Lecture 3: 12 April, 2021

Madhavan Mukund

https://www.cmi.ac.in/~madhavan

Data Mining and Machine Learning
April–July 2021

Market-Basket Analysis

- Items $I = \{i_1, i_2, \ldots, i_N\}$, transactions $T = \{t_1, t_2, \ldots, t_M\}$
- Identify all association rules $X \rightarrow Y$ meeting two thresholds
  - Confidence: $\dfrac{(X \cup Y).count}{X.count} \geq \chi$
  - Support: $\dfrac{(X \cup Y).count}{M} \geq \sigma$
- First identify frequent itemsets Z, such that $Z.count \geq \sigma M$
- Apriori algorithm
  - If X is not frequent, no $Y \supseteq X$ can be frequent
  - Find frequent sets levelwise: $F_1, F_2, \ldots$ are frequent itemsets of size $1, 2, \ldots$
- How do we generate association rules from frequent itemsets? ✓

Association rules

Naïve strategy

- For every frequent itemset Z
  - Enumerate all pairs $X, Y \subseteq Z$, $X \cap Y = \emptyset$
  - Check $\dfrac{(X \cup Y).count}{X.count} \geq \chi$

Association rules

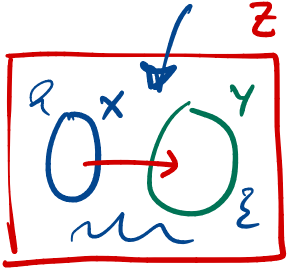
Naïve strategy

- For every frequent itemset Z
  - Enumerate all pairs $X, Y \subseteq Z$, $X \cap Y = \emptyset$
  - Check $\dfrac{(X \cup Y).count}{X.count} \geq \chi$
- Can we do better?

Association rules

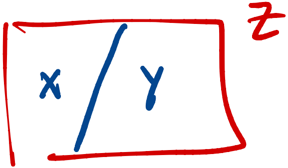
Naïve strategy

- For every frequent itemset Z
  - Enumerate all pairs $X, Y \subseteq Z$, $X \cap Y = \emptyset$
  - Check $\dfrac{(X \cup Y).count}{X.count} \geq \chi$
- Can we do better?
- Sufficient to check all partitions of Z
  - If $X, Y \subseteq Z$, $X \cup Y$ is also a frequent itemset



$X \cup Y \subseteq Z$

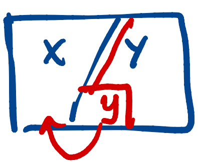
frequent



Association rules

- Sufficient to check all partitions of Z
- Suppose $Z = X \uplus Y$, $X \to Y$ is a valid rule and $y \in Y$
- What about $(X \cup \{y\}) \to Y \setminus \{y\}$?

$X \to Y$



$X \cup \{y\}$ $\quad$ $Y \setminus \{y\}$

Association rules

- Sufficient to check all partitions of Z
- Suppose $Z = X \uplus Y$, $X \to Y$ is a valid rule and $y \in Y$
- What about $(X \cup \{y\}) \to Y \setminus \{y\}$?
  - Know $\dfrac{(X \cup Y).count}{X.count} \geq \chi$
  - Check $\dfrac{(X \cup Y).count}{(X \cup \{y\}).count} \geq \chi$

Association rules

- Sufficient to check all partitions of Z
- Suppose $Z = X \uplus Y$, $X \to Y$ is a valid rule and $y \in Y$
- What about $(X \cup \{y\}) \to Y \setminus \{y\}$?
  - Know $\dfrac{(X \cup Y).count}{X.count} \geq \chi$
  - Check $\dfrac{(X \cup Y).count}{(X \cup \{y\}).count} \geq \chi$ — small denom.
  - X.count $\geq (X \cup \{y\})$.count, always
  - Second fraction has smaller denominator, so $(X \cup \{y\}) \to Y \setminus \{y\}$ is also a valid rule

$X \to Y$

Association rules

- Sufficient to check all partitions of Z
- Suppose $Z = X \uplus Y$, $X \to Y$ is a valid rule and $y \in Y$
- What about $(X \cup \{y\}) \to Y \setminus \{y\}$?
  - Know $\dfrac{(X \cup Y).count}{X.count} \geq \chi$
  - Check $\dfrac{(X \cup Y).count}{(X \cup \{y\}).count} \geq \chi$
  - X.count $\geq (X \cup \{y\})$.count, always
  - Second fraction has smaller denominator, so $(X \cup \{y\}) \to Y \setminus \{y\}$ is also a valid rule

Observation: Can use apriori principle again!

Z freq

$Z \setminus \{z\}$ freq

Apriori for association rules

- If $X \to Y$ is a valid rule, and $y \in Y$, $(X \cup \{y\}) \to Y \setminus \{y\}$ must also be a valid rule
- If $X \to Y$ is not a valid rule, and $x \in X$, $(X \setminus \{x\}) \to Y \cup \{x\}$ cannot be a valid rule

$X \to Y$

x

$X \setminus \{x\} \to Y \cup \{x\}$

Apriori for association rules

- If $X \to Y$ is a valid rule, and $y \in Y$, $(X \cup \{y\}) \to Y \setminus \{y\}$ must also be a valid rule
- If $X \to Y$ is not a valid rule, and $x \in X$, $(X \setminus \{x\}) \to Y \cup \{x\}$ cannot be a valid rule
- Start by checking rules with single element on the right
  - $Z \setminus z \to \{z\}$
- For $X \to \{x, y\}$ to be a valid rule, both $(X \cup \{x\}) \to \{y\}$ and $(X \cup \{y\}) \to \{x\}$ must be valid
- Explore partitions of each frequent itemset "level by level"

To check

$X \to \{x, y\}$

$X \cup \{x, y\} = Z$

$X \cup \{y\} \to \{x\}$

$X \cup \{x\} \to \{y\}$

Association rules for classification

- Classify documents by topic
- Consider the table on the right

Words in document	Topic
student, teach, school	Education
student, school	Education
teach, school, city, game	Education
cricket, football	Sports
football, player, spectator	Sports
cricket, coach, game, team	Sports
football, team, city, game	Sports

Association rules for classification

- Classify documents by topic
- Consider the table on the right
- Items are regular words and topics
- Documents are transactions — set of words and one topic

Words in document	Topic
student, teach, school	Education
student, school	Education
teach, school, city, game	Education
cricket, football	Sports
football, player, spectator	Sports
cricket, coach, game, team	Sports
football, team, city, game	Sports

Association rules for classification

- Classify documents by topic
- Consider the table on the right
- Items are regular words and topics
- Documents are transactions — set of words and one topic
- Look for association rules of a special form
  - $\{\text{student, school}\} \rightarrow \{\text{Education}\}$
  - $\{\text{game, team}\} \rightarrow \{\text{Sports}\}$

Words in document	**Topic**
student, teach, school	Education
student, school	Education
teach, school, city, game	Education
cricket, football	Sports
football, player, spectator	Sports
cricket, coach, game, team	Sports
football, team, city, game	Sports

Association rules for classification

- Classify documents by topic
- Consider the table on the right
- Items are regular words and topics
- Documents are transactions — set of words and one topic
- Look for association rules of a special form
  - {student, school} $\rightarrow$ {Education}
  - {game, team} $\rightarrow$ {Sports}
- Right hand side always a single topic
- Class Association Rules

Words in document	**Topic**
student, teach, school	Education
student, school	Education
teach, school, city, game	Education
cricket, football	Sports
football, player, spectator	Sports
cricket, coach, game, team	Sports
football, team, city, game	Sports

Cannot find associations that are not present

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
- Given a set of examples, predict 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 $(a_1, a_2, \ldots, a_k)$ $\in A_1$ $\in A_2$ $\in A_k$
  - Each item is assigned a class or category c label

- Given a set of examples, predict 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
- Given a set of examples, predict 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

Topics — multiclass

Sports? Y/N

↳ Arts? Y/N

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

9 Y
6 N
out of
15

Always Y
"Correct"
60% of time

Fundamental assumption of machine learning

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

Data from 2000 $\longrightarrow$ Generalize to 2021

Phone Ownership $\rightarrow$ Wealthy

Phone			Income > N
Y			Y
N			N

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

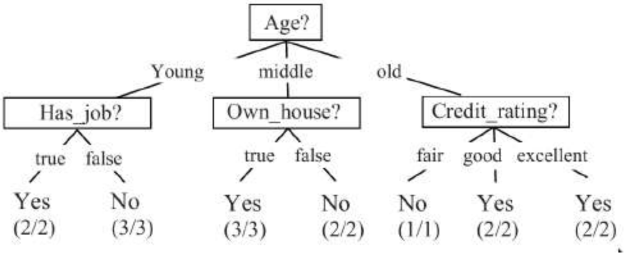
- Decision trees
- Probabilistic models — naïve Bayes classifiers
- Models based on geometric separators
  - Support vector machines (SVM)
  - Neural networks

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

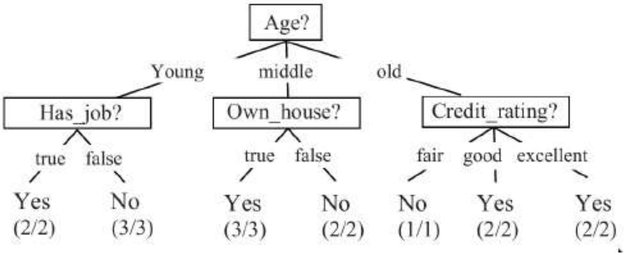
- Play "20 Questions" with the training data



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

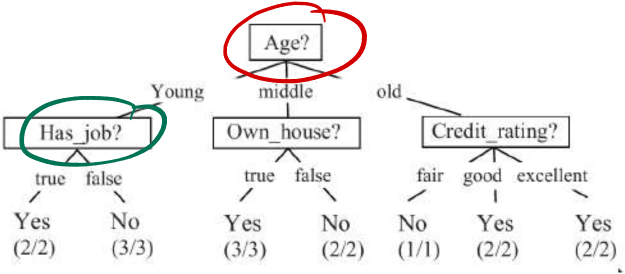
- Play "20 Questions" with the training data
- Query an attribute
  - Partition the training data based on the answer



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**

- 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

young hasjob nohouse poor credit



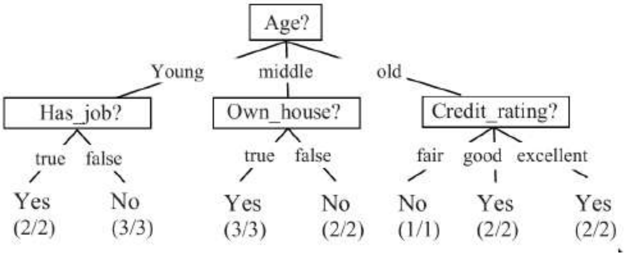
Y

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

2 1 2

Decision trees

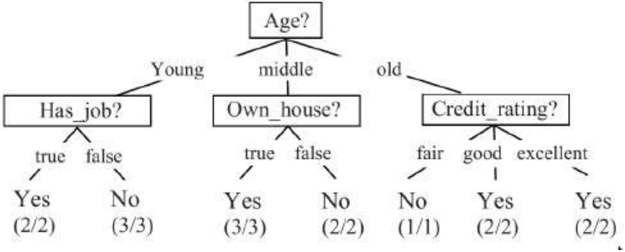
- Play "20 Questions" with the training data
- Query an attribute
  - Partition the training data based on the answer
- Repeat until you reach a partition with a uniform category
- Queries are adaptive
  - Different along each path, depends on history



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



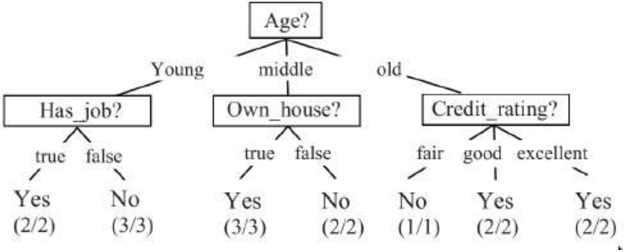
Decision tree algorithm

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

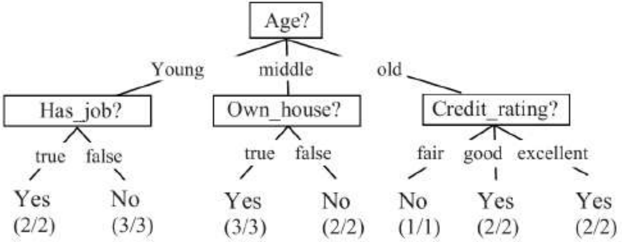
A : current set of attributes

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



Yes
3/5

Decision tree algorithm

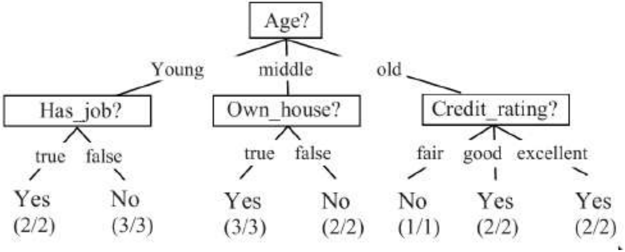
A : current set of attributes

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 leaf node — identical combination of attributes, but different classes

Decision tree algorithm

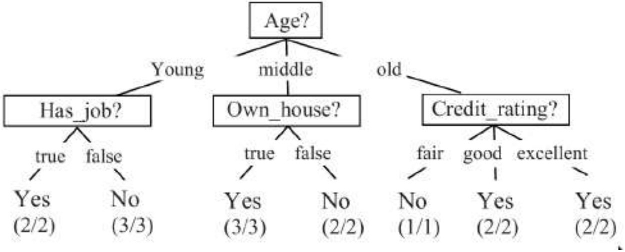
A : current set of attributes

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

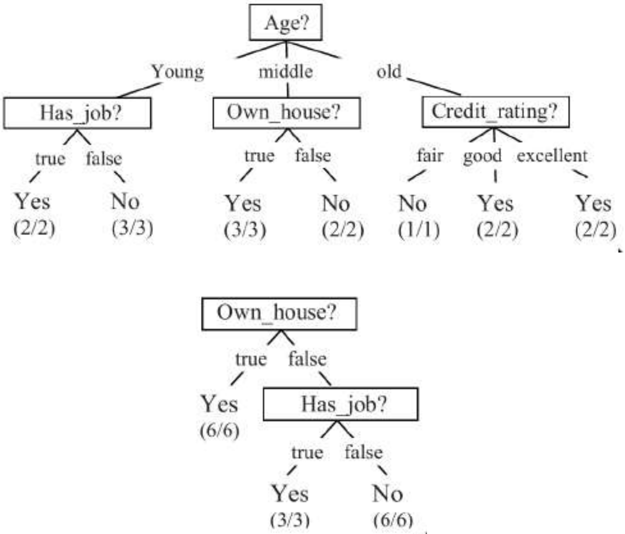
If a leaf node is not uniform, use majority class as prediction



- Non-uniform leaf node — identical combination of attributes, but different classes
- Attributes do not capture all criteria used for classification

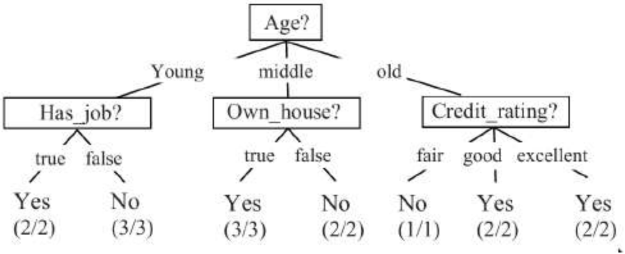
Decision trees

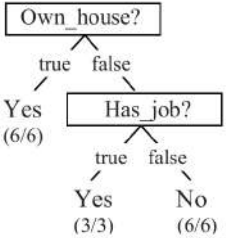
- Tree is not unique



Decision trees

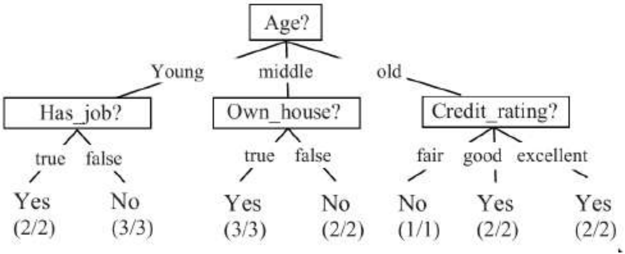
- Tree is not unique
- Which tree is better?

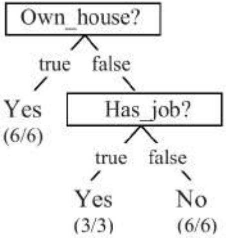




Decision trees

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

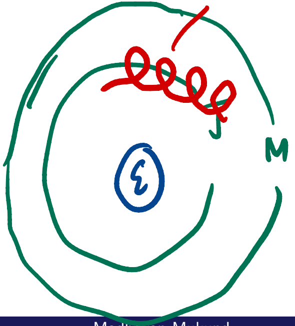




Decision trees

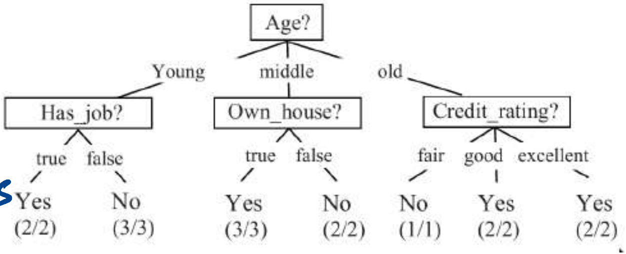
- Tree is not unique
- Which tree is better?
- Prefer small trees
  - Explainability

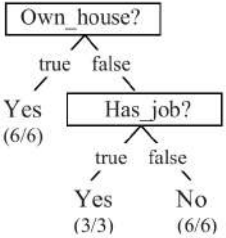
epicycles



Occam's Razor

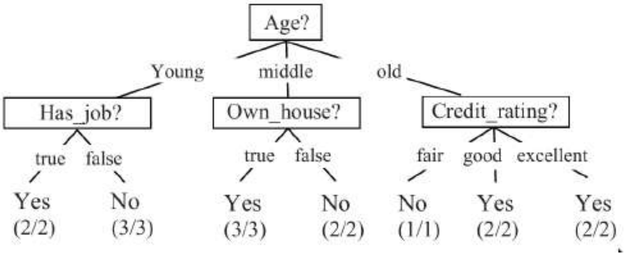
Prefer simpler explanations

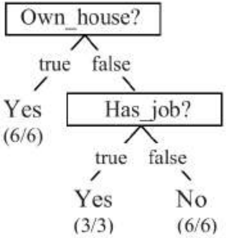




Decision trees

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



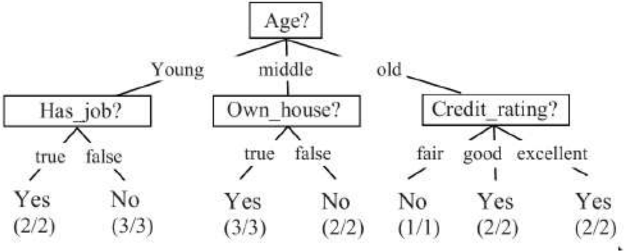


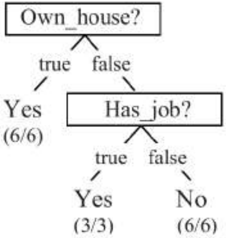
Decision trees

- 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”

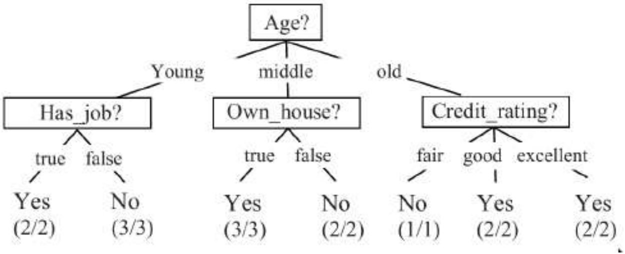




- Tree is not unique
- Which tree is better?
- Prefer small trees
  - Explainability
  - Generalize better (see later)

Unfortunately

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



Carton Packing