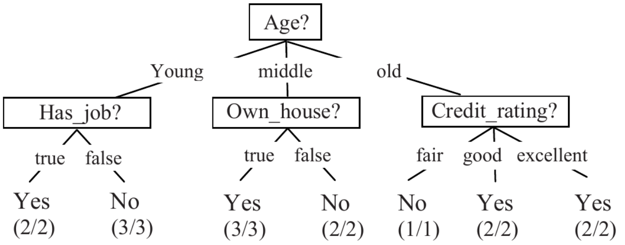
- Goal: partition with uniform category — pure leaf

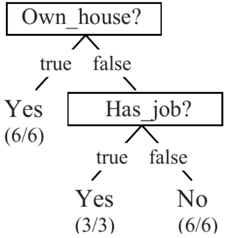




Greedy heuristic

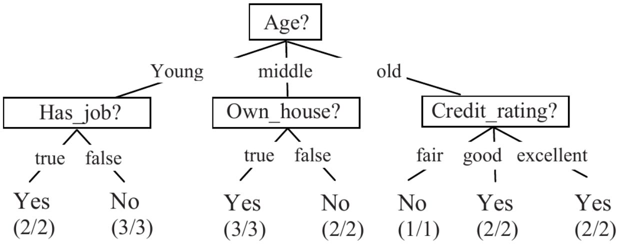
- Goal: partition with uniform category — pure leaf
- Impure node — best prediction is majority value

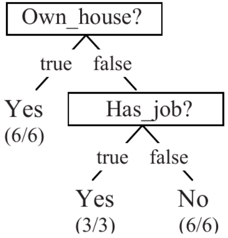




Greedy heuristic

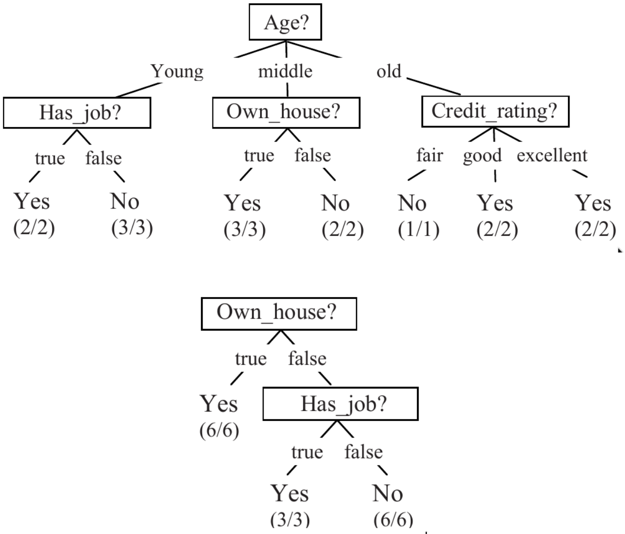
- Goal: partition with uniform category — pure leaf
- Impure node — best prediction is majority value
- Minority ratio is impurity of the training data-rows at a node.





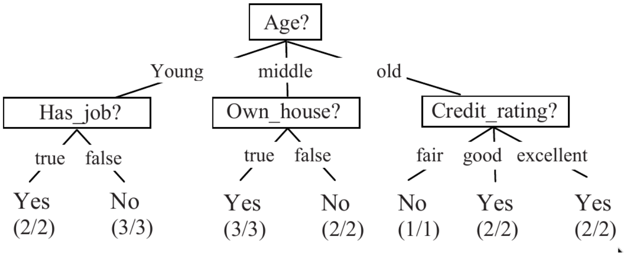
Greedy heuristic

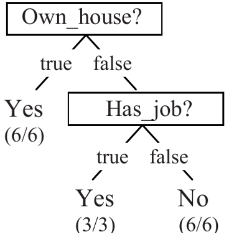
- Goal: partition with uniform category — pure leaf
- Impure node — best prediction is majority value
- Minority ratio is impurity of the training data-rows at a node.
- For each attribute, compute weighted average impurity of the child nodes obtained by splitting with respect to that attribute.



Greedy heuristic

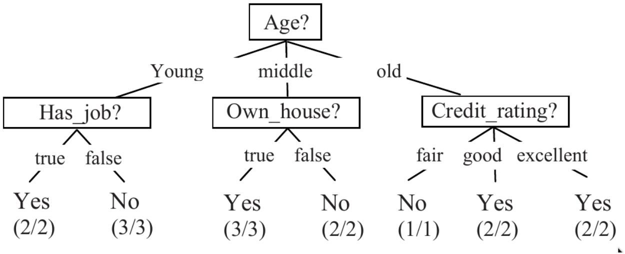
- Goal: partition with uniform category — pure leaf
- Impure node — best prediction is majority value
- Minority ratio is impurity of the training data-rows at a node.
- For each attribute, compute weighted average impurity of the child nodes obtained by splitting with respect to that attribute.
- Weight of a child-node is the fraction of training data-rows going to it.

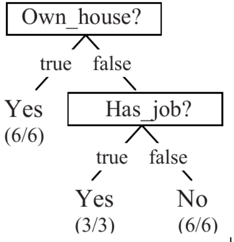




Greedy heuristic

- Goal: partition with uniform category — pure leaf
- Impure node — best prediction is majority value
- Minority ratio is impurity of the training data-rows at a node.
- For each attribute, compute weighted average impurity of the child nodes obtained by splitting with respect to that attribute.
- Weight of a child-node is the fraction of training data-rows going to it.
- Heuristic: Choose the attribute yielding minimum impurity

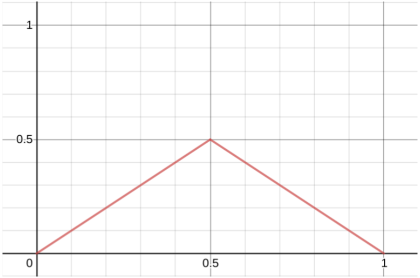




A better impurity function

What is wrong with this impurity function?

- Misclassification rate is linear



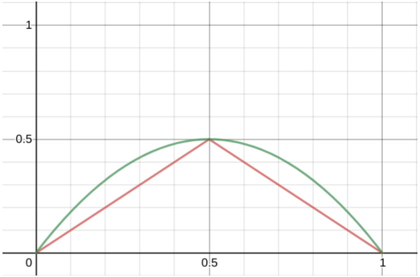
X-axis: fraction of data-rows at the node with label $c = 1$

Y-axis: impurity of the node

A better impurity function

What is wrong with this impurity function?

- Misclassification rate is linear
- Impurity measure that increases more sharply performs better, empirically
  - Intuitively, the green curve increases the urgency of moving towards a pure state.



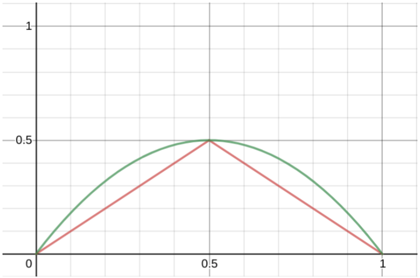
X-axis: fraction of data-rows at the node with label $c = 1$

Y-axis: impurity of the node

A better impurity function

What is wrong with this impurity function?

- Misclassification rate is linear
- Impurity measure that increases more sharply performs better, empirically
  - Intuitively, the green curve increases the urgency of moving towards a pure state.
- Entropy — [Quinlan]



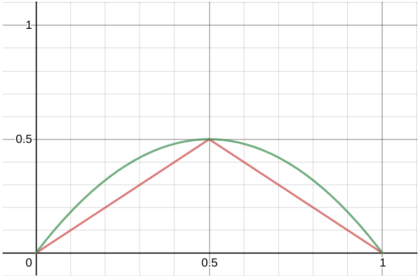
X-axis: fraction of data-rows at the node with label $c = 1$

Y-axis: impurity of the node

A better impurity function

What is wrong with this impurity function?

- Misclassification rate is linear
- Impurity measure that increases more sharply performs better, empirically
  - Intuitively, the green curve increases the urgency of moving towards a pure state.
- Entropy — [Quinlan]
- Gini index — [Breiman]



X-axis: fraction of data-rows at the node with label $c = 1$

Y-axis: impurity of the node

Entropy

- Information theoretic measure of randomness
- Minimum number of bits to transmit a message — [Shannon]

Entropy

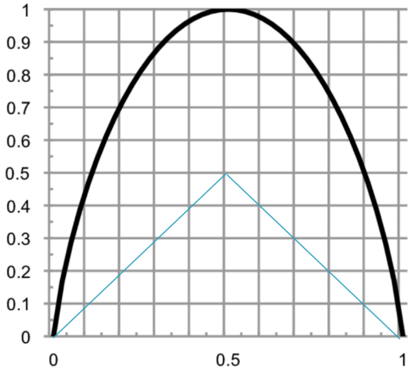
- Information theoretic measure of randomness
- Minimum number of bits to transmit a message — [Shannon]
- n data items
  - n_0 with $c = 0$, $p_0 = n_0/n$
  - n_1 with $c = 1$, $p_1 = n_1/n$

Entropy

- Information theoretic measure of randomness
- Minimum number of bits to transmit a message — [Shannon]
- n data items
  - n_0 with $c = 0$, $p_0 = n_0/n$
  - n_1 with $c = 1$, $p_1 = n_1/n$
- Entropy
  $E = -(p_0 \log_2 p_0 + p_1 \log_2 p_1)$

Entropy

- Information theoretic measure of randomness
- Minimum number of bits to transmit a message — [Shannon]
- n data items
  - n_0 with $c = 0$, $p_0 = n_0/n$
  - n_1 with $c = 1$, $p_1 = n_1/n$
- Entropy
  $E = -(p_0 \log_2 p_0 + p_1 \log_2 p_1)$
- Minimum when $p_0 = 1, p_1 = 0$ or vice versa — note, declare $0 \log_2 0$ to be 0
- Maximum when $p_0 = p_1 = 0.5$



$c = 1$

Gini Index

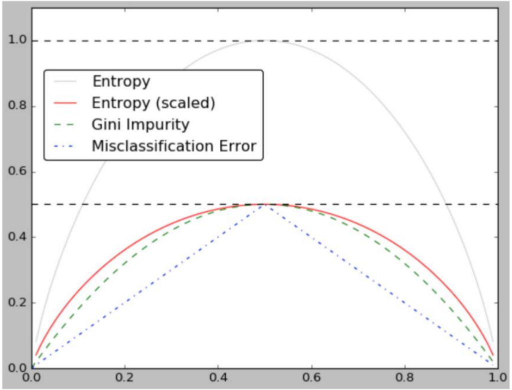
- Measure of unequal distribution of wealth
- Economics — [Corrado Gini]
- As before, n data items
  - n_0 with $c = 0$, $p_0 = n_0/n$
  - n_1 with $c = 1$, $p_1 = n_1/n$

Gini Index

- Measure of unequal distribution of wealth
- Economics — [Corrado Gini]
- As before, n data items
  - n_0 with $c = 0$, $p_0 = n_0/n$
  - n_1 with $c = 1$, $p_1 = n_1/n$
- Gini Index $G = 1 - (p_0^2 + p_1^2)$

Gini Index

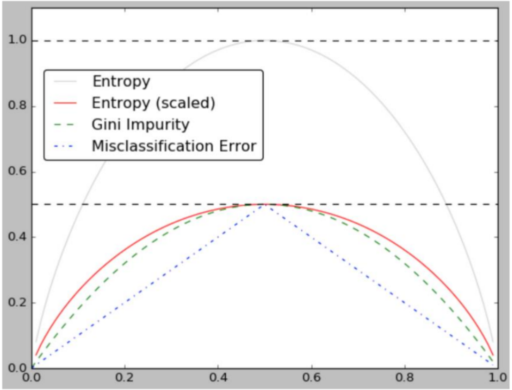
- Measure of unequal distribution of wealth
- Economics — [Corrado Gini]
- As before, n data items
  - n_0 with $c = 0$, $p_0 = n_0/n$
  - n_1 with $c = 1$, $p_1 = n_1/n$
- Gini Index $G = 1 - (p_0^2 + p_1^2)$
- $G = 0$ when $p_0 = 0$, $p_1 = 0$ or v.v.
  $G = 0.5$ when $p_0 = p_1 = 0.5$



$c = 1$

Gini Index

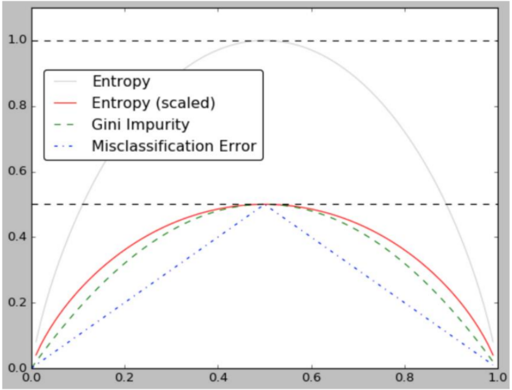
- Measure of unequal distribution of wealth
- Economics — [Corrado Gini]
- As before, n data items
  - n_0 with $c = 0$, $p_0 = n_0/n$
  - n_1 with $c = 1$, $p_1 = n_1/n$
- Gini Index $G = 1 - (p_0^2 + p_1^2)$
- $G = 0$ when $p_0 = 0$, $p_1 = 0$ or v.v.
  $G = 0.5$ when $p_0 = p_1 = 0.5$
- Entropy curve is slightly steeper, but Gini index is easier to compute



$c = 1$

Gini Index

- Measure of unequal distribution of wealth
- Economics — [Corrado Gini]
- As before, n data items
  - n_0 with $c = 0$, $p_0 = n_0/n$
  - n_1 with $c = 1$, $p_1 = n_1/n$
- Gini Index $G = 1 - (p_0^2 + p_1^2)$
- $G = 0$ when $p_0 = 0$, $p_1 = 0$ or v.v.
  $G = 0.5$ when $p_0 = p_1 = 0.5$
- Entropy curve is slightly steeper, but Gini index is easier to compute
- Decision tree libraries usually use Gini index



$c = 1$