INTRODUCTION TO ARTIFICIAL INTELLIGENCE

3. INTRODUCTION TO **MACHINE LEARNING: SUPERVISED LEARNING – CLASSIFICATION**

The plan for today

- Review Supervised Learning
- Learn about Classification and Classification algorithms:
  - Decision Trees
  - K-nearest Neighbour

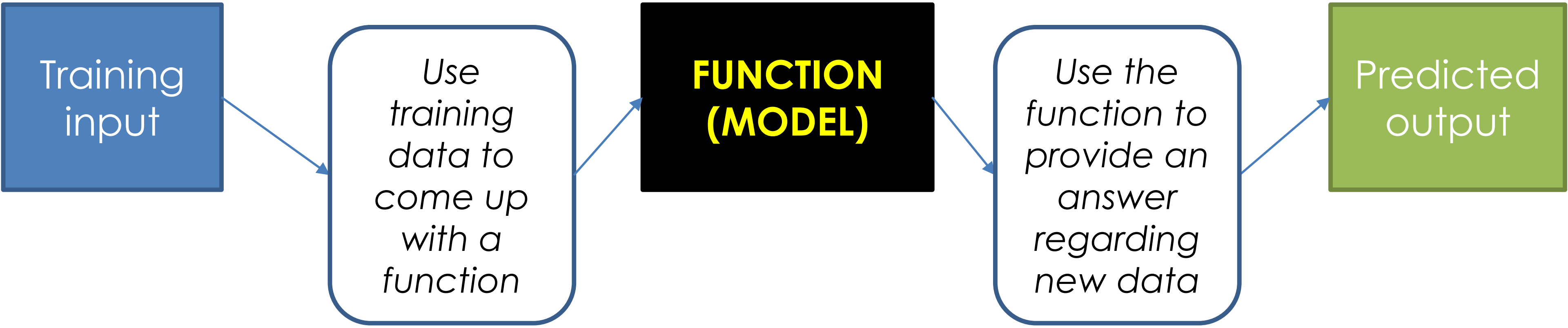
Let’s play

- http://en.akinator.com/

Machine learning types

- **Supervised learning:** The program is 'trained' on a given set of examples. It learns how to reach an accurate conclusion when given new data.
  - **We teach** the computer how to do something.
- **Unsupervised learning:** The program is given a bunch of data and must discover patterns and relationships in them.
  - We let the computer **learn** something **by itself.**
- **Reinforcement learning:** The program learns from the consequences of its actions (reward or punishment), rather than from being explicitly taught and it selects its actions on basis of its past experiences (exploitation) and also by new choices (exploration).

Supervised learning: in a nutshell

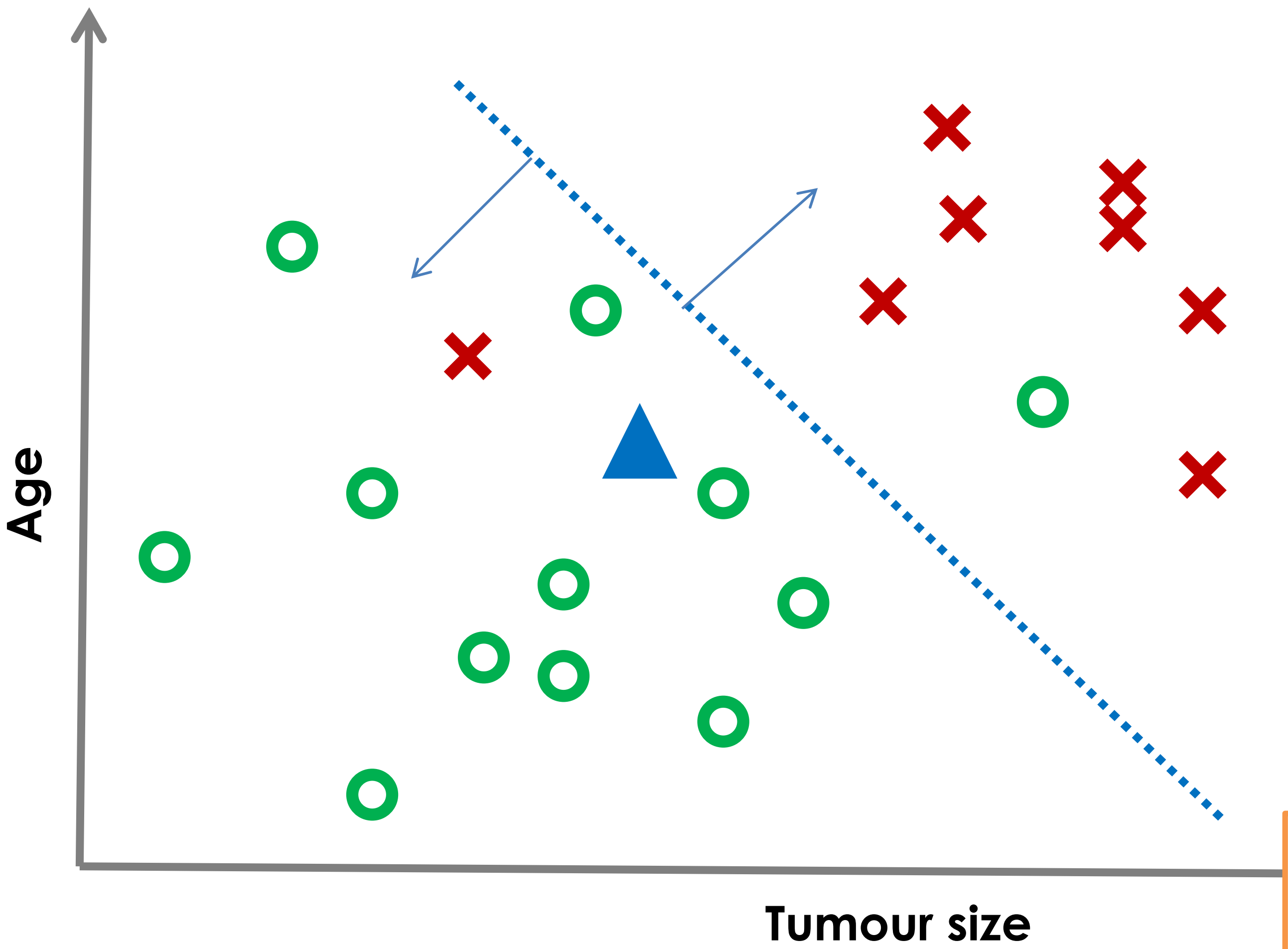


- When the prediction is a ***class (category)***, we use **classification**
- When the prediction is a ***number***, we use **regression**

Classification

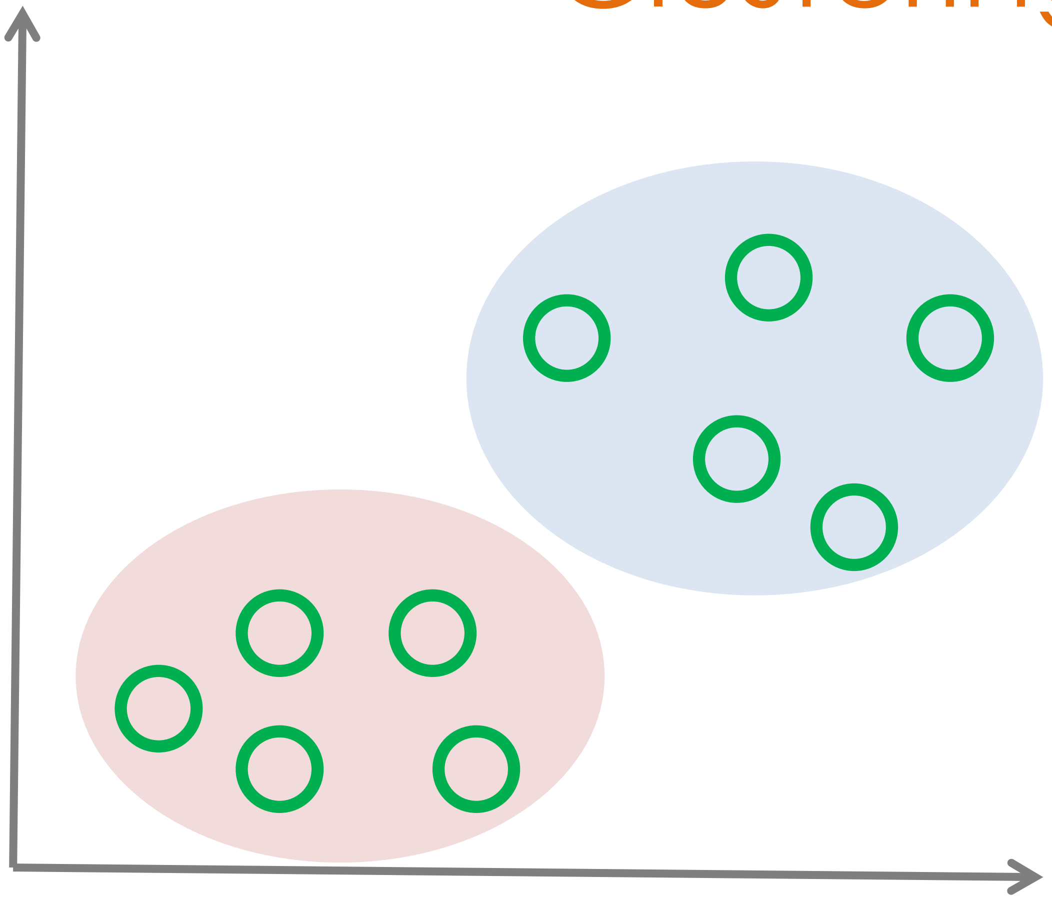
KITTEN or PUPPY?

Classification



Each data is labelled as Cancer (x) or Benign (o)

Clustering



No labels

We have to find some structure in the data set

An unsupervised algorithm may decide that the data belongs in two clusters (groups)

This algorithm is referred to as **Clustering** algorithm

Classification vs Clustering

- Classification (supervised learning)
  - Provide: labelled data
  - Learning task: be able to predict data

- Clustering (unsupervised learning)
  - Provide: unlabelled data
  - Learning task: group data by similarity

Quick quiz

- Where would you use a supervised (SL) or unsupervised learning (UL) algorithm?

1. Given email labelled as spam/not spam, train a spam filter **SL**
2. Given a set of news articles on the web, group them into a set of articles about the same story **UL**
3. Given a database of customer data, automatically discover market segments, and group customers into segments **UL**
4. Given a dataset of patients diagnosed with diabetes and individuals without diabetes, learn to classify new patients as having diabetes or not. **SL**

Classification (a formal definition)

- Given a collection of records (*training set*)
  - Each record contains a set of *attributes*, one of the attributes is the *class*.
- Find a *model* for class attribute as a function of the values of other attributes.
- Goal: <u>previously unseen</u> records should be assigned a class as accurately as possible.
  - A *test set* is used to determine the accuracy of the model. Usually, the given data set is divided into training and test sets, with training set used to build the model and test set used to validate it.
  - We'll talk about evaluation later in the session

Classification

Training data

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hiber-nates	Class Label
human	warm-blooded	hair	yes	no	no	yes	no	mammal
python	cold-blooded	scales	no	no	no	no	yes	reptile
salmon	cold-blooded	scales	no	yes	no	no	no	fish
whale	warm-blooded	hair	yes	yes	no	no	no	mammal
frog	cold-blooded	none	no	semi	no	yes	yes	amphibian
komodo dragon	cold-blooded	scales	no	no	no	yes	no	reptile
bat	warm-blooded	hair	yes	no	yes	yes	yes	mammal
pigeon	warm-blooded	feathers	no	no	yes	yes	no	bird
cat	warm-blooded	fur	yes	no	no	yes	no	mammal
leopard shark	cold-blooded	scales	yes	yes	no	no	no	fish
turtle	cold-blooded	scales	no	semi	no	yes	no	reptile
penguin	warm-blooded	feathers	no	semi	no	yes	no	bird
porcupine	warm-blooded	quills	yes	no	no	yes	yes	mammal
eel	cold-blooded	scales	no	yes	no	no	no	fish
salamander	cold-blooded	none	no	semi	no	yes	yes	amphibian

Use algorithm to learn model from training data

MODEL

Apply model to **classify** new data

New/test data

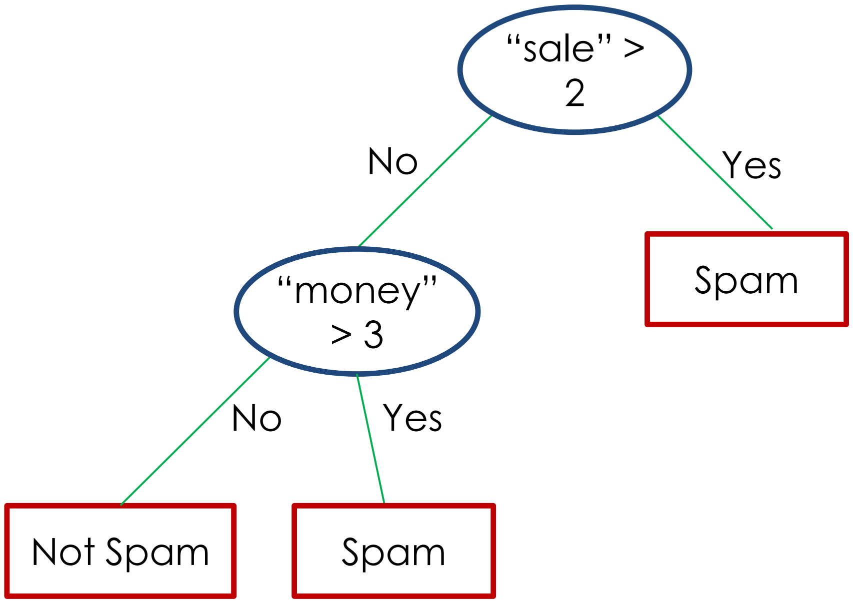
Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hiber-nates	Class Label
gila monster	cold-blooded	scales	no	no	no	yes	yes	?

Examples of classification algorithms

- **Decision Trees**
- **K-Nearest Neighbour**
- Neural Networks
- Support Vector Machines
- Naïve Bayes
- …

Decision tree

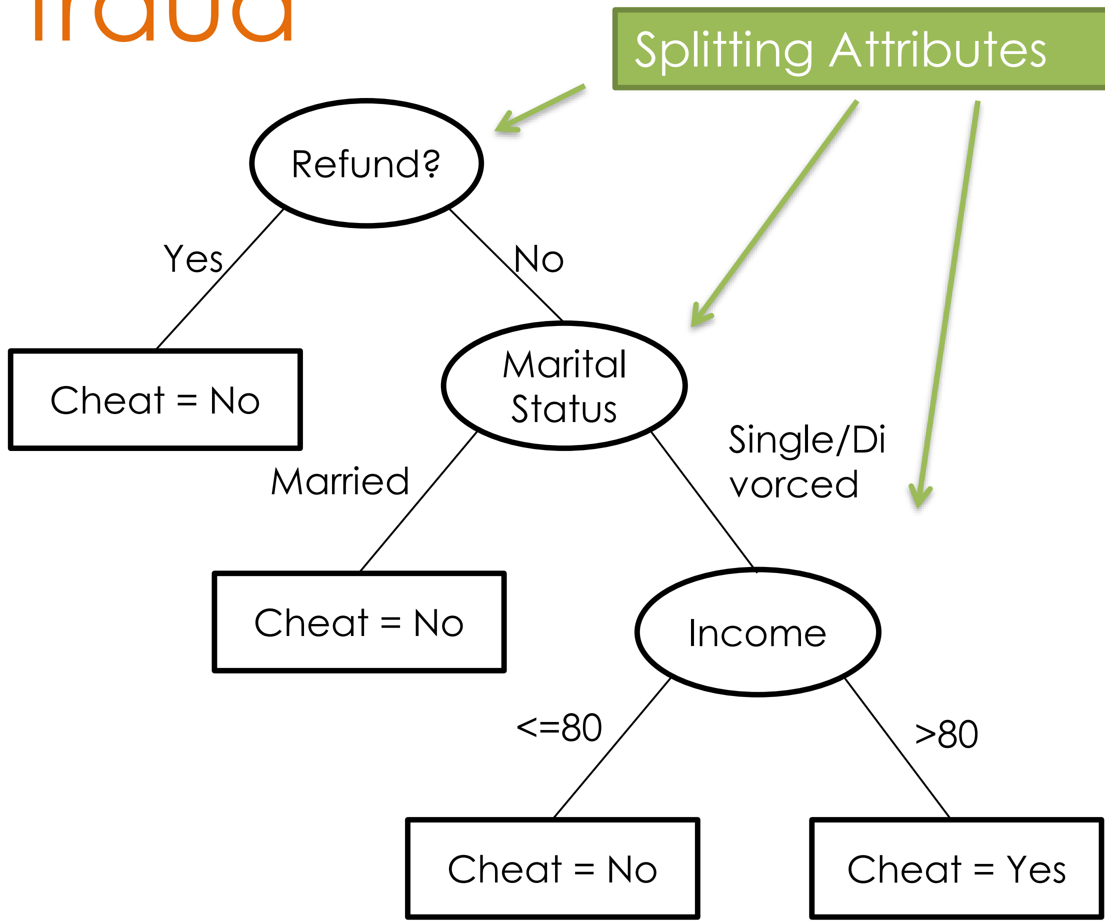
- Very popular classifier
- **Nodes** represent decisions
- **Arcs** represent possible answers
- **Terminal nodes** represent class labels



Decision tree: example from tax fraud

categorical categorical continuous class

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

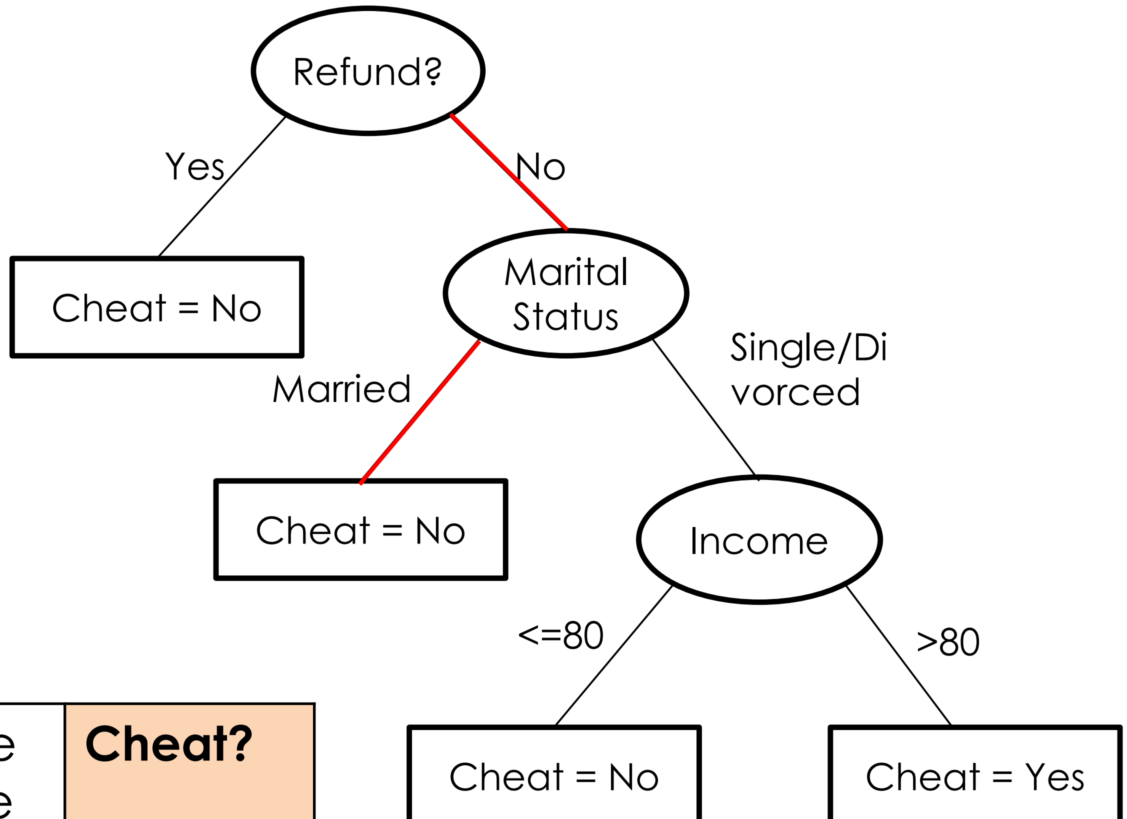


Training Data

Model: Decision Tree

Decision tree

Model: Decision Tree



Test Data

Refund	Marital Status	Taxable Income	**Cheat?**
No	Married	80K	**?**

Decision tree

categorical categorical continuous class

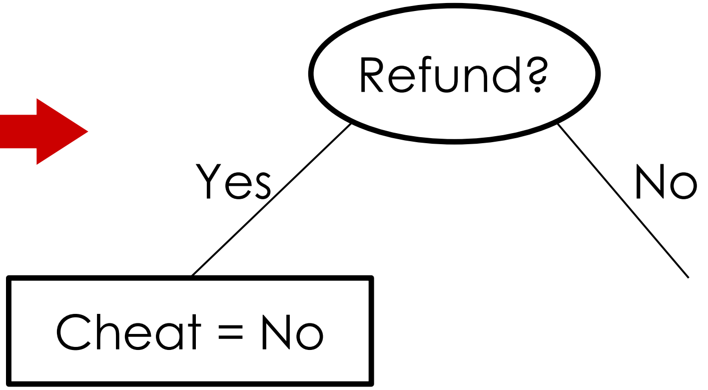
Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

Training Data

Decision tree

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data

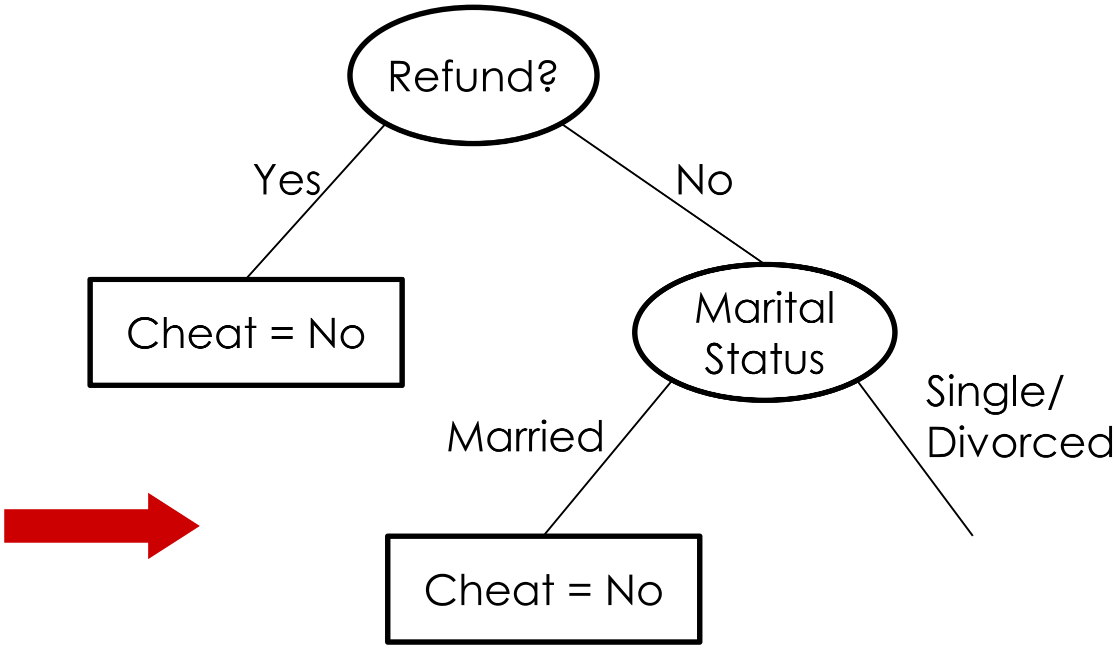


Model: Decision Tree

Decision tree

Tid	Refund	Marital Status	Taxable Income	Cheat
~~1~~	~~Yes~~	~~Single~~	~~125K~~	~~No~~
2	No	Married	100K	No
3	No	Single	70K	No
~~4~~	~~Yes~~	~~Married~~	~~120K~~	~~No~~
5	No	Divorced	95K	Yes
6	No	Married	60K	No
~~7~~	~~Yes~~	~~Divorced~~	~~220K~~	~~No~~
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data

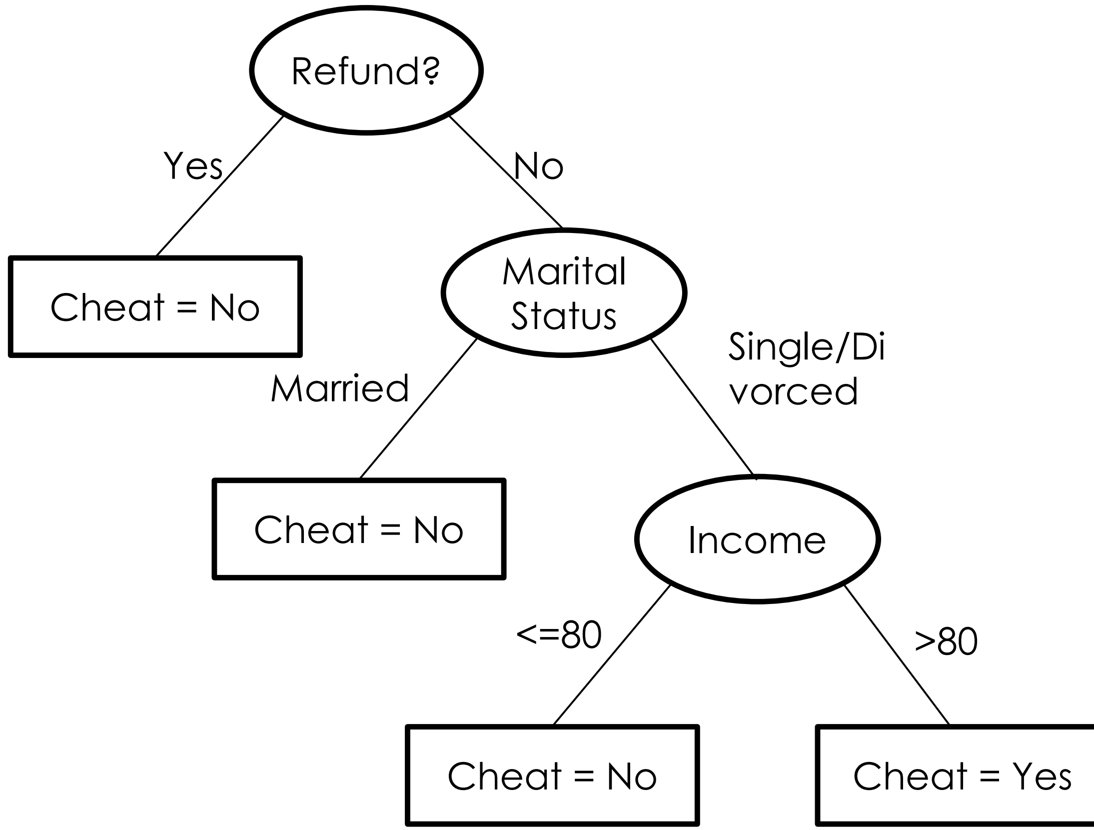


Model: Decision Tree

Decision tree

Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	~~Yes~~	~~Single~~	~~125K~~	~~**No**~~
2	~~No~~	~~Married~~	~~100K~~	~~**No**~~
3	No	Single	70K	**No**
4	~~Yes~~	~~Married~~	~~120K~~	~~**No**~~
5	No	Divorced	95K	**Yes**
6	~~No~~	~~Married~~	~~60K~~	~~**No**~~
7	~~Yes~~	~~Divorced~~	~~220K~~	~~**No**~~
8	No	Single	85K	**Yes**
9	~~No~~	~~Married~~	~~75K~~	~~**No**~~
10	No	Single	90K	**Yes**

Training Data



Model: Decision Tree

Decision tree algorithms

- Hunt's Algorithm (one of the earliest and basis for most existing algorithms)
- CART
- ID3, C4.5
- SLIQ,SPRINT
- …
- …

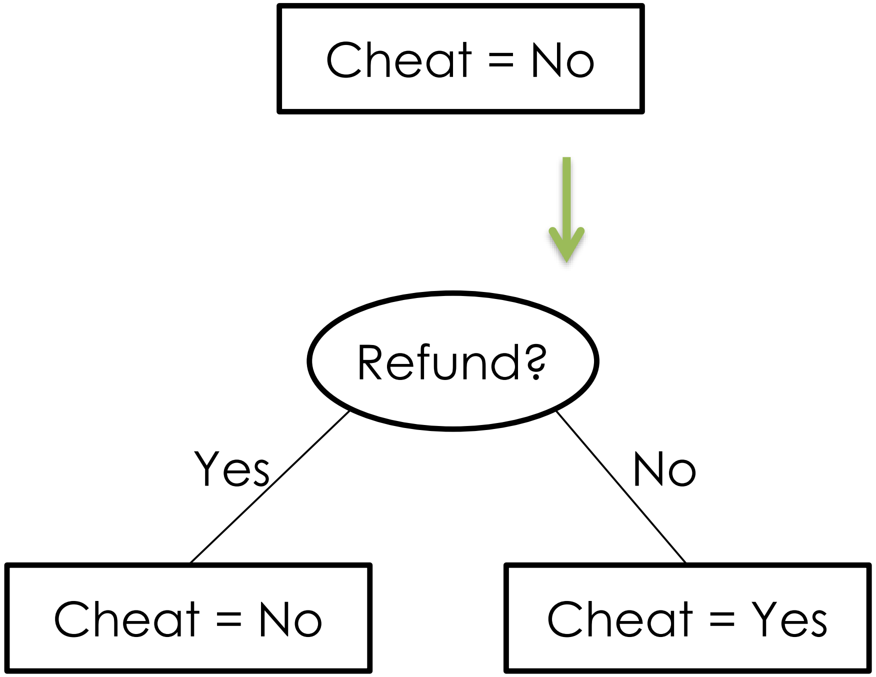
Building a decision tree

- Often known as *rule induction*
- Nodes are **repeatedly split** until all elements represented belong to one class
- Nodes then become terminal nodes
- Deciding which nodes to split next as well as the evaluation function used to split it depend on the algorithm

Building a decision tree algorithm (Hunt's)

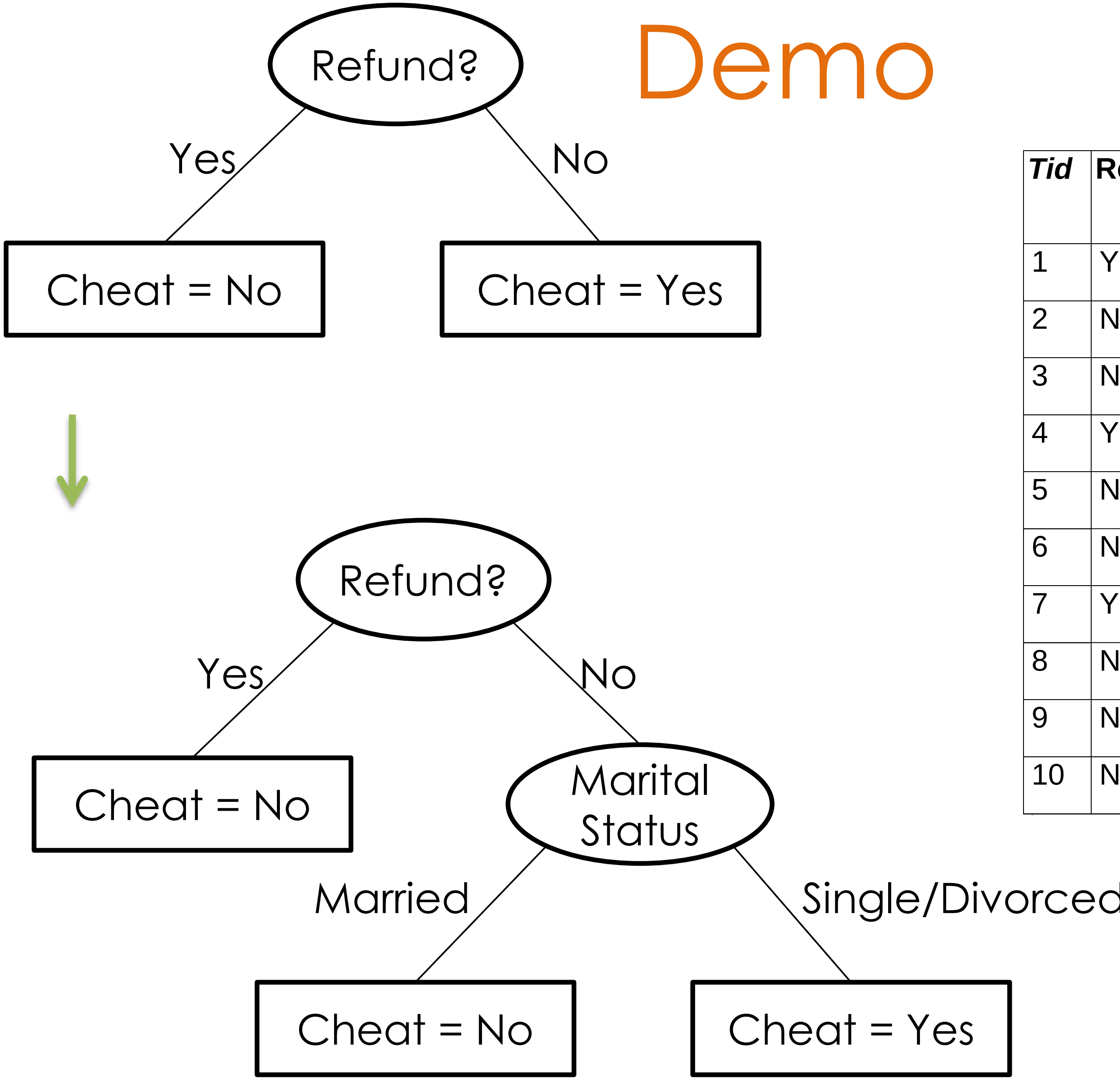
1. Let the set of training data be S. Put all S in a single tree node.
   – If some of the attributes are continuous-valued, make them discrete. For example, continuous age values can be binned into categories (under 18, 18-40, 41-65, over 65)
2. If all instances in S are in the same class, then stop.
3. Split the next node by selecting an attribute A from your list of attributes that **best splits** the objects in the node, and create a node.
4. Split the node according to the values of A.
5. Stop if either of the following conditions is met otherwise continue with Step 3:
   a) If this partition divides the data into subsets that belong to a single class and no other node needs splitting, or,
   b) If there are no remaining attributes on which the sample may be further divided.

Building a decision tree demo



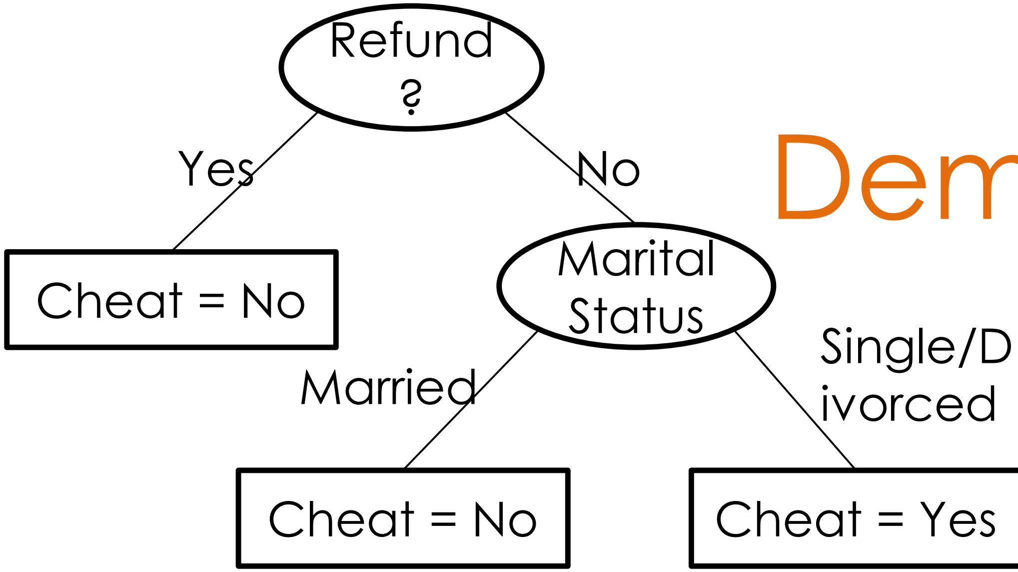
Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

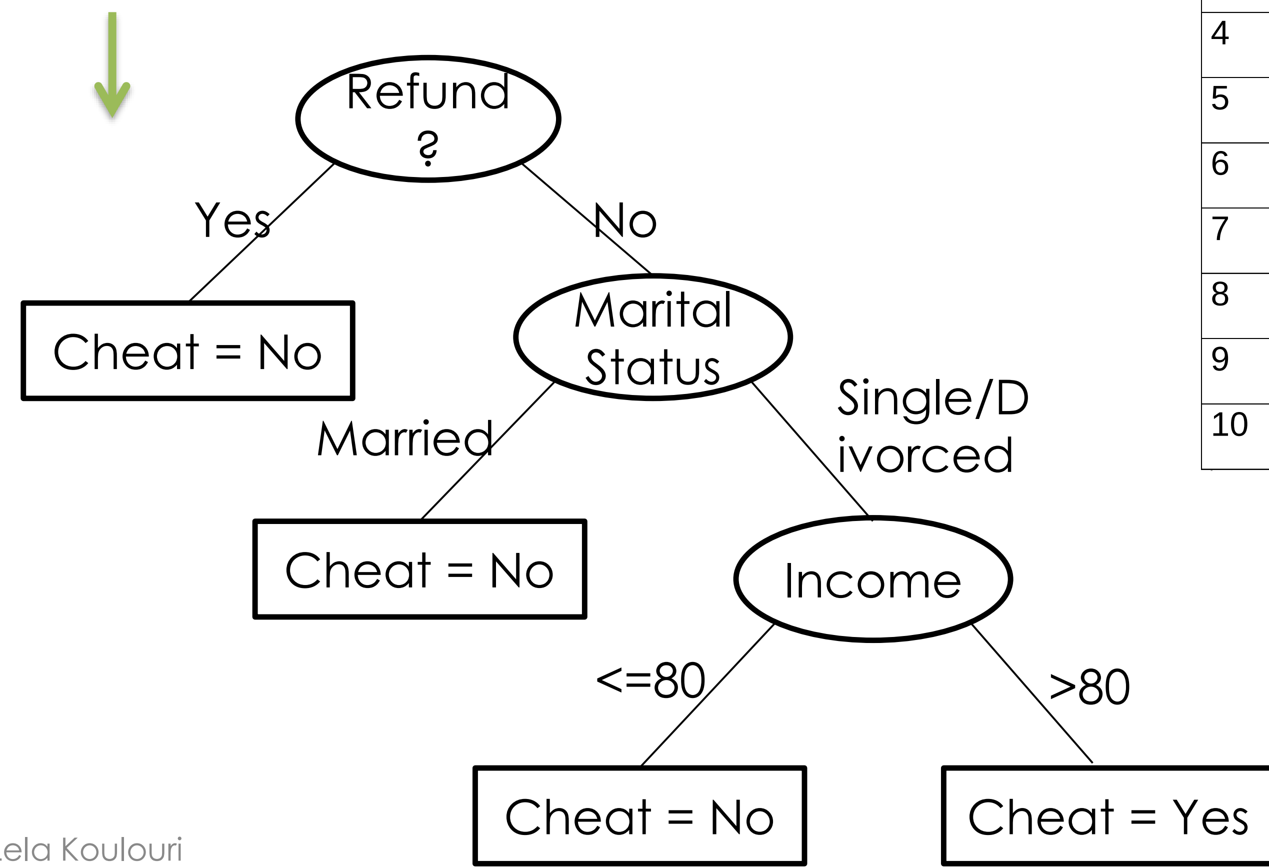
Demo



Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

Demo





Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

Building a decision tree

- How do we decide which attribute **best splits** the objects?
- Most commonly used measures to select best split:
  - GINI index
  - Entropy (information theory)

How to determine the **best split**

- Greedy approach:
  - We want nodes that give us **homogeneous**, **pure** classes
    - For example, in a 2-class situation, we split using attribute *X*, and all records go to Class0 and 0 to Class1. That is 0 impurity. Ideal!
    - If we split by attribute *B*, half of the records go to Class0 and the other half to Class1. That's is 0.5 impurity. Worst!

- Need a measure of node impurity (the lowest the better):

Attribute X

C0: 5
C1: 5

Non-homogeneous,

High degree of impurity

Attribute Y

C0: 9
C1: 1

Homogeneous,

Low degree of impurity

Deciding the best split: Entropy and GINI measures

- Let *p(j/t)* denote the relative frequency of class *j* (at a given node *t*).
- The most popular measures are:
  - The GINI index

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

  - Entropy (information theory)

$$Entropy(t) = -\sum_j p(j \mid t) \log p(j \mid t)$$

Computing impurity: GINI

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

X

C1	**0**
C2	**6**

P(C1) = 0/6 = 0   P(C2) = 6/6 = 1

Gini = 1 – $P(C1)^2$ – $P(C2)^2$ = 1 – 0 – 1 = 0

Y

C1	**1**
C2	**5**

P(C1) = 1/6   P(C2) = 5/6

Gini = 1 – $(1/6)^2$ – $(5/6)^2$ = 0.278

Z

C1	**2**
C2	**4**

P(C1) = 2/6   P(C2) = 4/6

Gini = 1 – $(2/6)^2$ – $(4/6)^2$ = 0.444

Which attribute (X, Y, Z) would give the best split?

Splitting Based on GINI

- Used in CART, SLIQ, SPRINT.
- When a node p is split into *k* partitions (children), the quality of split is computed as,

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

where, n_i = number of records at child i,
n = number of records at node p.

Example: splitting based on GINI

Owns home?	Married	Gender	Employed	Credit rating	Risk class
Yes	Yes	Male	Yes	A	B
No	No	Female	Yes	A	A
Yes	Yes	Female	Yes	B	C
Yes	No	Male	No	B	B
No	Yes	Female	Yes	B	C
No	No	Female	Yes	B	A
No	No	Male	No	B	B
Yes	No	Female	Yes	A	A
No	Yes	Female	Yes	A	C
Yes	Yes	Female	Yes	A	C

- **We have 10 records and 3 classes: A, B, C**
- **We calculate the GINI index for each attribute**
- **The attribute with the lowest GINI will be the one to split by!**

Married

Class	Yes	No
A		
B		
C		
Total		

Example: splitting based on GINI

Owns home?	*Married*	*Gender*	*Employed*	*Credit rating*	*Risk class*
Yes	Yes	Male	Yes	*A*	*B*
No	No	Female	Yes	*A*	*A*
Yes	Yes	Female	Yes	*B*	*C*
Yes	No	Male	No	*B*	*B*
No	Yes	Female	Yes	*B*	*C*
No	No	Female	Yes	*B*	*A*
No	No	Male	No	*B*	*B*
Yes	No	Female	Yes	*A*	*A*
No	Yes	Female	Yes	*A*	*C*
Yes	Yes	Female	Yes	*A*	*C*

- **We have 10 records and 3 classes: A, B, C**
- **We calculate the GINI index for each attribute**
- **The attribute with the lowest GINI will be the one to split by!**

Married

Class	Yes	No
A	0	**3**
B	**1**	**2**
C	**4**	0
Total	**5**	**5**

Example: splitting based on GINI

Owns home?	*Married*	*Gender*	*Employed*	*Credit rating*	*Risk class*
Yes	Yes	Male	Yes	*A*	*B*
No	No	Female	Yes	*A*	*A*
Yes	Yes	Female	Yes	*B*	*C*
Yes	No	Male	No	*B*	*B*
No	Yes	Female	Yes	*B*	*C*
No	No	Female	Yes	*B*	*A*
No	No	Male	No	*B*	*B*
Yes	No	Female	Yes	*A*	*A*
No	Yes	Female	Yes	*A*	*C*
Yes	Yes	Female	Yes	*A*	*C*

- **We have 10 records and 3 classes: A, B, C**
- **We calculate the GINI index for each attribute**
- **The attribute with the lowest GINI will be the one to split by!**

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

Married

Class	Yes	No
A	0	**3**
B	**1**	**2**
C	**4**	0
Total	**5**	**5**

Gini (Y) = $1 - (1/5)^2 - (4/5)^2 =$ **0.32**

Gini (N) = $1 - (3/5)^2 - (2/5)^2 =$ **0.48**

Example: splitting based on GINI

Owns home?	*Married*	*Gender*	*Employed*	*Credit rating*	*Risk class*
Yes	Yes	Male	Yes	*A*	*B*
No	No	Female	Yes	*A*	*A*
Yes	Yes	Female	Yes	*B*	*C*
Yes	No	Male	No	*B*	*B*
No	Yes	Female	Yes	*B*	*C*
No	No	Female	Yes	*B*	*A*
No	No	Male	No	*B*	*B*
Yes	No	Female	Yes	*A*	*A*
No	Yes	Female	Yes	*A*	*C*
Yes	Yes	Female	Yes	*A*	*C*

- **We have 10 records and 3 classes: A, B, C**
- **We calculate the GINI index for each attribute**
- **The attribute with the lowest GINI will be the one to split by!**

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

Married

Class	Yes	No
A	0	3
B	1	2
C	4	0
Total	5	5

Gini (Y) = $1 - (1/5)^2 - (4/5)^2 = \mathbf{0.32}$
Gini (N) = $1 - (3/5)^2 - (2/5)^2 = \mathbf{0.48}$

Total Gini is G = $5/10 * \mathbf{0.32} + 5/10 * \mathbf{0.48}$
= **0.40**

Example: splitting based on GINI

Owns home?	*Married*	*Gender*	*Employed*	*Credit rating*	*Risk class*
Yes	Yes	Male	Yes	*A*	*B*
No	No	Female	Yes	*A*	*A*
Yes	Yes	Female	Yes	*B*	*C*
Yes	No	Male	No	*B*	*B*
No	Yes	Female	Yes	*B*	*C*
No	No	Female	Yes	*B*	*A*
No	No	Male	No	*B*	*B*
Yes	No	Female	Yes	*A*	*A*
No	Yes	Female	Yes	*A*	*C*
Yes	Yes	Female	Yes	*A*	*C*

- **We have 10 records and 3 classes: A, B, C**
- **We calculate the GINI index for each attribute**
- **The attribute with the lowest GINI will be the one to split by!**

Gender

Class	Female	Male
A		
B		
C		
Total		

Gini (F) = ?
Gini (M) = ?

Total Gini is G =

Example: splitting based on GINI

Owns home?	*Married*	*Gender*	*Employed*	*Credit rating*	*Risk class*
Yes	Yes	Male	Yes	*A*	*B*
No	No	Female	Yes	*A*	*A*
Yes	Yes	Female	Yes	*B*	*C*
Yes	No	Male	No	*B*	*B*
No	Yes	Female	Yes	*B*	*C*
No	No	Female	Yes	*B*	*A*
No	No	Male	No	*B*	*B*
Yes	No	Female	Yes	*A*	*A*
No	Yes	Female	Yes	*A*	*C*
Yes	Yes	Female	Yes	*A*	*C*

- **We have 10 records and 3 classes: A, B, C**
- **We calculate the GINI index for each attribute**
- **The attribute with the lowest GINI will be the one to split by!**

Gender

Class	Male	Female
A	0	**3**
B	**3**	0
C	0	4
Total	**3**	**7**

Gini (M) = 1 – 1= 0

Gini (F) = 1 – $(3/7)^2$ – $(4/7)^2$ = 0.490

Total Gini is G = 3/10 * 0 + 7/10 * 0.490

= **0.343**

Example: splitting based on GINI

Owns home?	*Married*	*Gender*	*Employed*	*Credit rating*	*Risk class*
Yes	Yes	Male	Yes	*A*	*B*
No	No	Female	Yes	*A*	*A*
Yes	Yes	Female	Yes	*B*	*C*
Yes	No	Male	No	*B*	*B*
No	Yes	Female	Yes	*B*	*C*
No	No	Female	Yes	*B*	*A*
No	No	Male	No	*B*	*B*
Yes	No	Female	Yes	*A*	*A*
No	Yes	Female	Yes	*A*	*C*
Yes	Yes	Female	Yes	*A*	*C*

- **We have 10 records and 3 classes: A, B, C**
- **We calculate the GINI index for each attribute**
- **The attribute with the lowest GINI will be the one to split by!**

- **Owns home**

Total Gini index = 0.64

- **Employed**

Total Gini index = 0.475

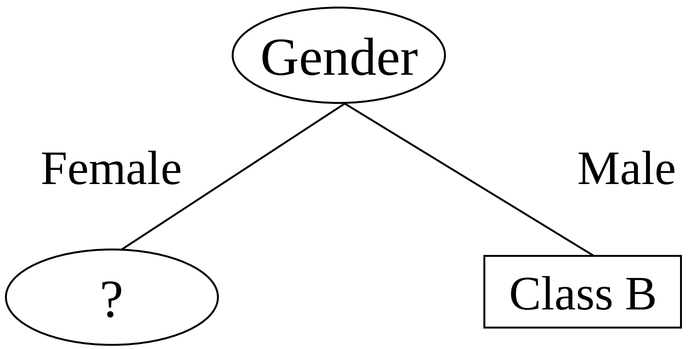
- **Credit Rating**

Total Gini index = 0.64

Calculations as part of Tutorial exercise 1

Example: splitting based on GINI

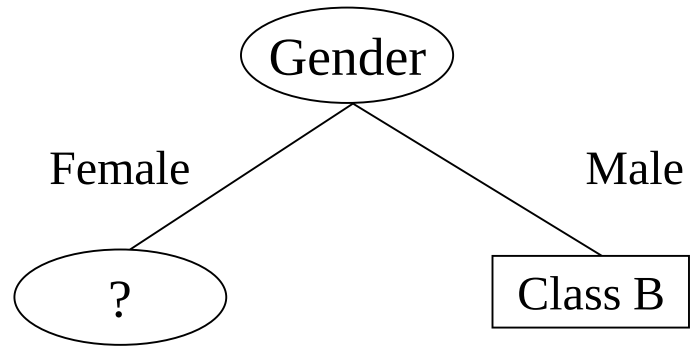
Attribute	GINI Index
OwnsHome	0.64
Married	0.40
Gender	0.343
Employed	0.475
CreditRating	0.64



- The attribute with the lowest GINI index is Gender. So the split attribute at this point is Gender.
- Now we can continue building the tree determining which of the remaining attributes we should split next, doing the same process.
- In fact, since all Males have already been classified (all 3 were in class B, and no Females were in class B), we don't need to consider these records again.

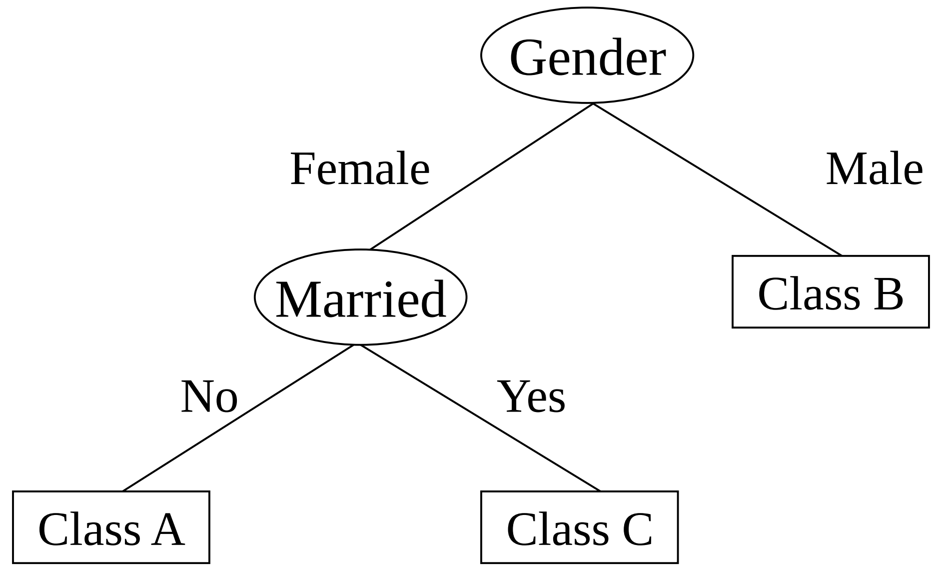
Example: splitting based on GINI

Attribute	GINI Index
OwnsHome	0.64
Married	0.40
Gender	0.343
Employed	0.475
CreditRating	0.64



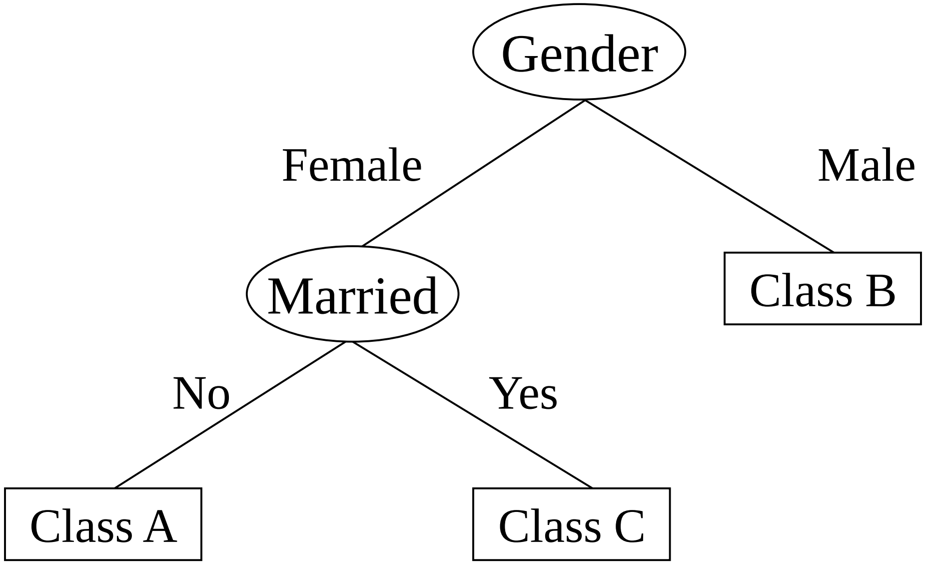
Owns home?	*Married*	*Gender*	*Employed*	*Credit rating*	*Risk class*
~~Yes~~	~~Yes~~	~~Male~~	~~Yes~~	~~*A*~~	~~*B*~~
No	No	Female	Yes	*A*	*A*
Yes	Yes	Female	Yes	*B*	*C*
~~Yes~~	~~No~~	~~Male~~	~~No~~	~~*B*~~	~~*B*~~
No	Yes	Female	Yes	*B*	*C*
No	No	Female	Yes	*B*	*A*
~~No~~	~~No~~	~~Male~~	~~No~~	~~*B*~~	~~*B*~~
Yes	No	Female	Yes	*A*	*A*
No	Yes	Female	Yes	*A*	*C*
Yes	Yes	Female	Yes	*A*	*C*

Example: final decision tree



Owns home?	*Married*	*Gender*	*Employed*	*Credit rating*	*Risk class*
~~Yes~~	~~Yes~~	~~Male~~	~~Yes~~	~~*A*~~	~~*B*~~
No	No	Female	Yes	*A*	*A*
Yes	Yes	Female	Yes	*B*	*C*
~~Yes~~	~~No~~	~~Male~~	~~No~~	~~*B*~~	~~*B*~~
No	Yes	Female	Yes	*B*	*C*
No	No	Female	Yes	*B*	*A*
~~No~~	~~No~~	~~Male~~	~~No~~	~~*B*~~	~~*B*~~
Yes	No	Female	Yes	*A*	*A*
No	Yes	Female	Yes	*A*	*C*
Yes	Yes	Female	Yes	*A*	*C*

Decision tree rules



- Decision trees produce clear 'if-then' rules.
- Rules can be used to query a relational database.
  - If Gender = 'Male' then Class = B
  - If Gender = 'Female' and Married = 'Yes' then Class = C, else Class = A.

Strengths and limitations of decision trees

Strengths:

- Can be easily interpreted
- Can handle both categorical and numerical variables
- Show the most important features in a dataset (the ones at the top that gave the best split)
- Are not sensitive to outliers or missing data

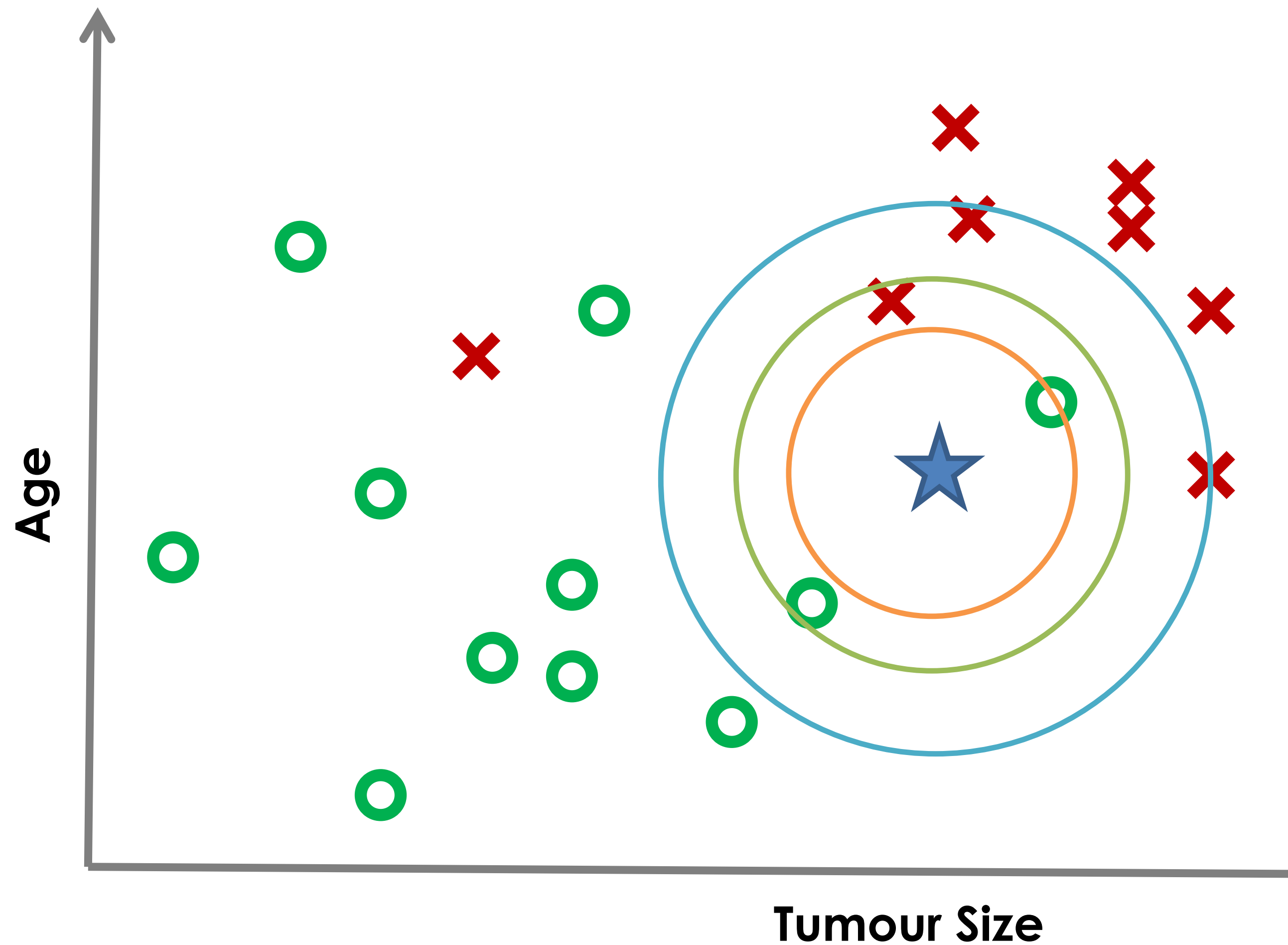
Limitations:

- Classifiers can create too complex trees that do not generalise well to new data (this is called overfitting).
- Sensitive to even small changes in the data

K-nearest neighbour (KNN)

- Object is assigned to the class most common among its ***K* nearest neighbours**
- Requires:
  – Distance Metric (e.g. Euclidean)
  – ***k*** parameter (no. of neighbours – nothing to do with K means clustering)
  – Weighting function
  – How to combine the info from neighbours

Classification with KNN



K= 1
Class?

K= 3
Class?

K= 5
Class?

Example:
- Metric = Euclidean
- No weighting function
- Maximum vote of neighbours

Summary

- Classification
- Decision Trees (algorithm and best split measures)
- KNN

Software implementations of decision tree and classification algorithms

- R *
- Weka *
- ScipY and scikit-learn python libraries *
- Orange *
- MATLAB
- SPSS
- SAS
- STATA
- SQL Server Analysis Services

* Free software/OS licence

Additional resources

Reading:

- Tan et al. 'Introduction to Data Mining'

http://www-users.cs.umn.edu/~kumar/dmbook/index.php

Chapter 4 - Classification

Free download

Watching:

- Andrew Ng's Stanford Machine Learning lectures.

Brilliant module – it covers everything in ML!