Data Science for Business
Supervised Segmentation

Asst. Prof. Teerapong Leelanupab (Ph.D.)
Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang (KMITL)

Week 6.1

"Supervised Segmentation" Example: Market Life Insurance

- We have a particular life insurance product we would like to sell
- We have a nice offer, but we incur a cost to target it
- How should we proceed?

"Supervised Segmentation" Example: Market Life Insurance

- Buy a large mailing list with demographic information

Age	Income
35	75K
68	83K
43	61K
71	56K
…	…

Example: Market Life Insurance

- Send a letter to some prospects in a mailing list
- Wait for a response …

predict

Age	Income	Response?
35	75K	no
68	83K	yes
43	61K	no
71	56K	yes
…	…	…

Recall: what might a predictive model look like?

- There are different sorts of model. Here are just two examples:

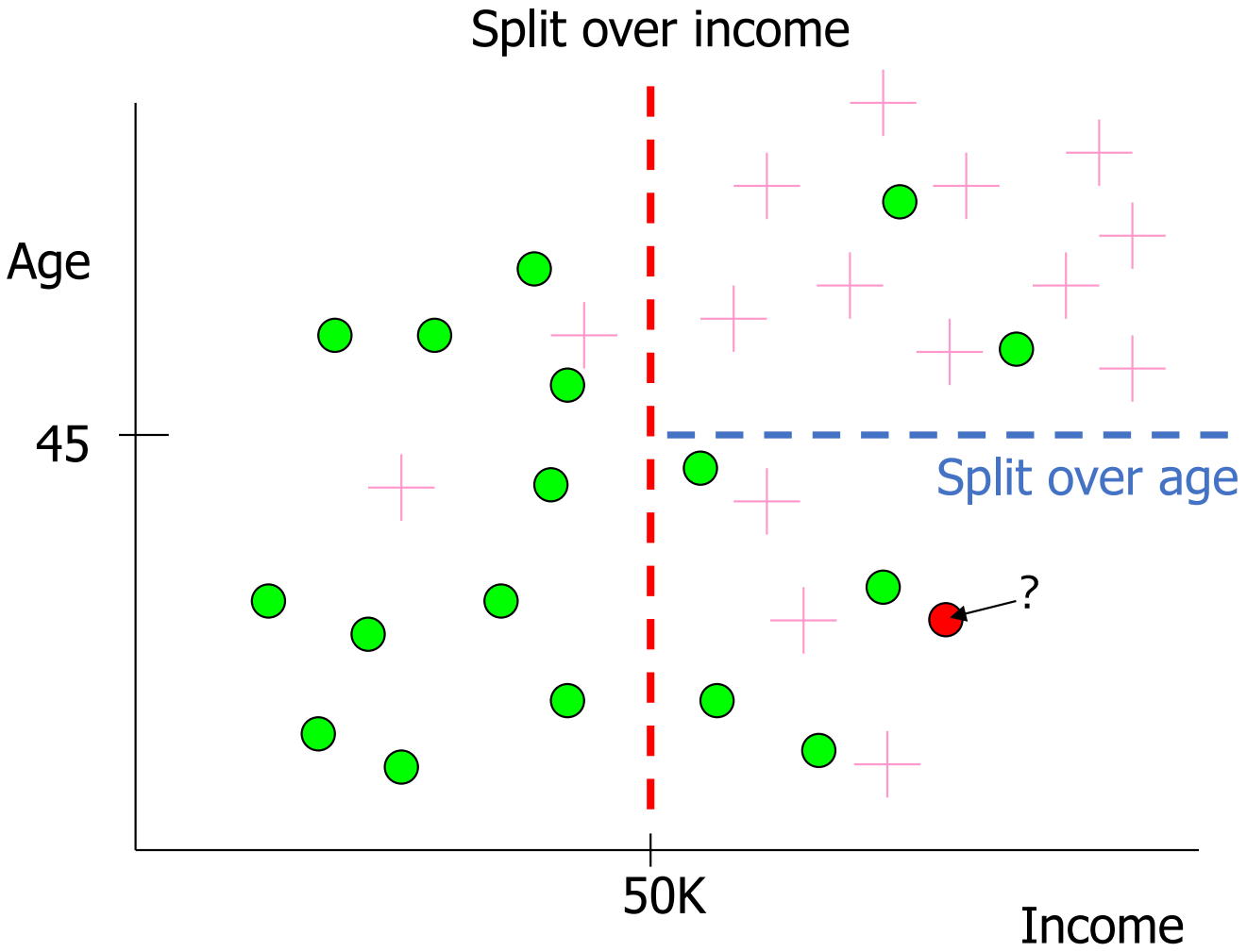
- Tree/Rule: (a supervised segmentation)
- **If (income > $50K) & (age > 45) then LI = YES**
- **If … then …**

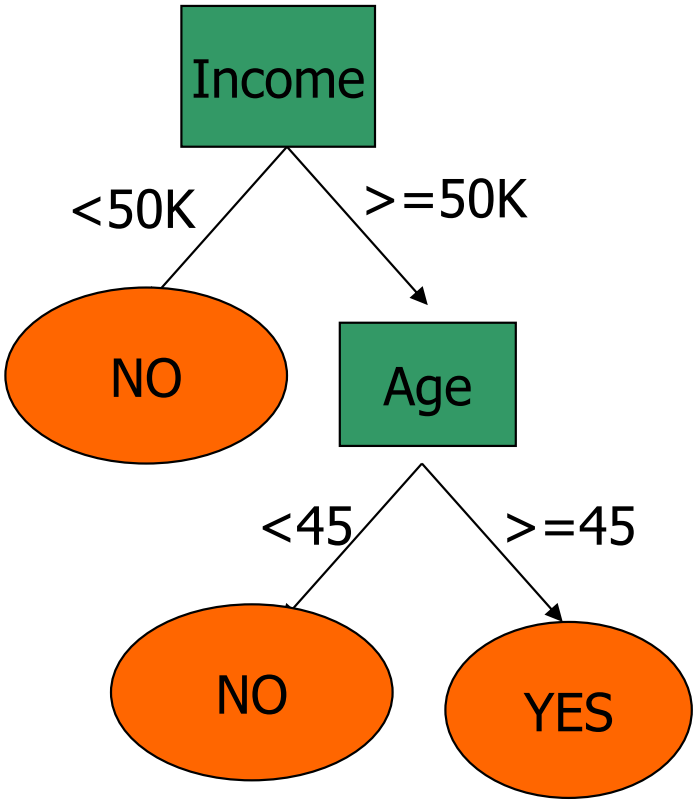
- Numeric function:
- **P(LI) = f(x_1,x_2,…, x_k)**

A supervised segmentation for targeting our Life Insurance product

can output the probability (not only label data)

Classification tree



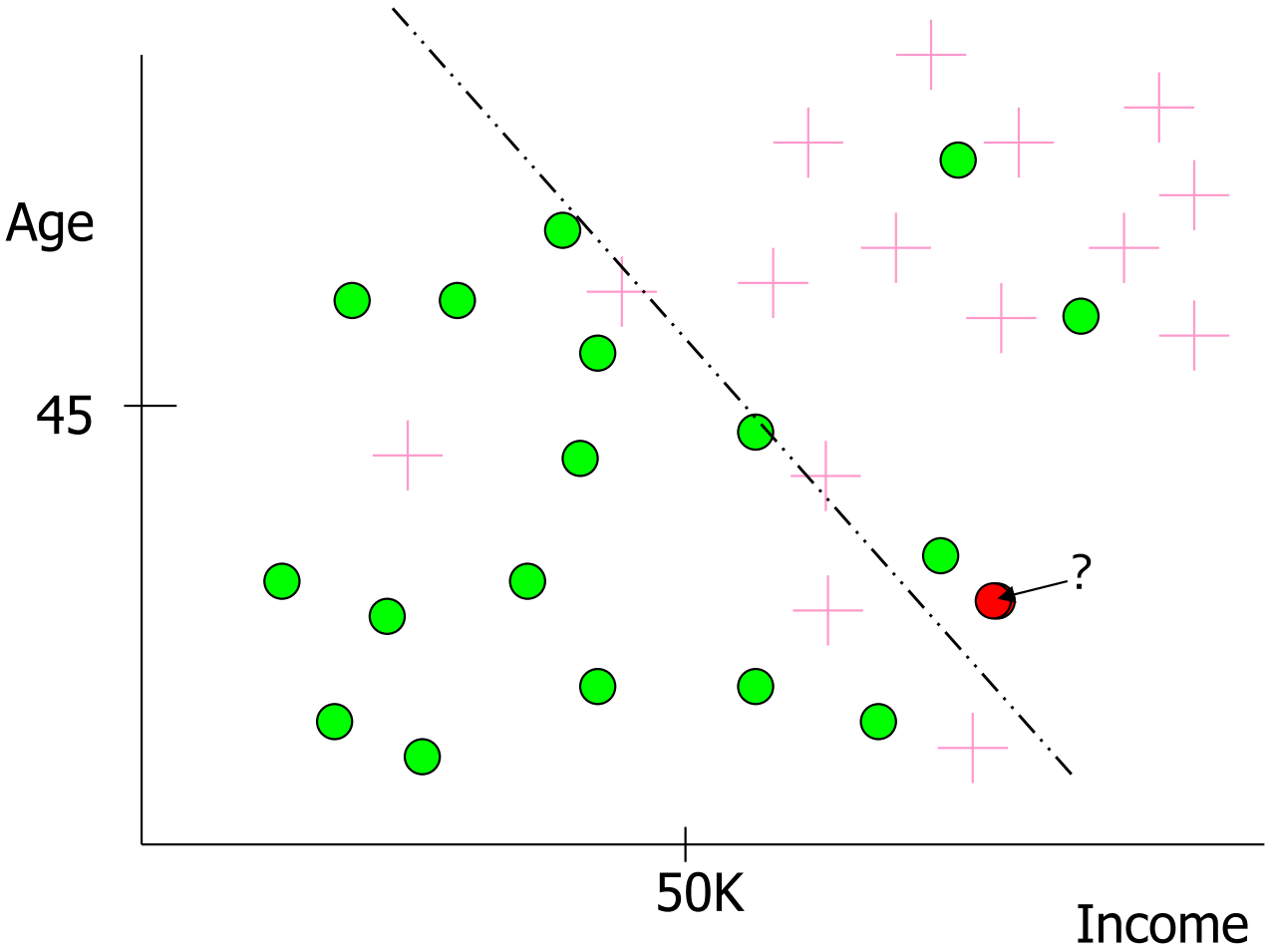


Did not buy life insurance

Bought life insurance

Interested in LI? NO

A different sort of supervised segmentation for our Life Insurance product



Logistic Regression

$$p(+|x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

$\beta_0 = 123$
$\beta_1 = -1.3$

p(LI|x) = 0.48

Credit Card Application – 16 cases

No Credit Card Application – 14 cases

Example: Life Insurance Marketing

Classification vs. Regression?

Think:

- What is the target variable?
- What values can it take in your data?

Age	**Income**	**Response?**
35	75K	no
68	83K	yes
43	61K	no
71	56K	yes
…	…	…

Caveat of classification?

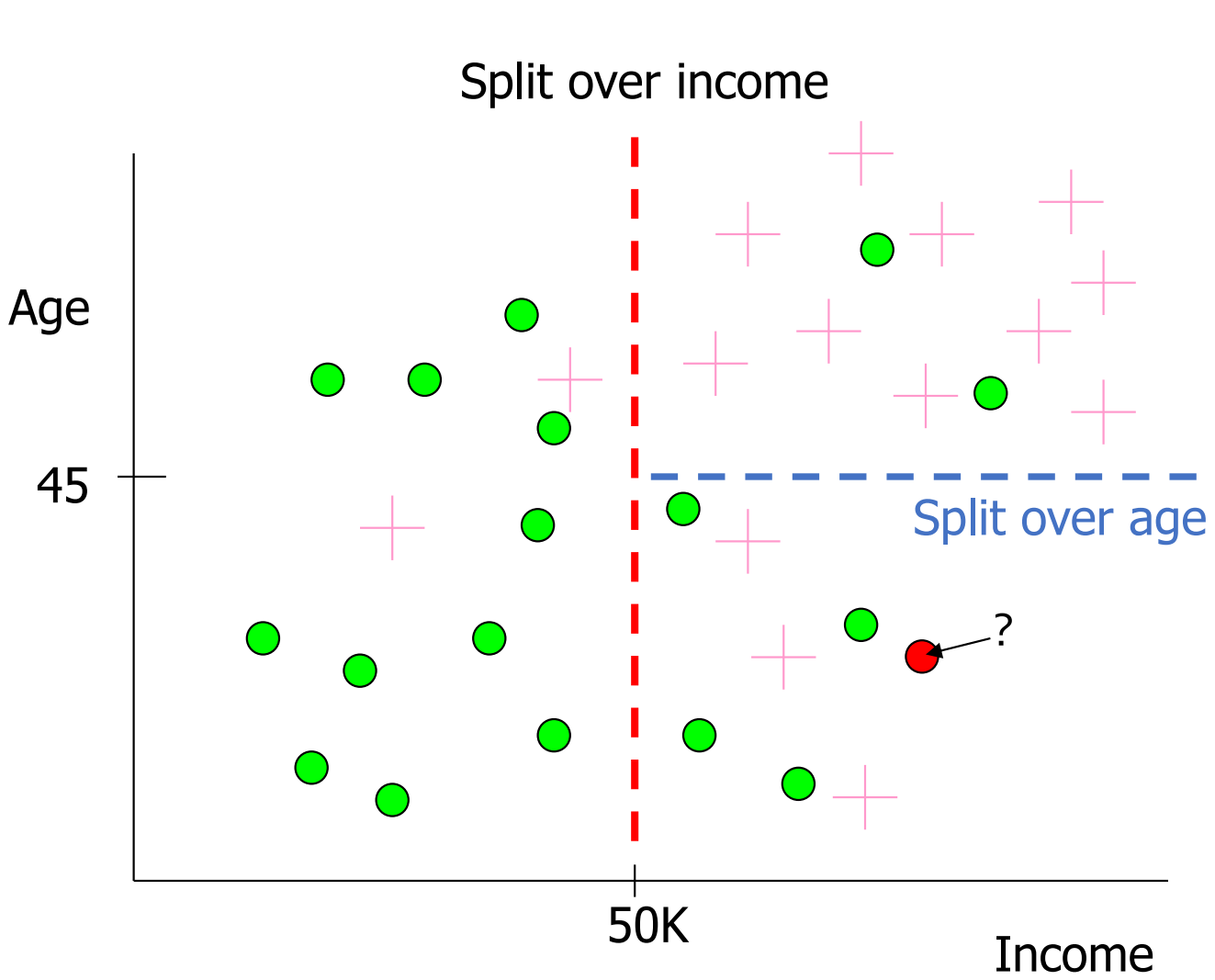
Type of target variable:

- classification → categorical target

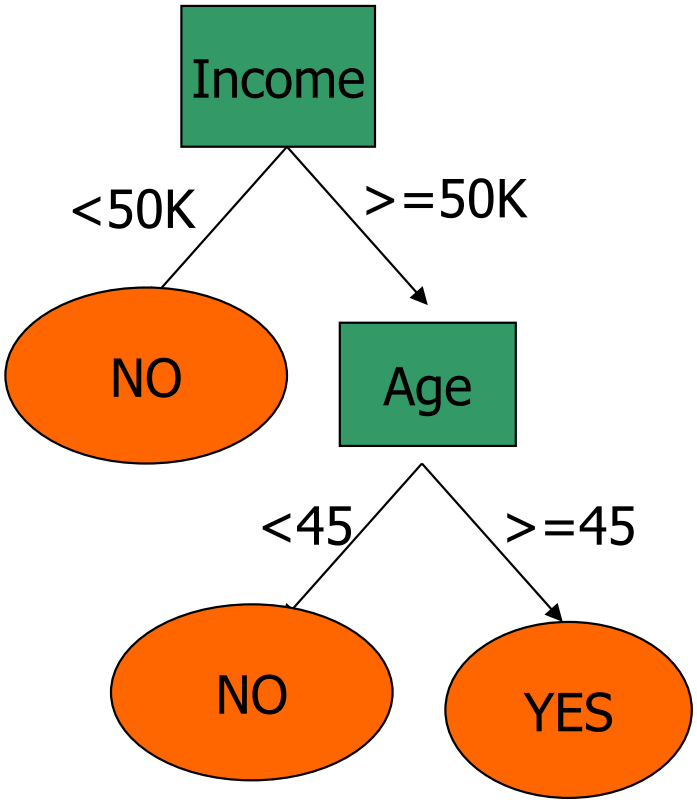
Many classification models can predict continuous values (probabilities, or "ranks"/"scores")

In that case classification can also be referred to as probability estimation or ranking

What are we predicting?



Classification tree



Did not buy life insurance

Bought life insurance

Interested in LI? NO

When might a probability be more useful than a yes/no?

- Life insurance targeting?

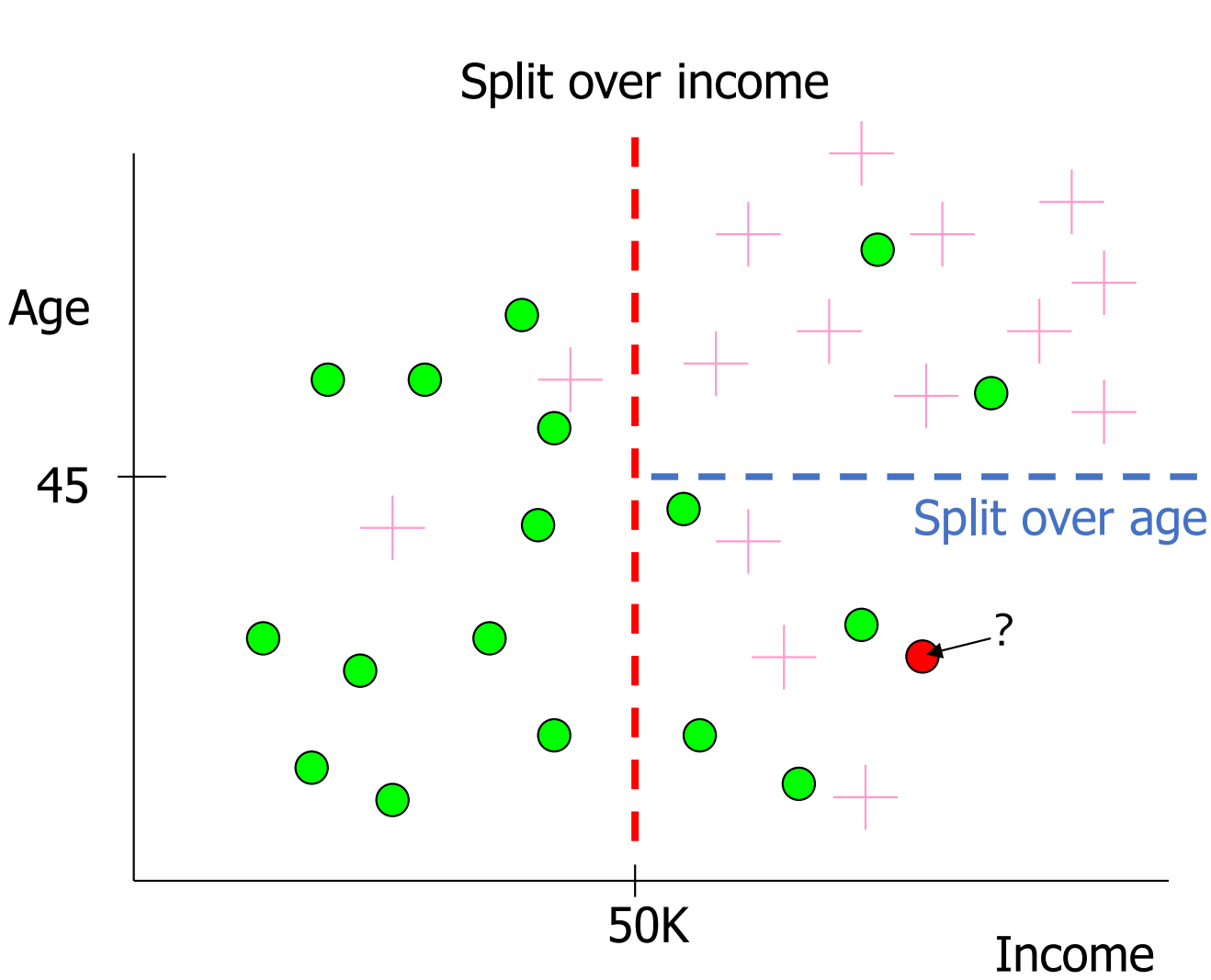
Limited offer so you would send offer to more trust customer for safe cost

- Default prediction?

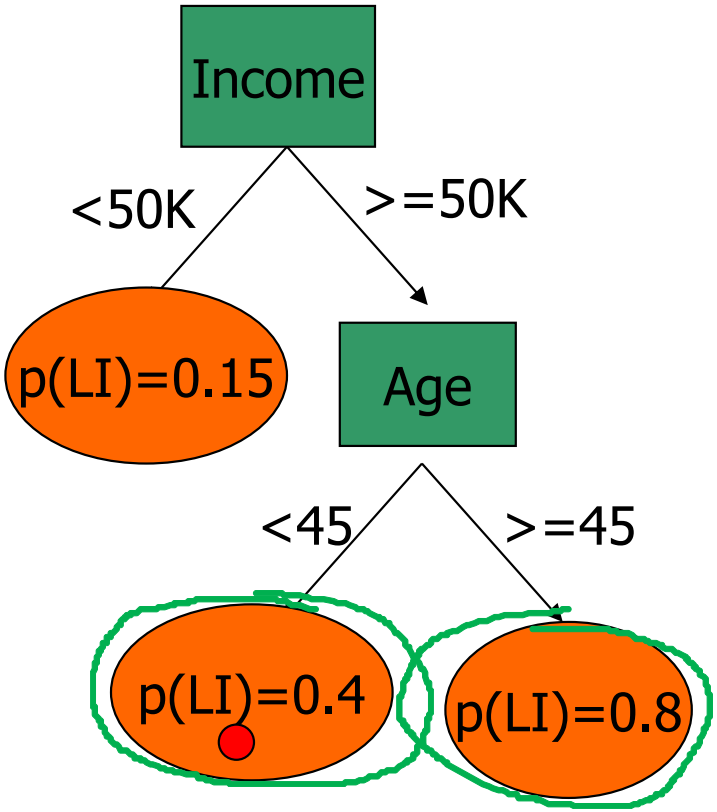
ทำนายการผิดชำระหนี้ (ลูกค้าคนไหนจะผิดชำระ)
ถ้ารู้ว่าคนนั้นๆ อาจจะผิดชำระก็ไม่ส่ง offer ไปให้

- ..... ?

What are we predicting?



Classification tree



ชัวร์คนไหนมากกว่า (prob สูงกว่า)

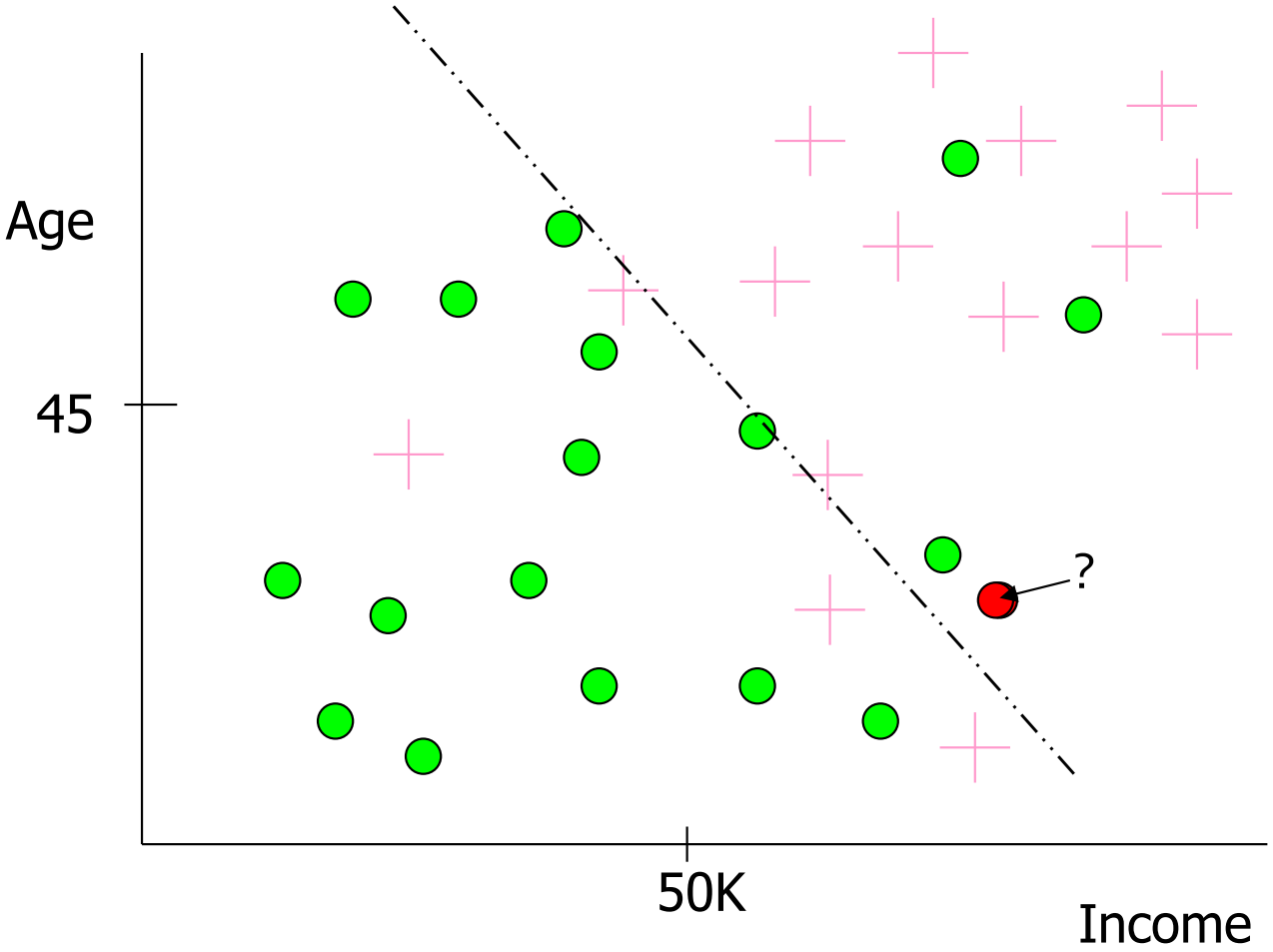
Did not buy life insurance

Bought life insurance

Interested in LI?=3/7

Classification, ranking, or probability estimation?

Logistic Regression



$$p(+|x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

$\beta_0 = 123$
$\beta_1 = -1.3$

p(LI|x) = 0.48

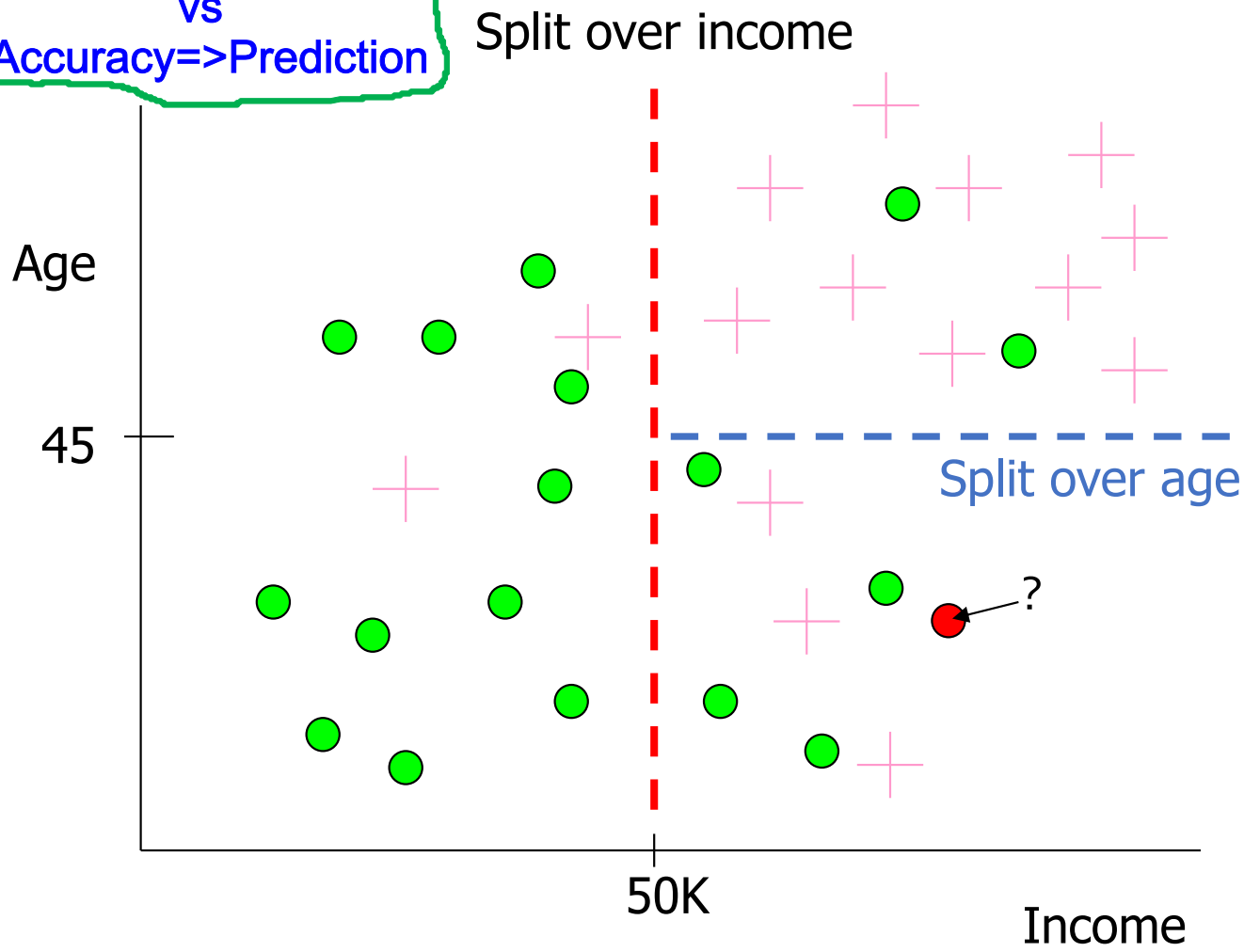
Credit Card Application – 16 cases

No Credit Card Application – 14 cases

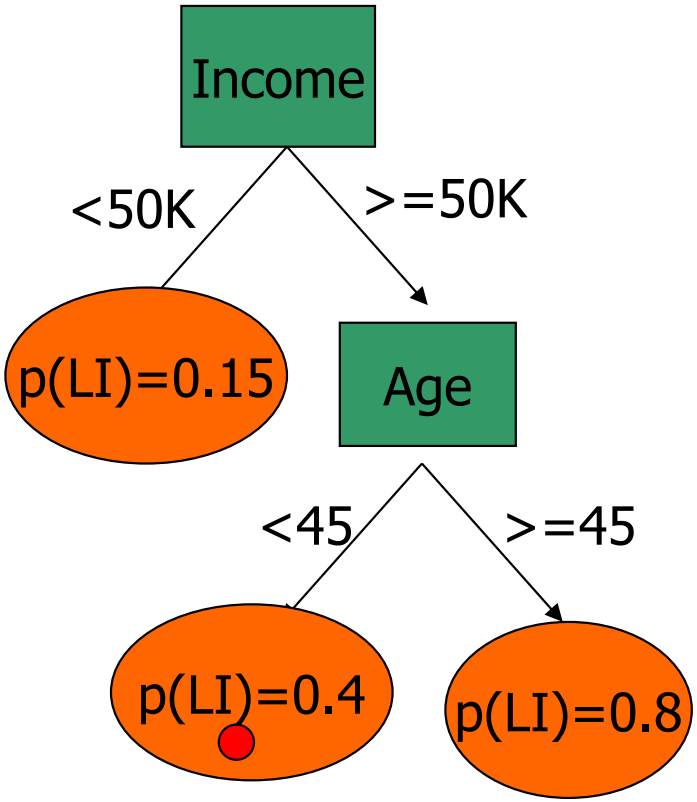
Which part is prediction?
Which part is understanding?

at USA Bank still use Decision Tree(not most accuracy) because they want to use advantage of decision tree (it can be explain)

Explain=>Understand
vs
Accuracy=>Prediction



Classification tree

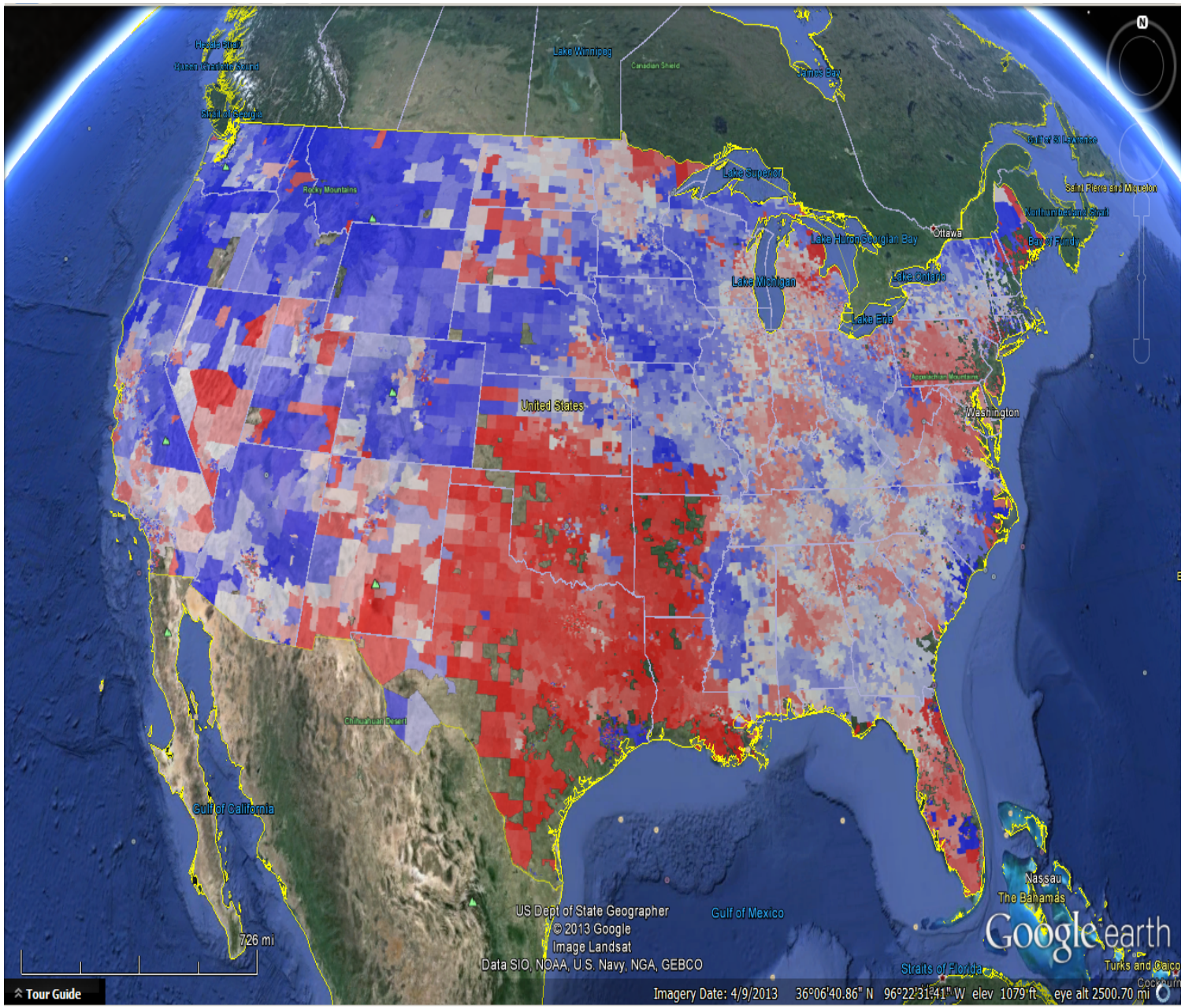


Did not buy life insurance

Bought life insurance

Interested in LI?=3/7

Heatmap of Westin Hotels geographic brand affinity



Data Science for Business
Decision Trees

Asst. Prof. Teerapong Leelanupab (Ph.D.)
Faculty of Information Technology
King Mongkut's Institute of Technology Ladkrabang (KMITL)

Week 6.2

Overview

- What is a Decision Tree?
- Feature Selection
  - Good v Bad features
  - Information Theory - Entropy
- How to build a Decision Tree?
  - ID3 top-down algorithm
  - Information Gain
- Decision Trees in **scikit-learn**

Decision Tree Learning

- Goal: Build a tree model that splits the training set into subsets in which all examples have the same class.
- A feature can be used to split the training set, one for each value or range of the feature…
  - e.g. insured = {true, false}
  - e.g. income = {low, average, high}
  - e.g. height < 6ft, height ≥ 6ft
- If necessary, each subset can be split again using another feature, and so on until all examples have the same class.
- Selecting a feature on which to split can be done using a measure based on uncertainty.
- Once the tree is built, we can use it to quickly classify new input examples (i.e. eager learning).

Example: Apples v Pears

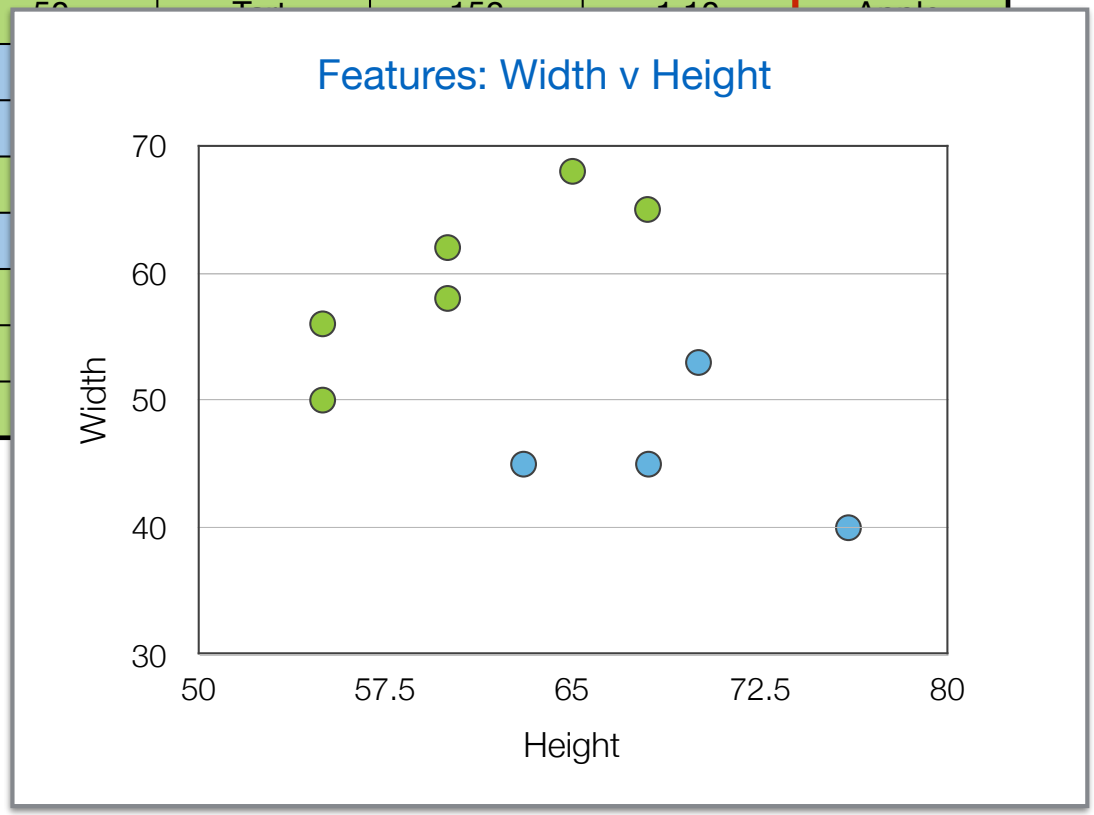
- 10 training examples such that: each has a class label (“apple” or “pear”), and each is described with 6 features.

Example	*Colour*	*Height*	*Width*	*Taste*	*Weight*	*H/W*	*Class*
1	210	60	62	Sweet	186	0.97	Apple
2	220	70	53	Sweet	180	1.32	Pear
3	215	55	50	Tart	152	1.10	Apple
4	180	76	40	Sweet	152	1.90	Pear
5	220	68	45	Sweet	153	1.51	Pear
6	160	65	68	Sour	221	0.96	Apple
7	215	63	45	Sweet	140	1.40	Pear
8	180	55	56	Sweet	154	0.98	Apple
9	220	68	65	Tart	221	1.05	Apple
10	190	60	58	Sour	175	1.03	Apple

Example: Apples v Pears

- 10 training examples such that: each has a class label (“apple” or “pear”), and each is described with 6 features.

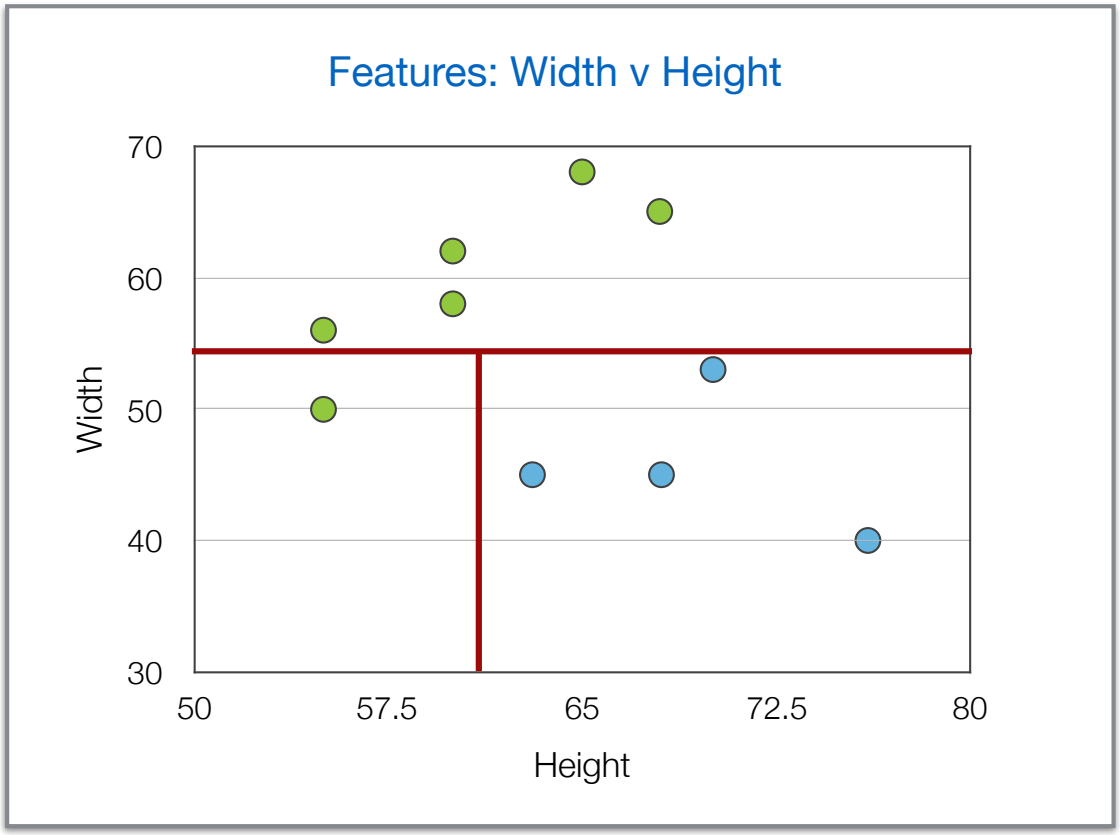
Example	Colour	Height	Width	Taste	Weight	H/W	Class
1	210	60	62	Sweet	186	0.97	Apple
2	220	70	53	Sweet	180	1.32	Pear
3	215	55					
4	180	76					
5	220	68					
6	160	65					
7	215	63					
8	180	55					
9	220	68					
10	190	60					

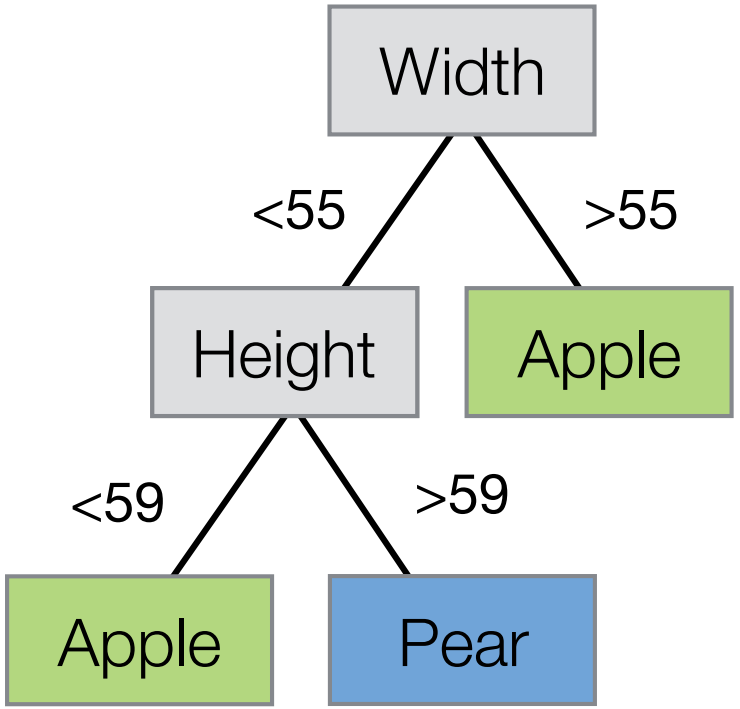


to see if their are some correlation

Example: Apples v Pears

- Simple decision tree for classifying Apples v Pears using only 2 features: {Height, Weight}



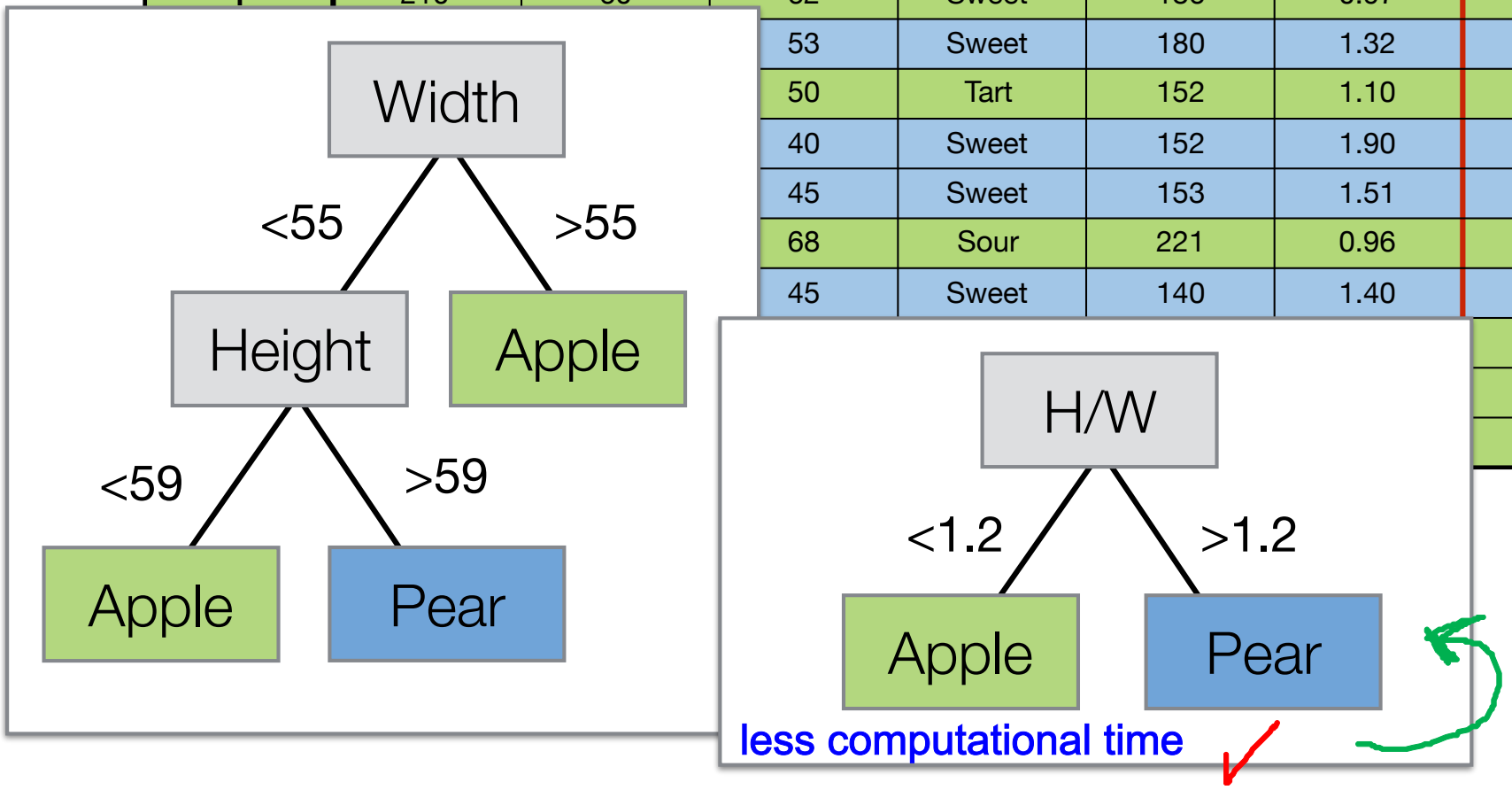


Just 2 features can split the data based on these decision rules.

Example: Apples v Pears

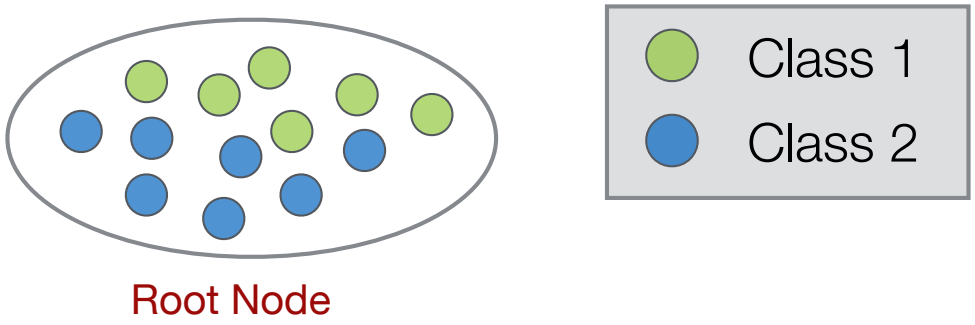
- 10 training examples such that: each has a class label (“apple” or “pear”), and each is described with 6 features.

Example	Colour	Height	Width	Taste	Weight	H/W	Class
1	210	60	62	Sweet	186	0.97	Apple
			53	Sweet	180	1.32	Pear
			50	Tart	152	1.10	Apple
			40	Sweet	152	1.90	Pear
			45	Sweet	153	1.51	Pear
			68	Sour	221	0.96	Apple
			45	Sweet	140	1.40	Pear
							Apple
							Apple
							Apple

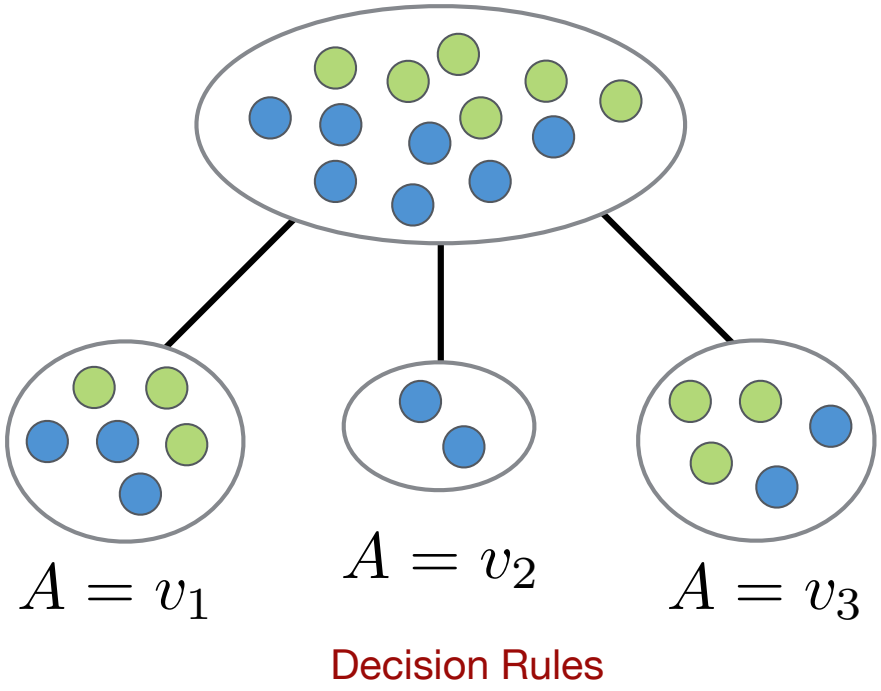


Decision Tree Learning

1. Initially all examples in the training set are placed at the root node of the tree.

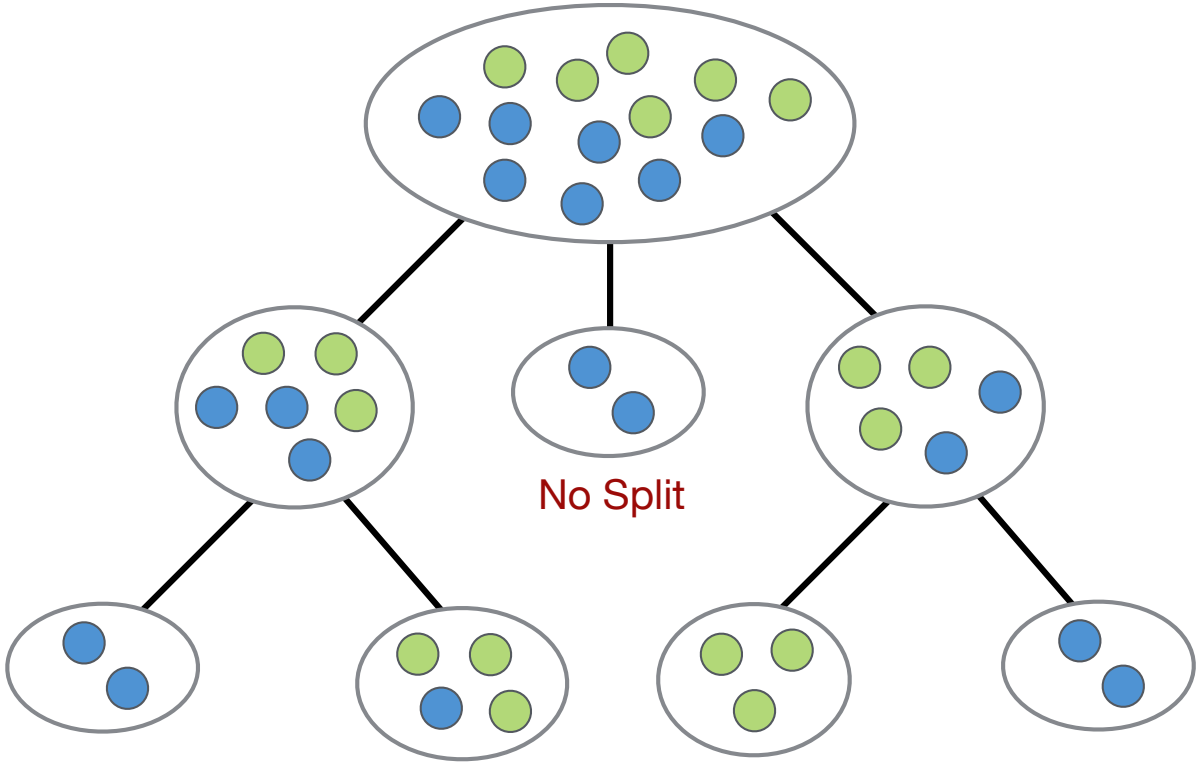


2. One of the available features (*A*) is now used to split the examples at the root node into two or more child nodes containing subsets of examples.



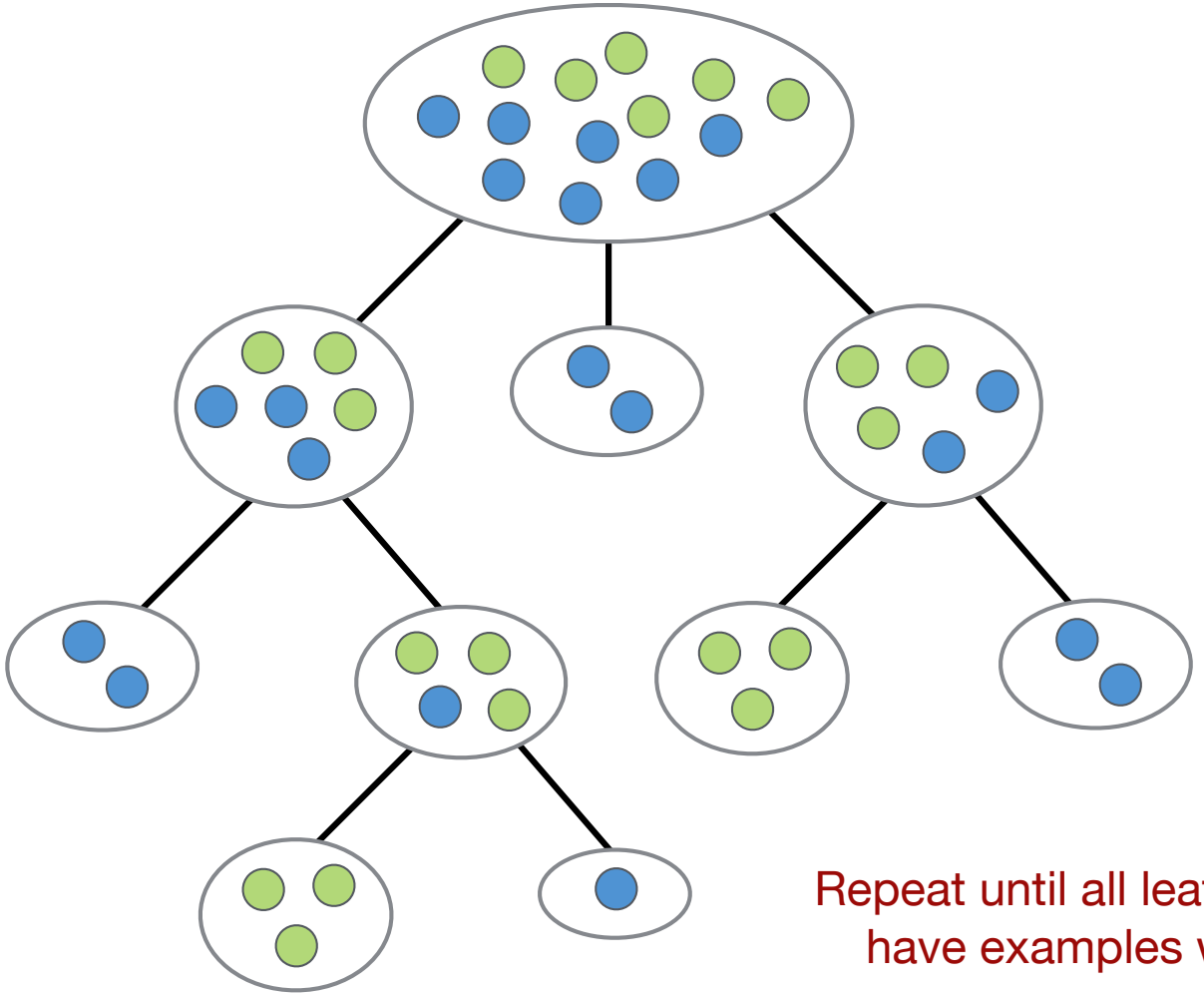
Decision Tree Learning

3. The same process is now applied to each child node, except for any child node at which all examples have the same class.



Decision Tree Learning

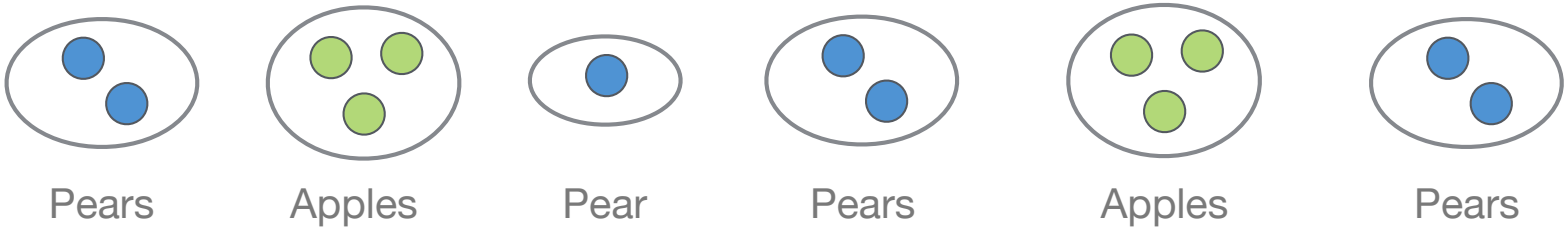
4. This continues until the training set has been divided into subsets in which all the examples have the same class.



Repeat until all leaf nodes in the tree have examples with same class

Node Purity

- A tree node is pure if all examples at that node have the same class label.



- A decision tree in which all the leaf nodes are pure can always be constructed provided there are no clashes in the data.
  - i.e. examples having the same “description” in terms of features, but with different class labels.
- Most decision tree algorithms use some measure of node (im)purity to choose features to split when building the tree. This measure guides the learning process.

Decision Trees Example

Q. "Will a customer wait for a restaurant table?"

Russell & Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 2009.

Binary classification task (WillWait = {Yes,No}), with examples described by 10 different features:

Feature	Description
Alternate	Is a suitable alternative restaurant nearby? (Yes/No)
Bar	Does the restaurant have a comfortable bar area to wait in? (Yes/No)
Fri/Sat	True on Fridays and Saturdays, False otherwise.
Hungry	Is the customer hungry? (Yes/No)
Patrons	How many people are in the restaurant: {None, Some, Full}?
Price	Restaurant's price range: {€, €€, €€€}
Raining	Is it raining outside? (Yes/No)
Reservation	Has the customer made a reservation? (Yes/No)
Type	Type of restaurant: {French, Italian, Thai, Burger}
WaitEstimate	Length of wait estimated by the host: {0-10, 10-30, 30-60, > 60 minutes}

Decision Trees Example

Q. "Will a customer wait for a restaurant table?"
Russell & Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 2009.

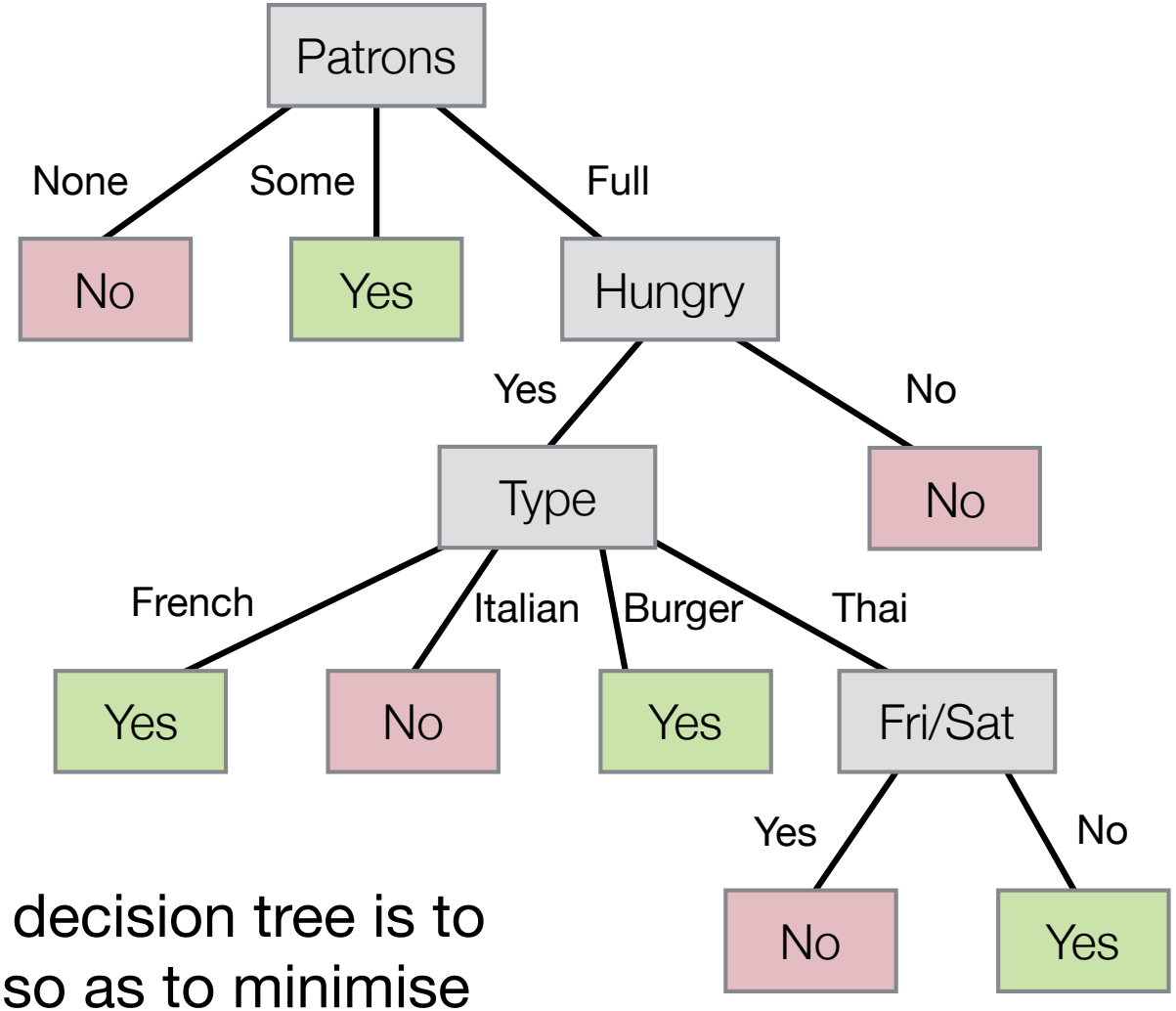
Binary classification task (WillWait = {Yes,No}), with examples described by 10 different features:

Example	Alternate	Bar	Fri/Sat	Hungry	Patrons	Price	Raining	Reservation	Type	WaitEst	WillWait?
1	Yes	No	No	Yes	Some	€€€	No	Yes	French	0-10	Yes
2	Yes	No	No	Yes	Full	€	No	No	Thai	30-60	No
3	No	Yes	No	No	Some	€	No	No	Burger	0-10	Yes
4	Yes	No	Yes	Yes	Full	€	No	No	Thai	10-30	Yes
5	Yes	No	Yes	No	Full	€€€	No	Yes	French	>60	No
6	No	Yes	No	Yes	Some	€€	Yes	Yes	Italian	0-10	Yes
7	No	Yes	No	No	None	€	Yes	No	Burger	0-10	No
8	No	No	No	Yes	Some	€€	Yes	Yes	Thai	0-10	Yes
9	No	Yes	Yes	No	Full	€	Yes	No	Burger	>60	No
10	Yes	Yes	Yes	Yes	Full	€€€	No	Yes	Italian	10-30	No
11	No	No	No	No	None	€	No	No	Thai	0-10	No
12	Yes	Yes	Yes	Yes	Full	€	No	No	Burger	30-60	Yes

➡ How do we build a "good" decision tree for this data set?

Decision Trees - Objective

A “good” decision tree will classify all examples correctly using as few tree nodes as possible.



Objective in building a decision tree is to choose good features so as to minimise the depth of the tree.

Supervised Segmentation Using Decision Tree

- How can we segment the population into groups that differ from each with respect to some quantity of interest?
- Informative attributes/features
    - Find **knowable** attributes/features that correlate with the target of interest
        - Increase accuracy
        - Alleviate computational problems
        - E.g., *tree induction*

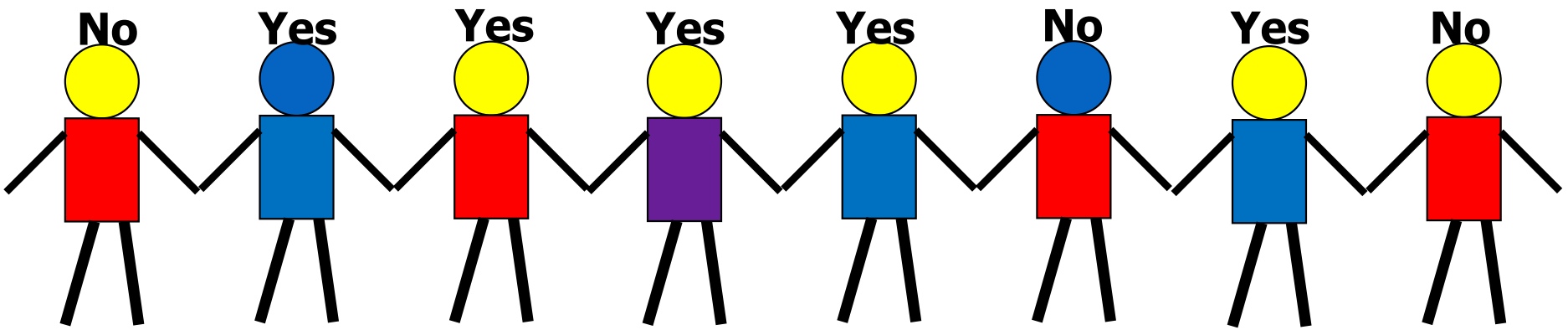
Informative Features in Tree Induction

- How can we judge whether a variable contains important information about the target variable?
    - How much?

Selecting Informative Features

Feature ที่ให้ข้อมูลเยอะๆ(ใช้ตัดได้ดี)

- Objective: Based on customer attributes/features, partition the customers into subgroups that are less impure – with respect to the class (i.e., such that in each group as many instances as possible belong to the same class)



Feature Selection

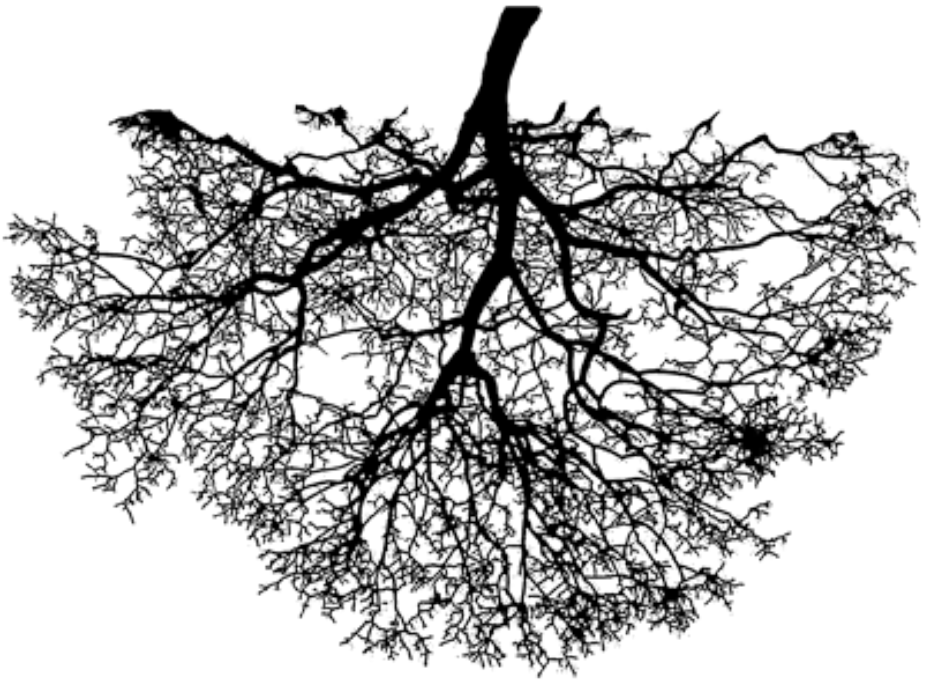
- Reasons for selecting only a subset of features:
- Better insights and business understanding
- Better explanations and more tractable models
- Reduced cost
- Faster predictions
- Better predictions!
    - Over-fitting (*to be continued*..)

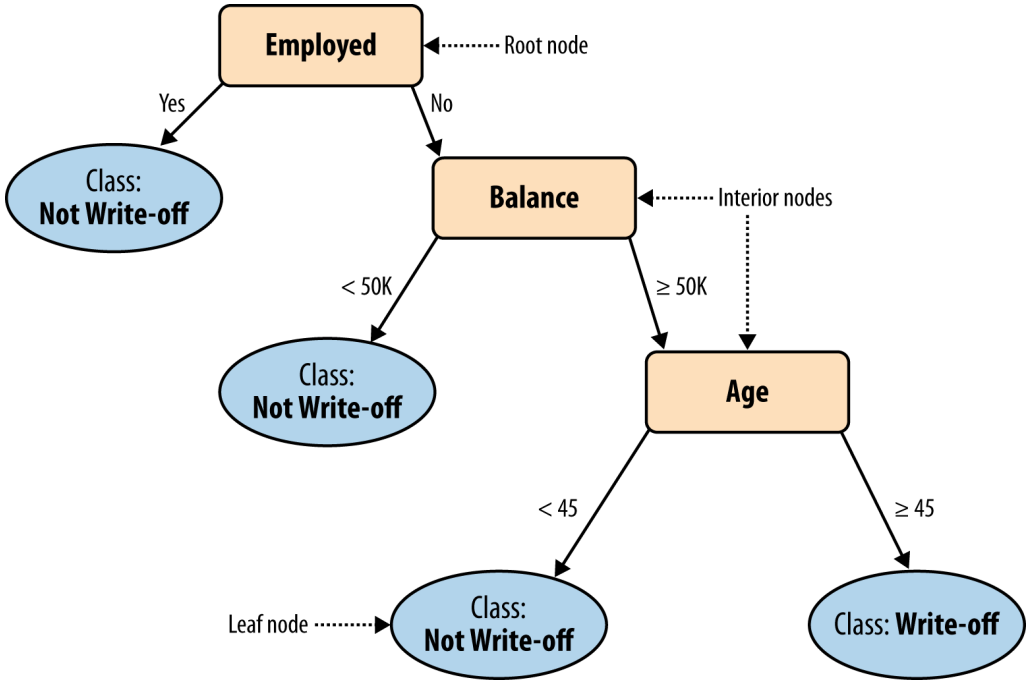
and also determining the most informative features..

Multivariate Supervised Segmentation

- If we select the *single* variable that gives the most information gain, we create a very *simple* segmentation
- If we select multiple attributes each giving some information gain, how do we put them together?

Tree-Structured Models





Tree-Structured Models: "Rules"

- No two parents share descendants
- There are no cycles
- The branches always "point downwards"
- Every example always ends up at a leaf node with some specific class determination
    - Probability estimation trees, regression trees (*to be continued*..)

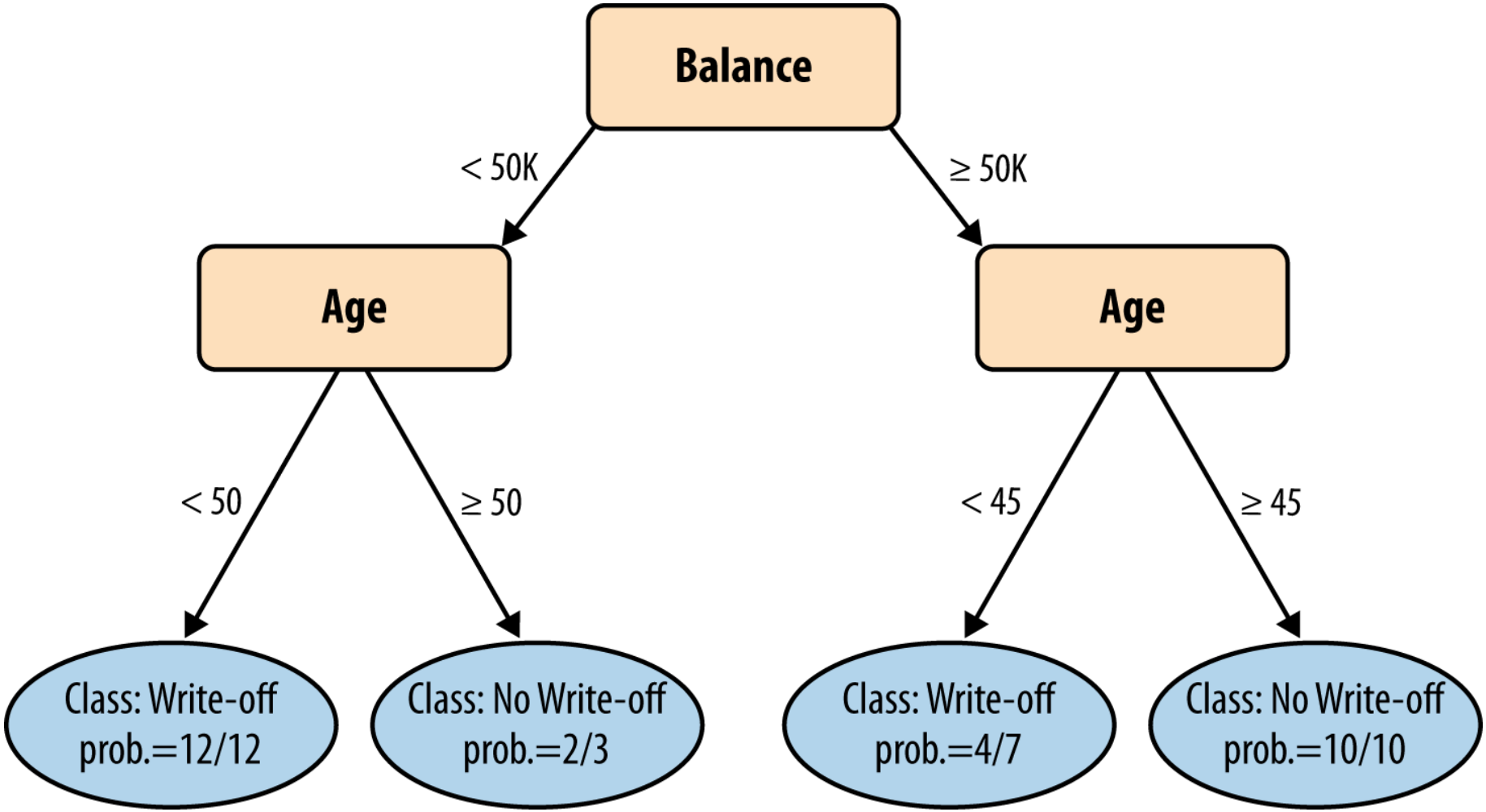
Tree Induction

- How do we create a classification tree from data?
  - **divide-and-conquer** approach
  - take each data subset and ***recursively*** apply attribute selection to find the best attribute to partition it
- When do we stop?
  - The nodes are pure,
  - there are no more variables, or
  - even earlier (over-fitting – *to be continued*..)

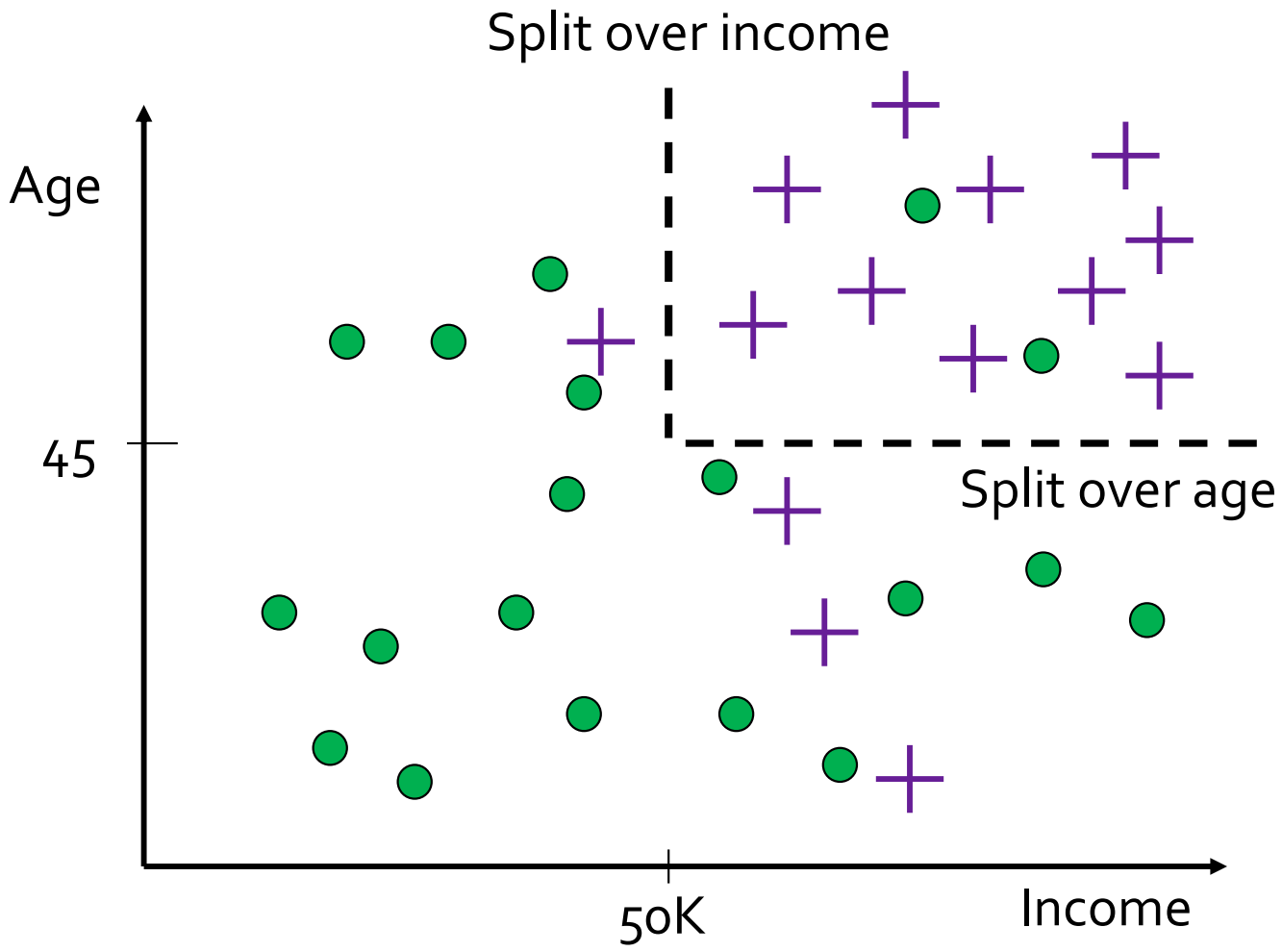
Why trees?

- Decision trees (DTs), or classification trees, are one of the most popular data mining tools
  - (along with linear and logistic regression)
- They are:
  - Easy to understand
  - Easy to implement
  - Easy to use
  - Computationally cheap
- Almost all data science packages include DTs
- They have advantages for model comprehensibility, which is important for:
  - model evaluation
  - communication to non-DM-savvy stakeholders

Visualizing Segmentations



Geometric interpretation of a model



Pattern:

IF Balance >= 50K & Age > 45

THEN Default = 'no'

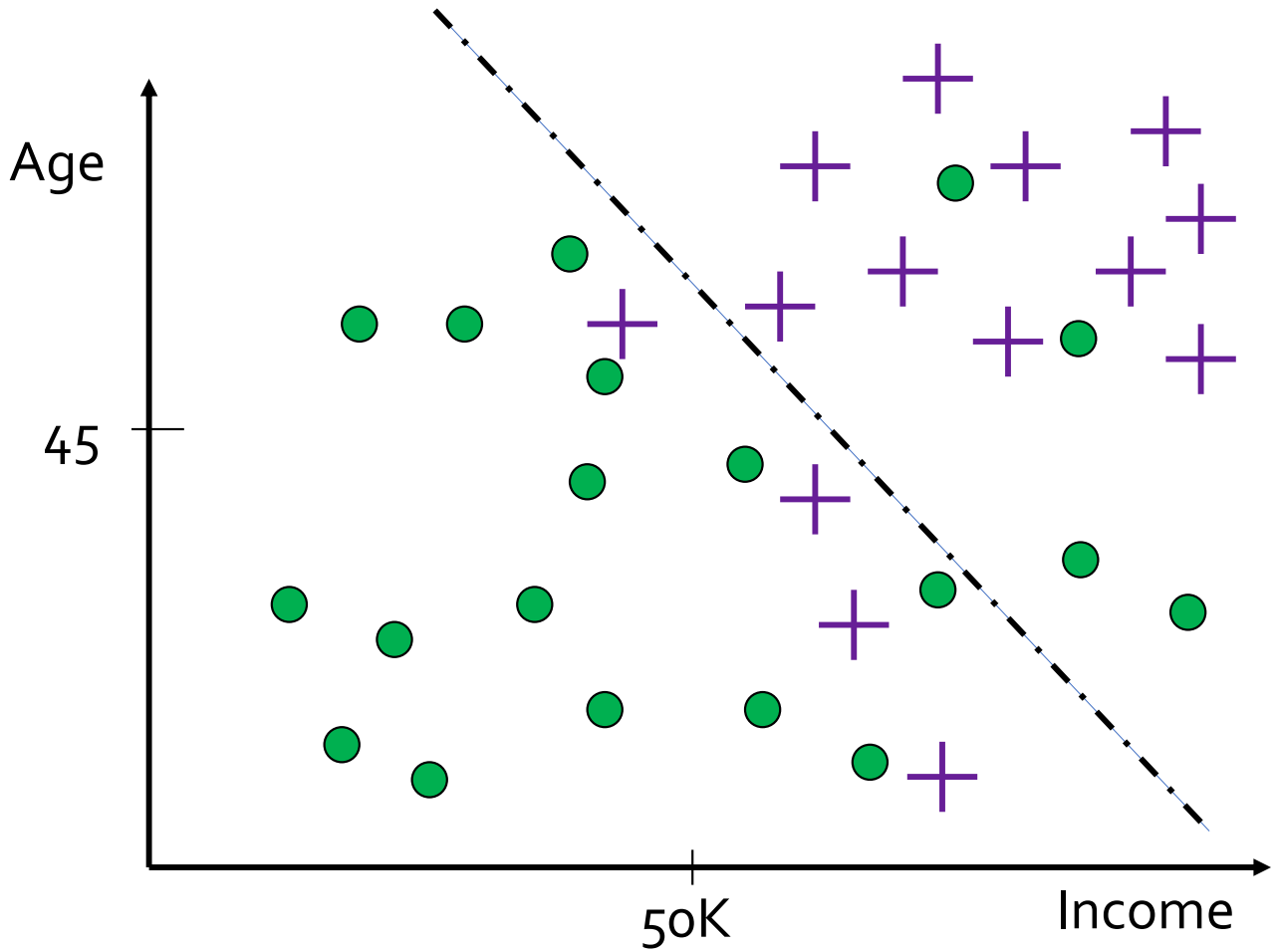
ELSE Default = 'yes'

Did not buy life insurance

Bought life insurance

Geometric interpretation of a model

What alternatives are there to partitioning this way?



"True" boundary may not be closely approximated by a linear boundary!

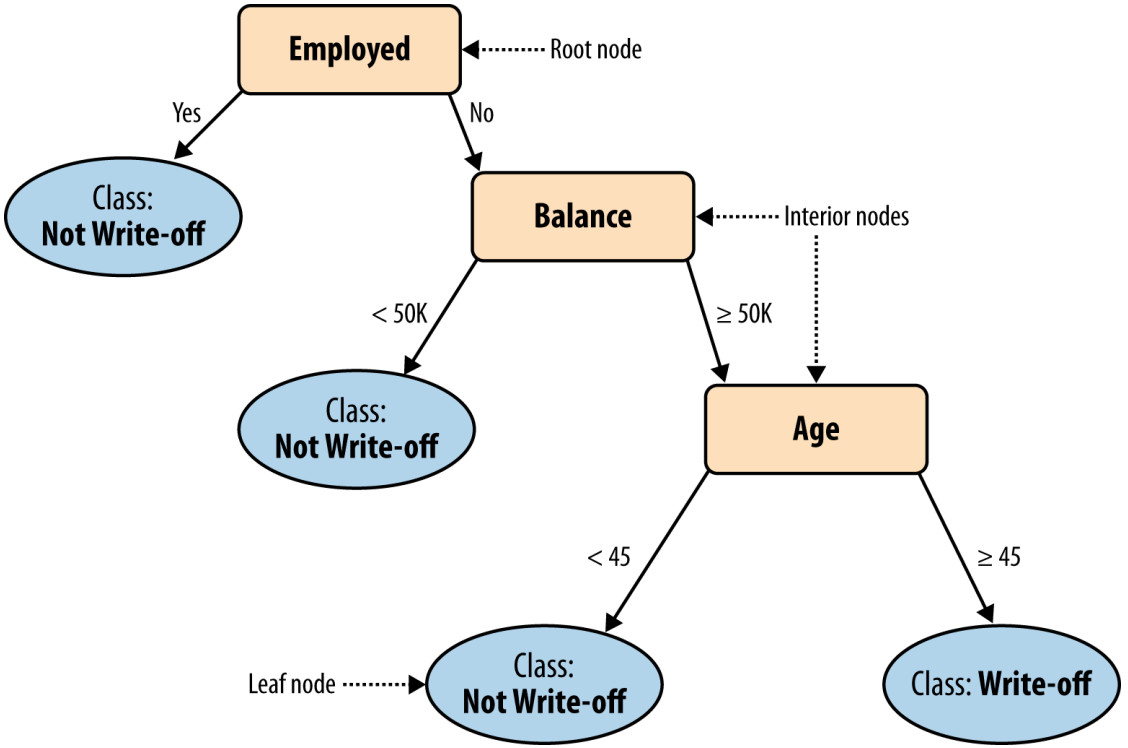
Did not buy life insurance

Bought life insurance

Trees as Sets of Rules

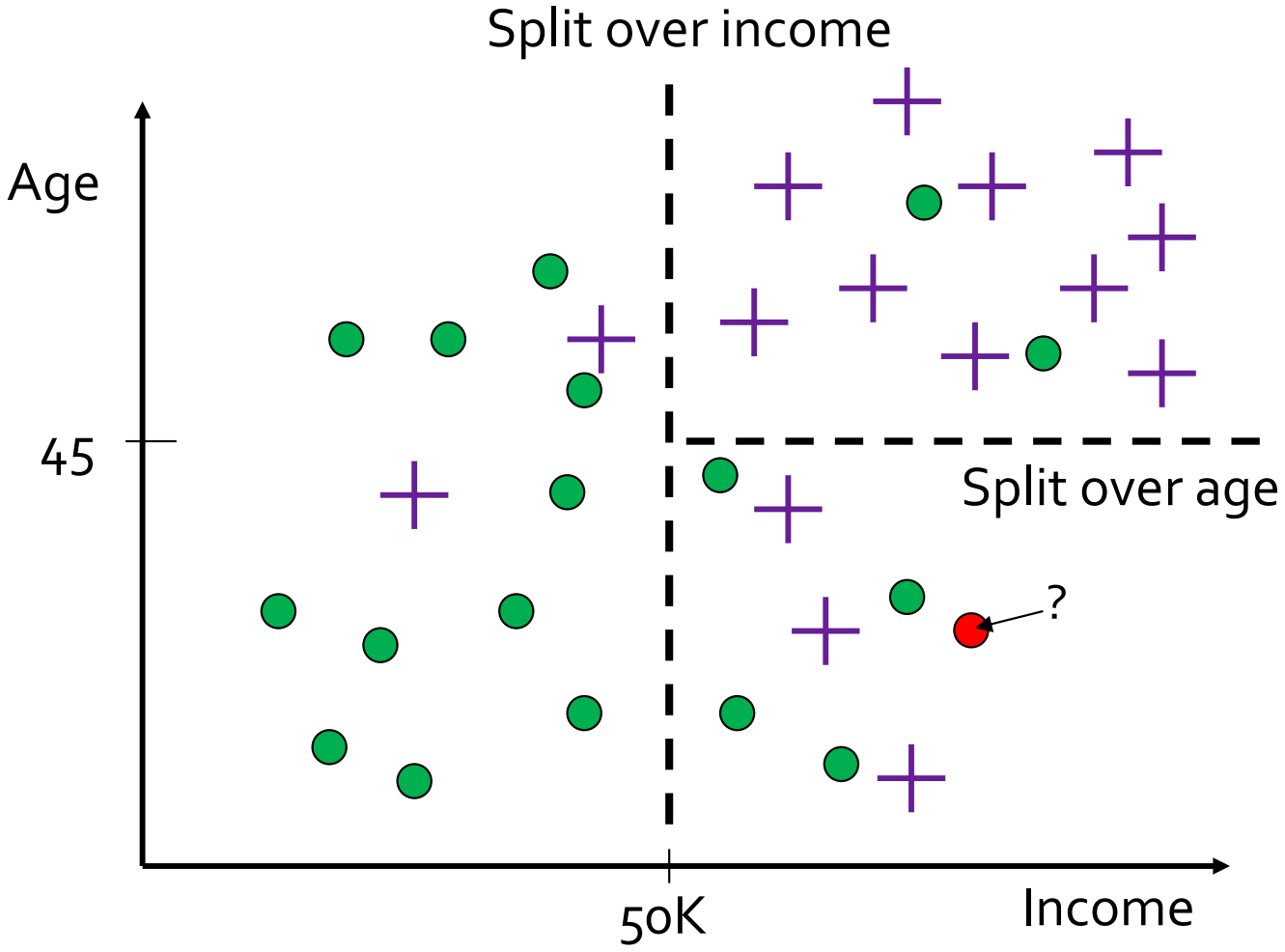
- The classification tree is equivalent to this rule set
- Each rule consists of the attribute tests along the path connected with **AND**

Trees as Sets of Rules

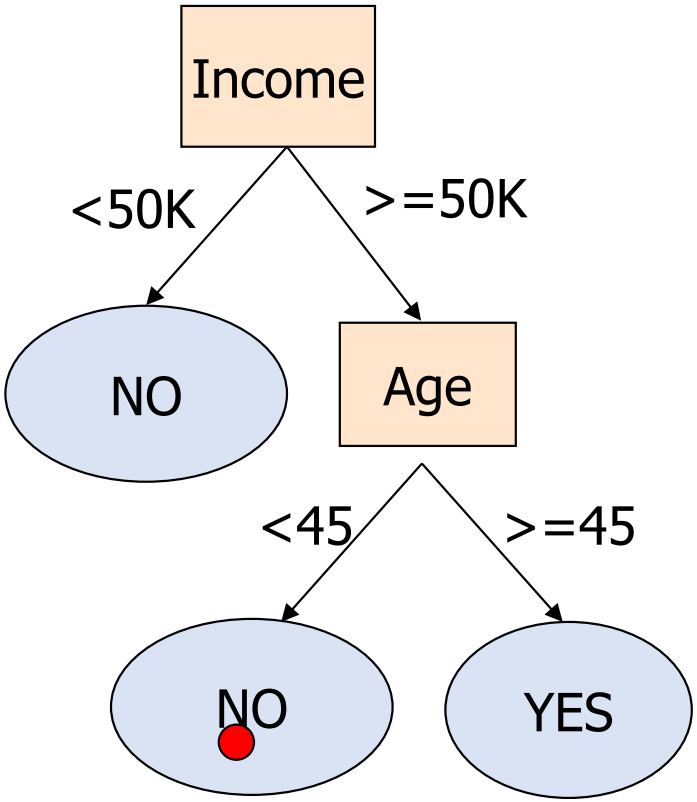


- IF (Employed = Yes) THEN Class=No Write-off
- IF (Employed = No) AND (Balance < 50k) THEN Class=No Write-off
- IF (Employed = No) AND (Balance ≥ 50k) AND (Age < 45) THEN Class=No Write-off
- IF (Employed = No) AND (Balance ≥ 50k) AND (Age ≥ 45) THEN Class=Write-off

What are we predicting?



Classification tree

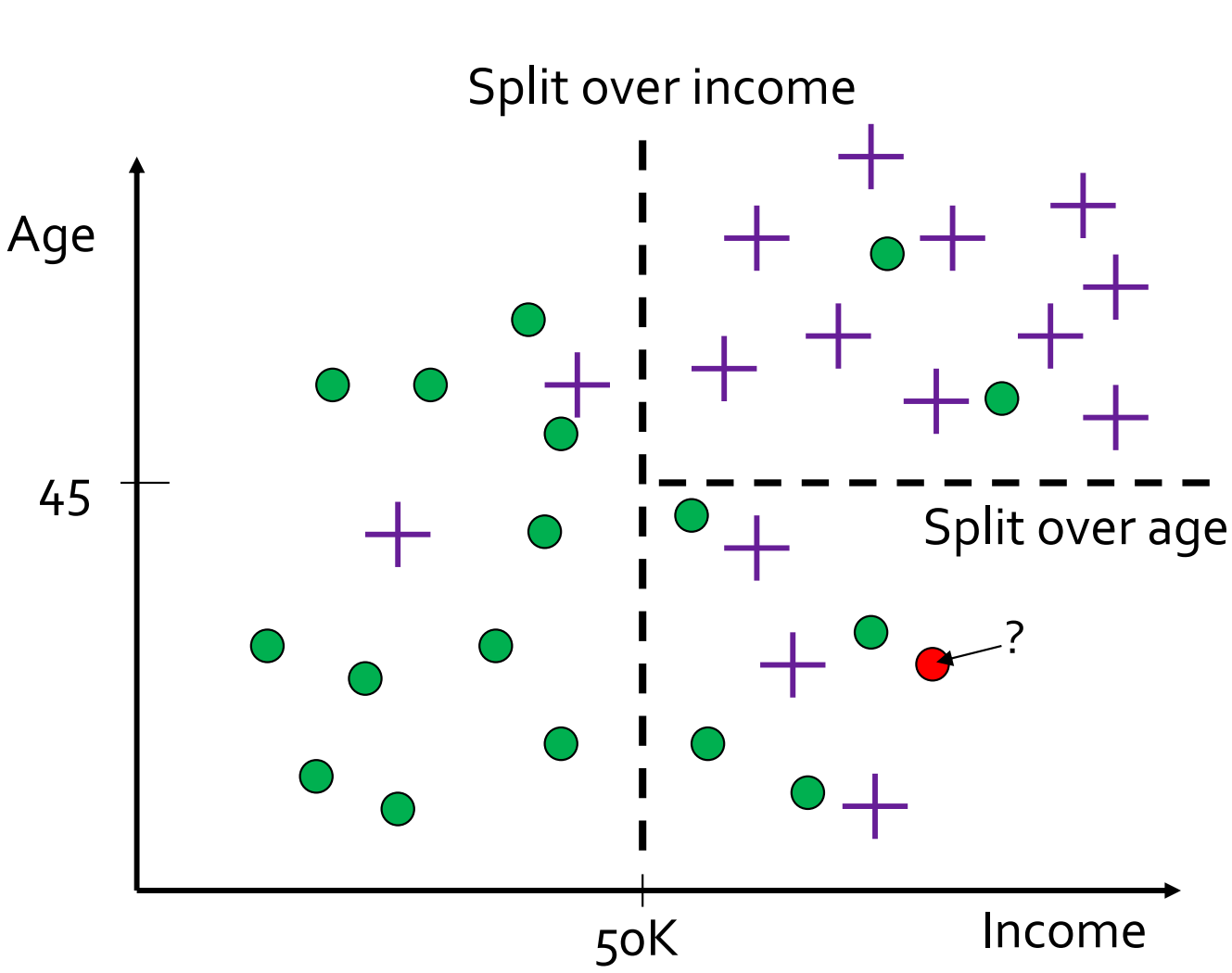


Did not buy life insurance

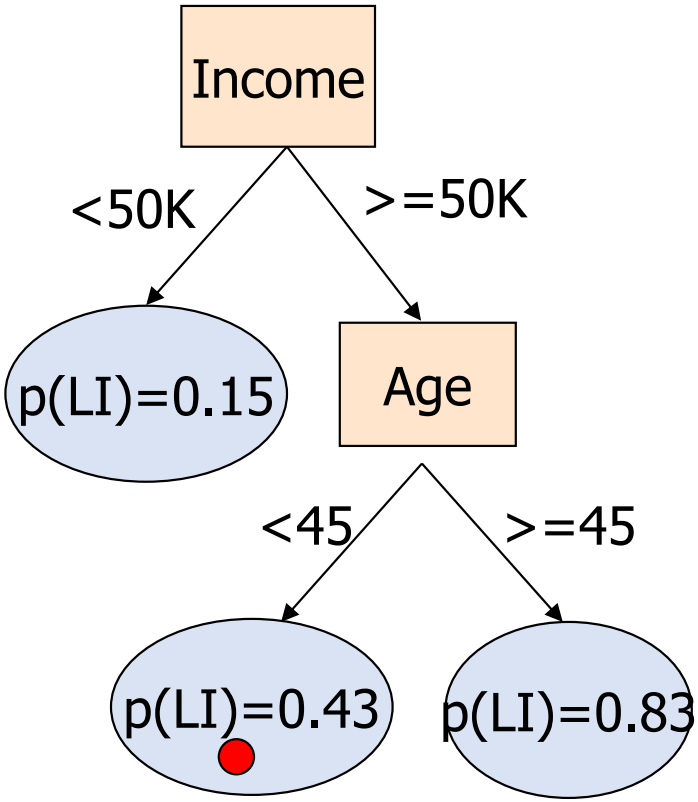
Bought life insurance

Interested in LI? = NO

What are we predicting?



Classification tree



Did not buy life insurance

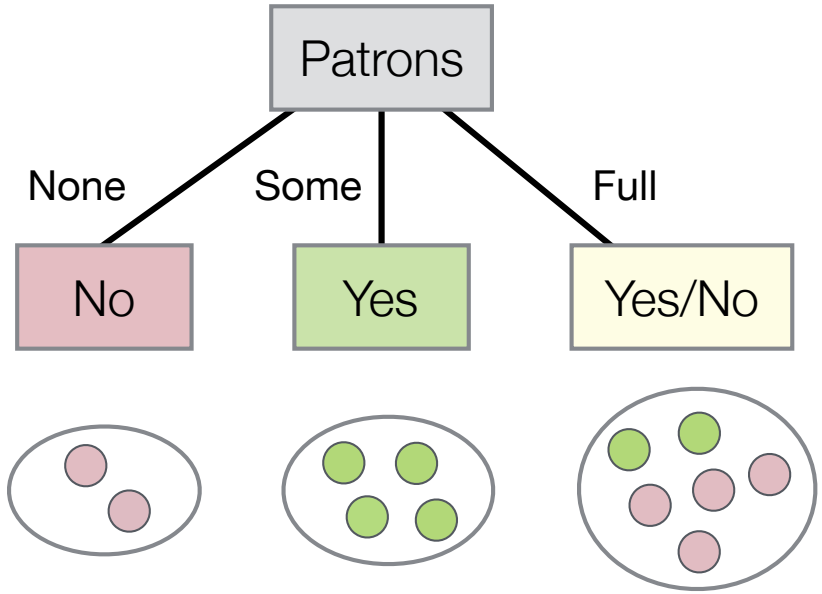
Bought life insurance

Interested in LI? = 3/7

Good v Bad Features

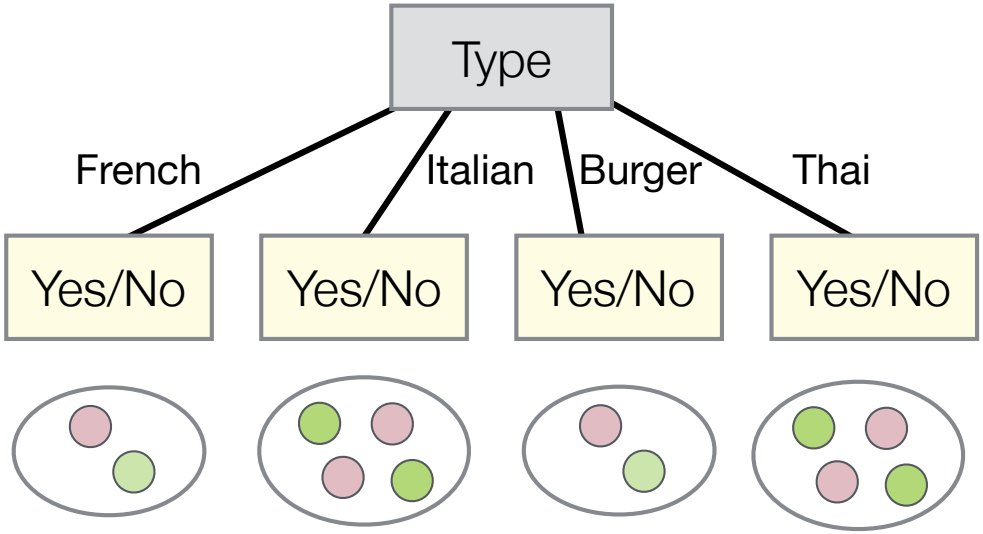
- Good Features:
  A perfect feature divides examples into categories of one class
  ⇒ high purity

  ดีมากใช้ตัดได้ดี



- Bad Features:
  A poor choice of feature produces categories of mixed classes
  ⇒ high impurity

  ไม่ดีเลย ใช้ตัดไม่ดี
  less informative



Feature Selection

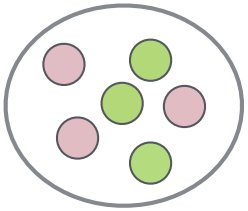
- Goal: Find good features which divide examples into categories of a single class.
- Feature selection algorithms have been developed which use impurity as an objective to guide feature selection (e.g. Entropy, Gini impurity).

- **Common selection strategy in decision trees:**
  - For each feature, some measure of impurity is applied to the current set of tree nodes.
  - The feature that maximises the reduction in impurity is selected as the next most useful feature.

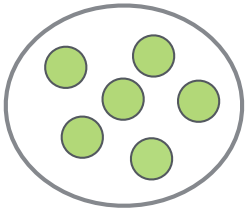
Entropy

- Entropy: In information theory, a measure of uncertainty around a source of information. Low for predictable sources, higher for more random sources.

- In the context of decision trees, entropy provides a measure of impurity - how uncertain we are about the decision for a given set of examples.



High uncertainty
→ High entropy



Low uncertainty
→ Low entropy

- **Definition:**

Entropy of a set of examples S with class labels $\{C_1, \ldots, C_n\}$:

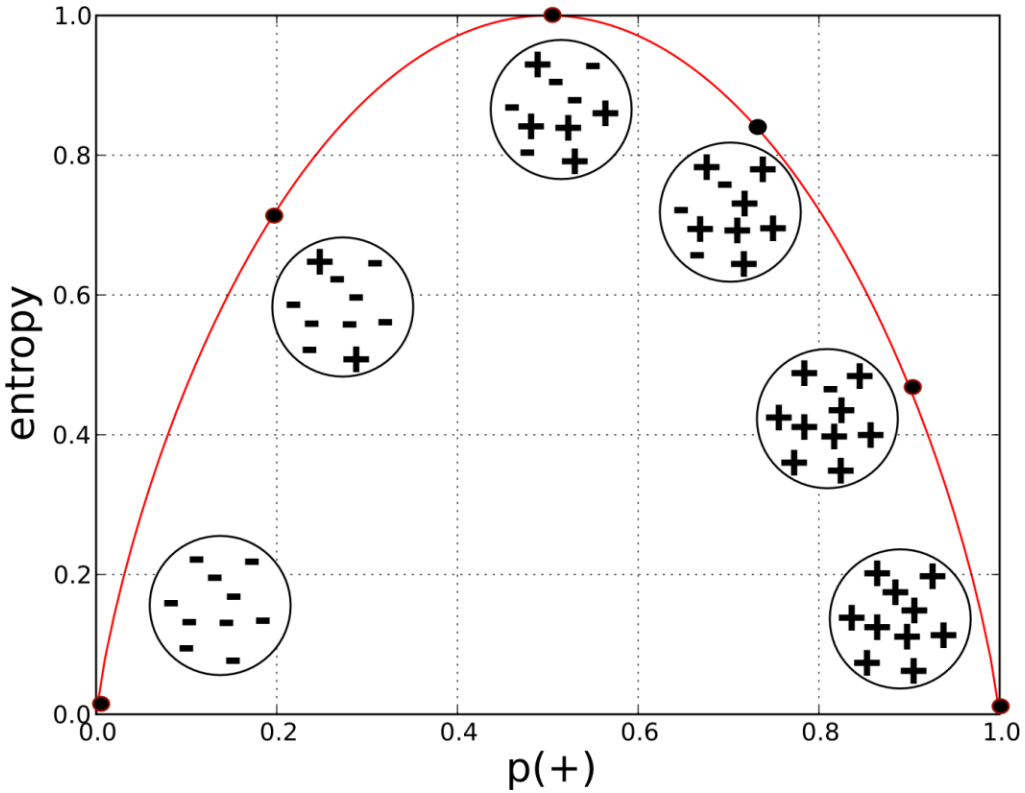
$$H(S) = -\sum_{i=1}^{n} p_i \log_2 p_i$$

3/6 * log_2v3/6

where p_i is the relative frequency (probability) of class C_i.

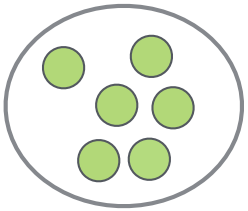
Entropy

- The **Impurity Measure** that estimates the general disorder/uncertainty of a set. (The higher entropy is, the lower the purity in the set of examples *S* is.)



Entropy Examples

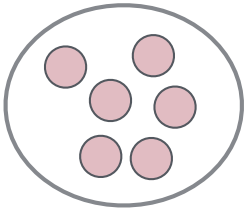
- The lowest possible entropy (i.e. zero) occurs when all examples have the same class label.
- The highest entropy occurs when we are most uncertain.



$$p_1 = 6/6 = 1.0 \qquad p_2 = 0/6 = 0.0$$

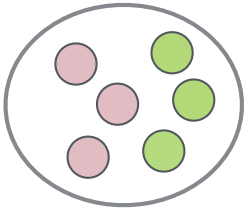
NB: Define $\log_2(0)=0$

$$H(S) = -((1 \times \log_2(1)) + (0 \times \log_2(0))) = -(0 + 0) = 0$$



$$p_1 = 0/6 = 0.0 \qquad p_2 = 6/6 = 1.0$$

$$H(S) = -((0 \times \log_2(0)) + (1 \times \log_2(1))) = -(0 + 0) = 0$$



$$p_1 = 3/6 = 0.5 \qquad p_2 = 3/6 = 0.5$$

$$H(S) = -((0.5 \times \log_2(0.5)) + (0.5 \times \log_2(0.5))) = -(-0.5 - 0.5) = 1$$

Top-Down Induction of Decision Trees

- ID3 Algorithm: Popular algorithm which repeatedly builds a decision tree from the top down (Quinlan, 1986).
- Start with an empty tree and set of all training examples S.

ID3(S):

- IF all examples in S belong to the same class C THEN
  - Return new leaf node and label it with class C.
- ELSE
  - Select a feature A based on some feature selection criterion.
  - Generate a new tree node with A as the test feature.
  - FOR EACH value v_i of A:
    - Let $S_i \subset S$ contain all examples with $A = v_i$.
    - Build subtree by applying ID3(S_i)

node ที่ไม่มี node อื่นต่อ

Criterion: Information Gain

- Information Gain (IG): Popular information theoretic approach for selecting features in decision trees, based on entropy.
- Measures a feature's overall impact on entropy when used to split a set of training examples into two or more subsets.
  - How much information do we learn by splitting on the feature?
  - How much is the reduction in entropy?

- **Definition:**

IG for feature A that splits a set of examples S into $\{S_1, \ldots, S_m\}$:

$$IG(S, A) = (\text{original entropy}) - (\text{entropy after split})$$

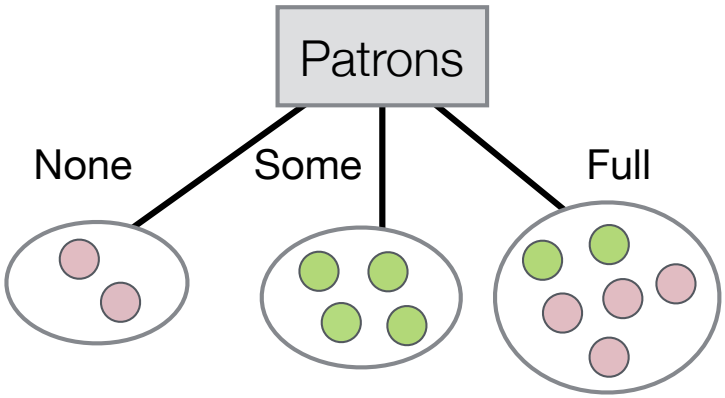
$$IG(S, A) = H(S) - \sum_{i=1}^{m} \frac{|S_i|}{|S|} H(S_i)$$

Each subset is weighted in proportion to its size

ค.ไม่แน่นอนก่อนตัด - ค.ไม่แน่นอนหลังตัด

Information Gain Example

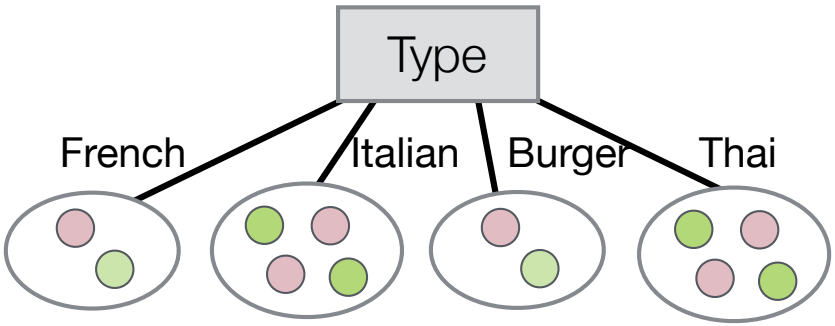
- Previous example: Initial training set has 6 Yes, 6 No examples.
- Which feature should we select to split at the root node?



original after split

$$IG = H([\frac{6}{12}, \frac{6}{12}]) - \left(\frac{2}{12}H([0,1]) + \frac{4}{12}H([1,0]) + \frac{6}{12}H([\frac{2}{6}, \frac{4}{6}])\right)$$

$$IG(\text{Patrons}) = 1 - 0.459 = 0.541$$ ✓



$$IG = H([\frac{6}{12}, \frac{6}{12}]) - \left(\frac{2}{12}H([\frac{1}{2}, \frac{1}{2}]) + \frac{4}{12}H([\frac{2}{4}, \frac{2}{4}]) + \frac{2}{12}H([\frac{1}{2}, \frac{1}{2}]) + \frac{4}{12}H([\frac{2}{4}, \frac{2}{4}])\right)$$

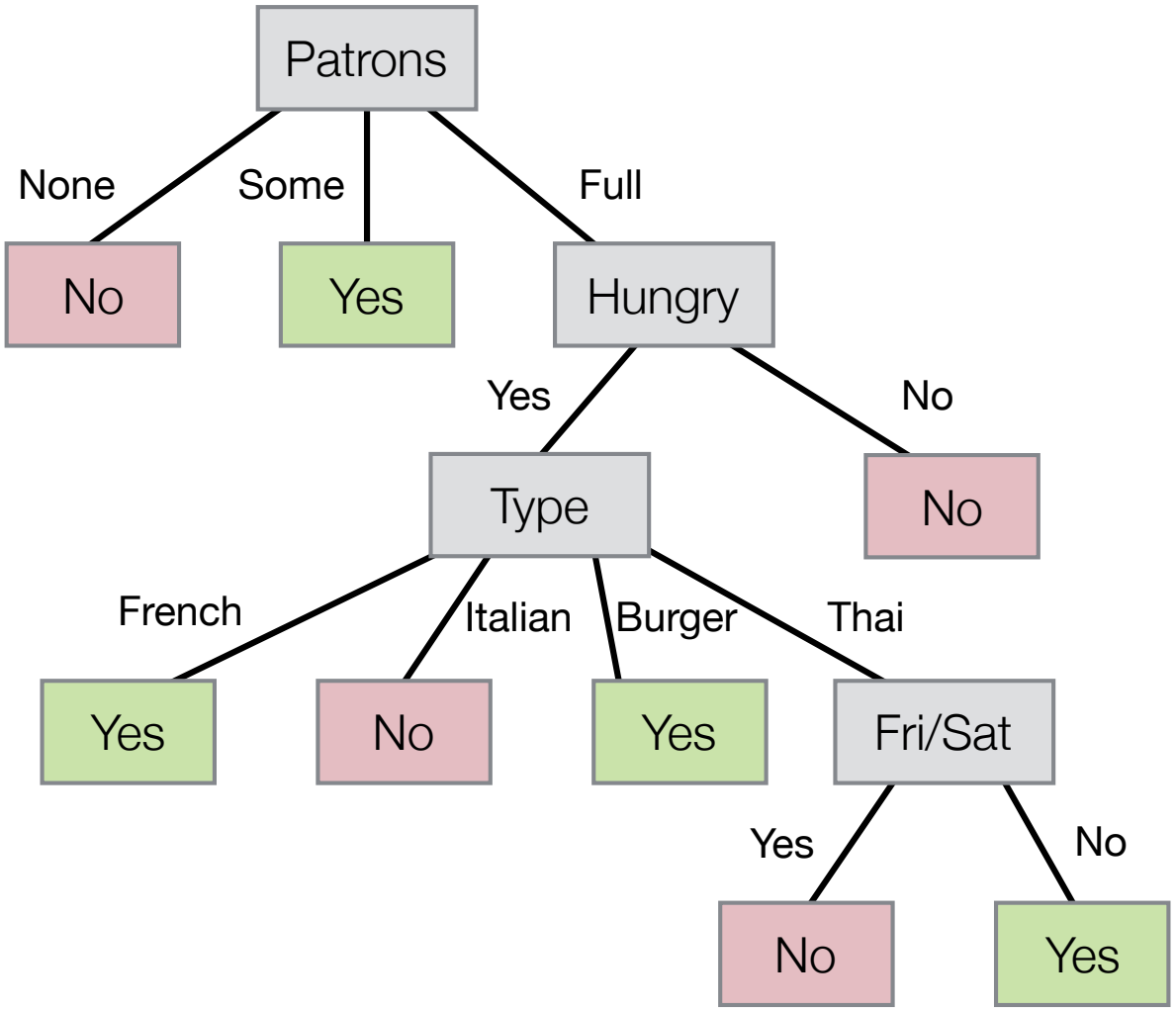
อย่างละ 6(เขียว,แดง)

$$IG(\text{Type}) = 1 - 1 = 0$$

➡ Feature “Patrons” has higher IG, so a better choice for splitting.

Information Gain Example

- ID3 repeats the feature selection + splitting process until all examples have the same class, or no features are left to split.



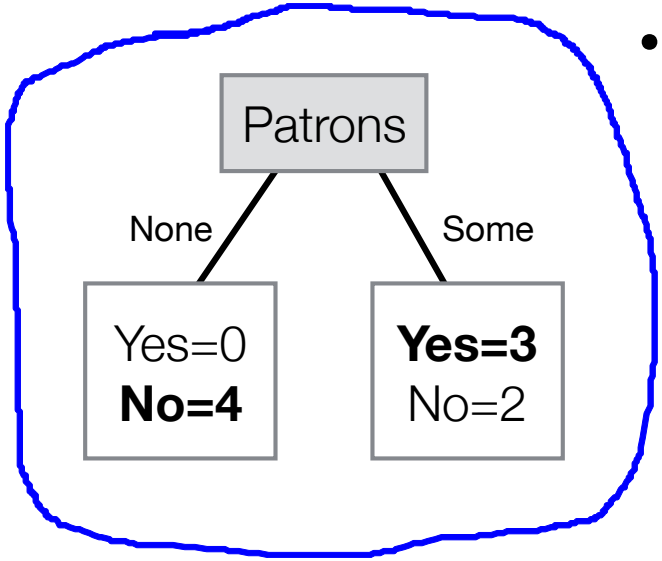
Handling Inconsistent Data

- Inconsistent training data occurs when two identical examples from the training set have different labels:

```
true,female,town,false,yes,no,yes,no = YES
true,female,town,false,yes,no,yes,no = NO
```

Same

- When building a tree, this can result in cases where leaf nodes are not "pure" and no features are left to split.



- Simple solution:
  - For the label, take the majority vote at the leaf node.
  - If there is a tie, randomly choose one of the class labels.

Majority Voting

VodafoneTelCo: Predicting Customer Churn

- Why would VodafoneTelCo want an estimate of the **probability** that a customer will leave the company within 90 days of contract expiration rather than simply predicting whether a person will leave the company within that time?
    - You might want to rank customers their probability of leaving

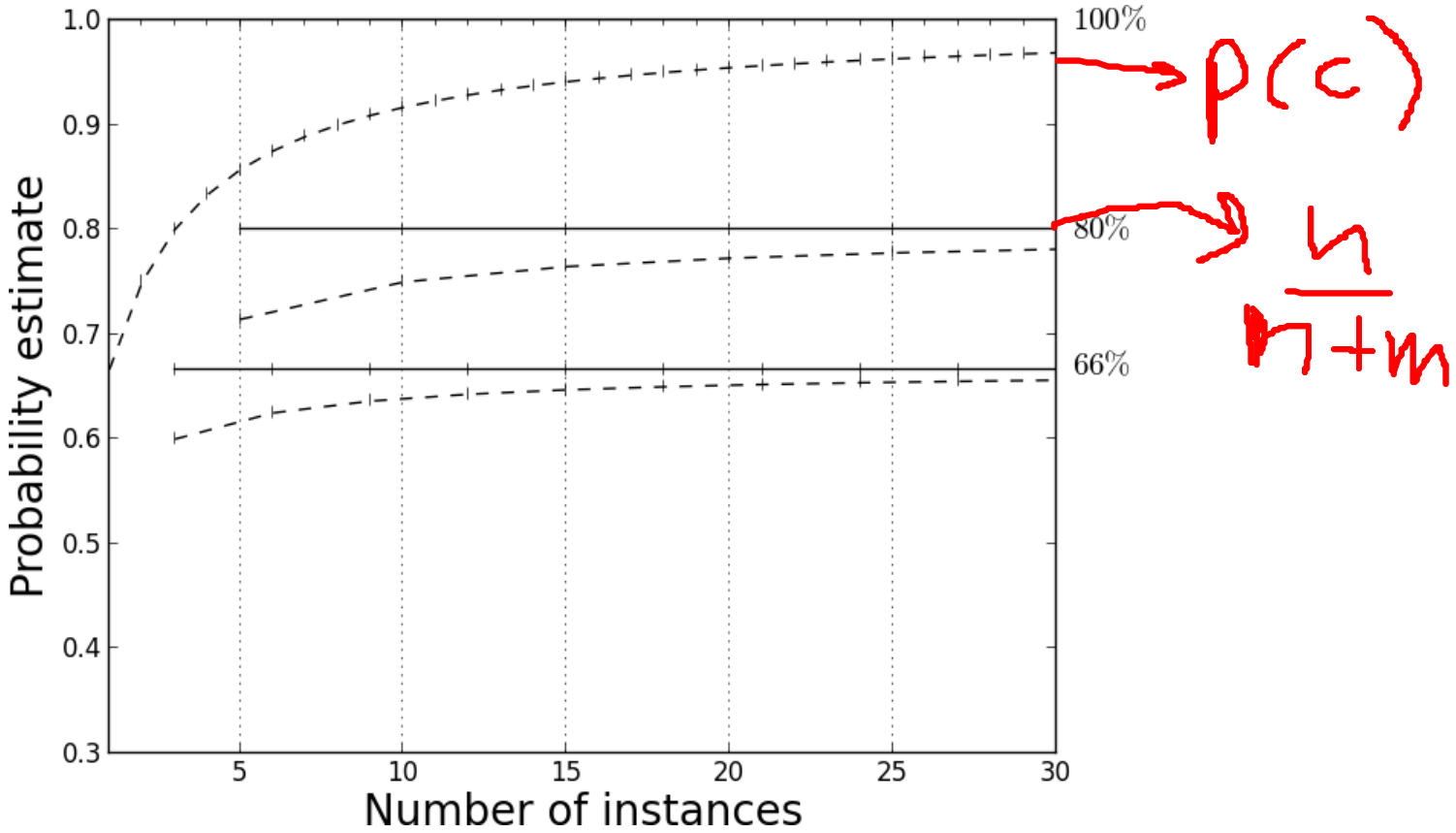
From Classification Trees to Probability Estimation Trees

- **Frequency-based estimate**
  - **Basic assumption**: Each member of a segment corresponding to a tree leaf has the same probability to belong in the corresponding class
  - If a leaf contains n positive instances and m negative instances (binary classification), the probability of any new instance being positive may be estimated as $\frac{n}{n+m}$
- Prone to **overfitting**..

Laplace Correction

- $p(c) = \frac{n+1}{n+m+2}$
  - where n is the number of examples in the leaf belonging to class c, and m is the number of examples not belonging to class c

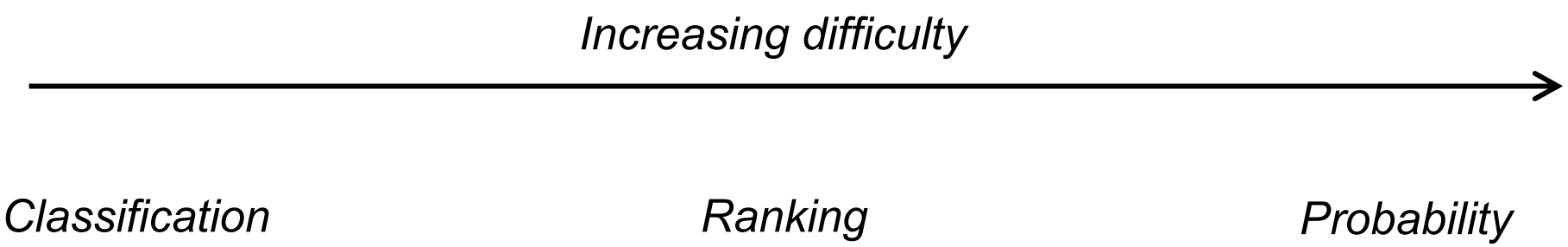
ภาพชัดตอน instance ไม่เยอะ แต่ถ้า instance เริ่มเยอะก็จะยิ่งเข้าใกล้กัน



The many faces of classification: Classification / Probability Estimation / Ranking

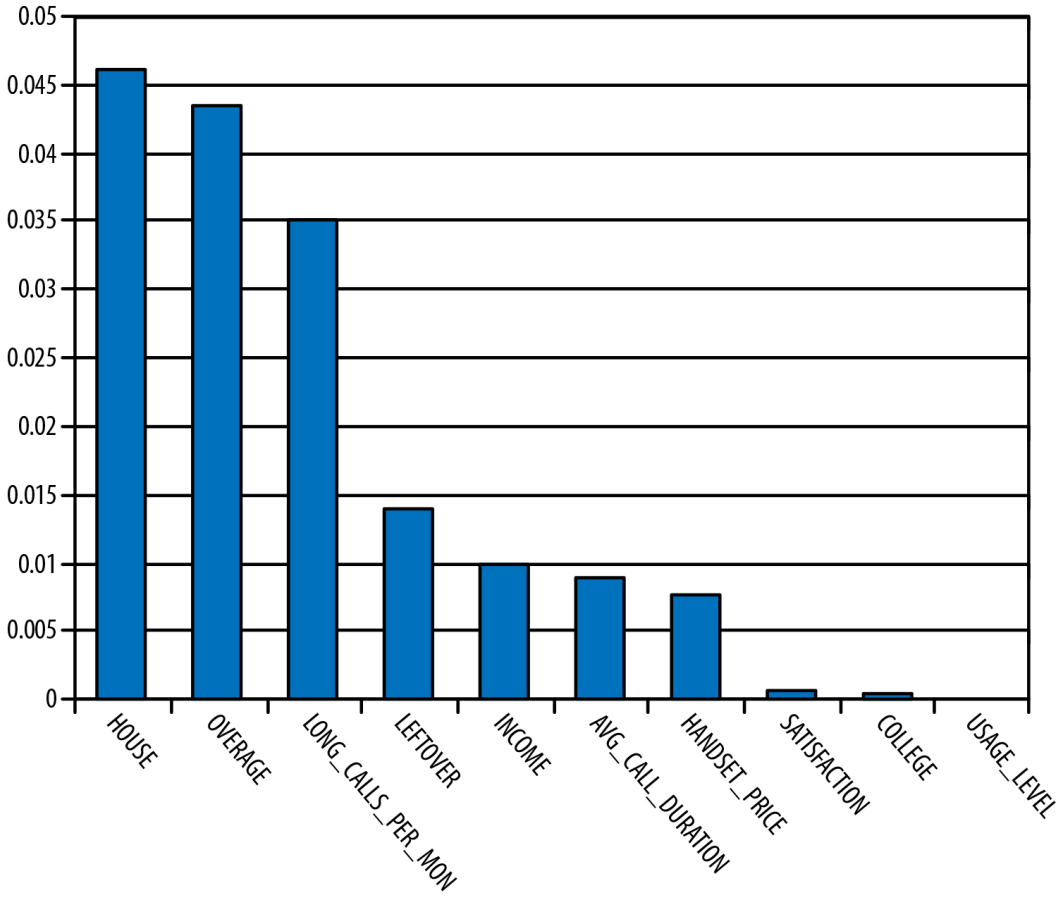
- Classification Problem
  - Most general case: The target takes on discrete values that are **NOT** ordered
  - Most common: binary classification where the target is either 0 or 1
- 3 Different Solutions to Classification
  - Classifier model: Model predicts the same set of **discrete value** as the data had
  - In binary case:
    - **Ranking**: Model predicts **a score** where a higher score indicates that the model think the example to be more likely to be in one class
    - **Probability estimation**: Model predicts **a score between 0 and 1 that is meant to be the probability** of being in that class

The many faces of classification: Classification / Probability Estimation / Ranking



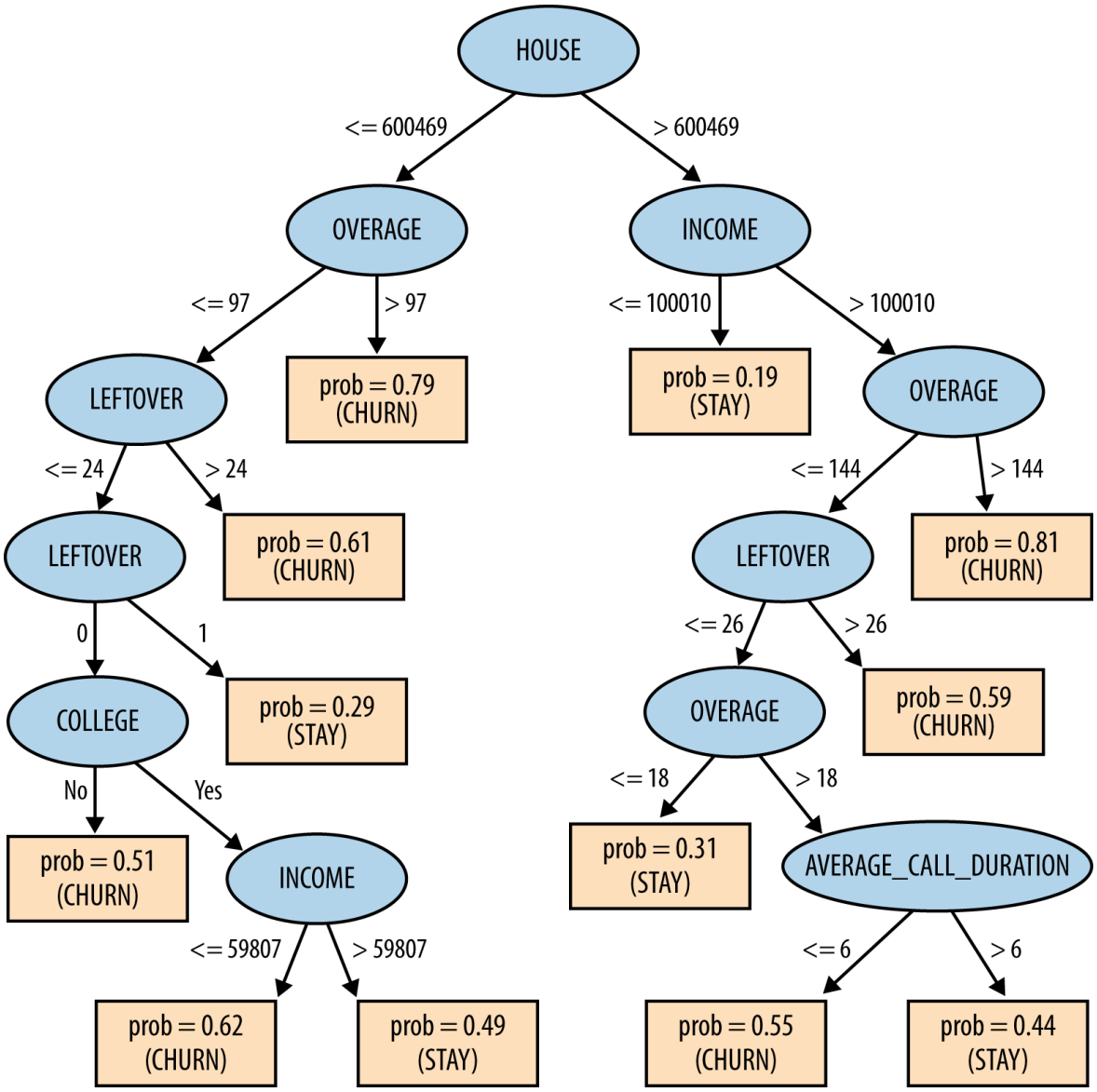
- Ranking:
  - business context determines the number of actions ("how far down the list")
  - cost/benefit is constant, unknown, or difficult to calculate
- Probability:
  - you can always rank / classify if you have probabilities!
  - cost/benefit is not constant across examples and known relatively precisely

VodafoneTelCo: Predicting Churn with Tree Induction



Rank	Info. gain	Attribute name
1	0.0461	HOUSE
2	0.0436	OVERAGE
3	0.0350	LONG_CALLS_PER_MON
4	0.0136	LEFTOVER
5	0.0101	INCOME
6	0.0089	AVG_CALL_DURATION
7	0.0076	HANDSET_PRICE
8	0.0003	SATISFACTION
9	0.000	COLLEGE
10	0.000	USAGE_LEVEL

VodafoneTelCo: Predicting Churn with Tree Induction



C4.5 Algorithm

- C4.5 is an improved version of ID3 algorithm to overcome some of its disadvantages (Quinlan, 1993).
- It contains several improvements to make it "an industrial strength" decision tree learner, including:
  - Handling continuous numeric features.
  - Handling training data with missing values.
  - Choosing an appropriate feature selection measure.
  - Providing an option for pruning trees after creation to reduce likelihood of overfitting.

entropy, gini

ตัด

ที่เอามายกตัวอย่างก่อนหน้านี้

Tree algorithms in a Python package, i.e. **scikit-learn**

- **CART** (Classification and Regression Trees) is very similar to C4.5, but it differs in that it supports numerical target variables (regression) and does not compute rule sets.

  CART constructs binary trees using the feature and threshold that yield the largest information gain at each node.

- scikit-learn uses an optimized version of the CART algorithm; however, scikit-learn implementation does *not* support **categorical variables** for now.

According to the documentation, the training input samples are converted to np.float32

https://scikit-learn.org/stable/modules/tree.html#tree-algorithms-id3-c4-5-c5-0-and-cart

https://github.com/scikit-learn/scikit-learn/issues/12398

Decision Tree Classifier in scikit-kearn

```
>>> from sklearn.datasets import load_iris
>>> from sklearn import tree
>>> iris = load_iris()
>>> clf = tree.DecisionTreeClassifier()
>>> clf = clf.fit(iris.data, iris.target)
```

```
>>> tree.plot_tree(clf.fit(iris.data, iris.target))
```

https://scikit-learn.org/stable/modules/tree.html#tree

Summary

- A decision tree is an eager learning algorithm where the model is induced from the data in the form of decision rules.
- Many different trees can correctly model same training data.
- We want the simplest tree possible that can generalise to new unseen examples.
- Information Gain helps us select features to use when building simple decision trees.
- The ID3 algorithm for decision trees does not handle numeric data. The extended C4.5 algorithm can handle this type of feature.
- An implementation of the **CART** algorithm is available in the **scikit-learn.**

References

- Russell & Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 2009.
- Quinlan, J. R. 1986. “Induction of Decision Trees”. Machine Learning 1, 1 (Mar. 1986), 81-106.
- Mitchell, Tom M. Machine Learning. McGraw-Hill, 1997. pp. 55–58.
- Quinlan, J. R. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.