School of
Computing and
Information Systems

SMU
SINGAPORE MANAGEMENT
UNIVERSITY

IS217

Analytics Foundation

Week 10: **Classification**

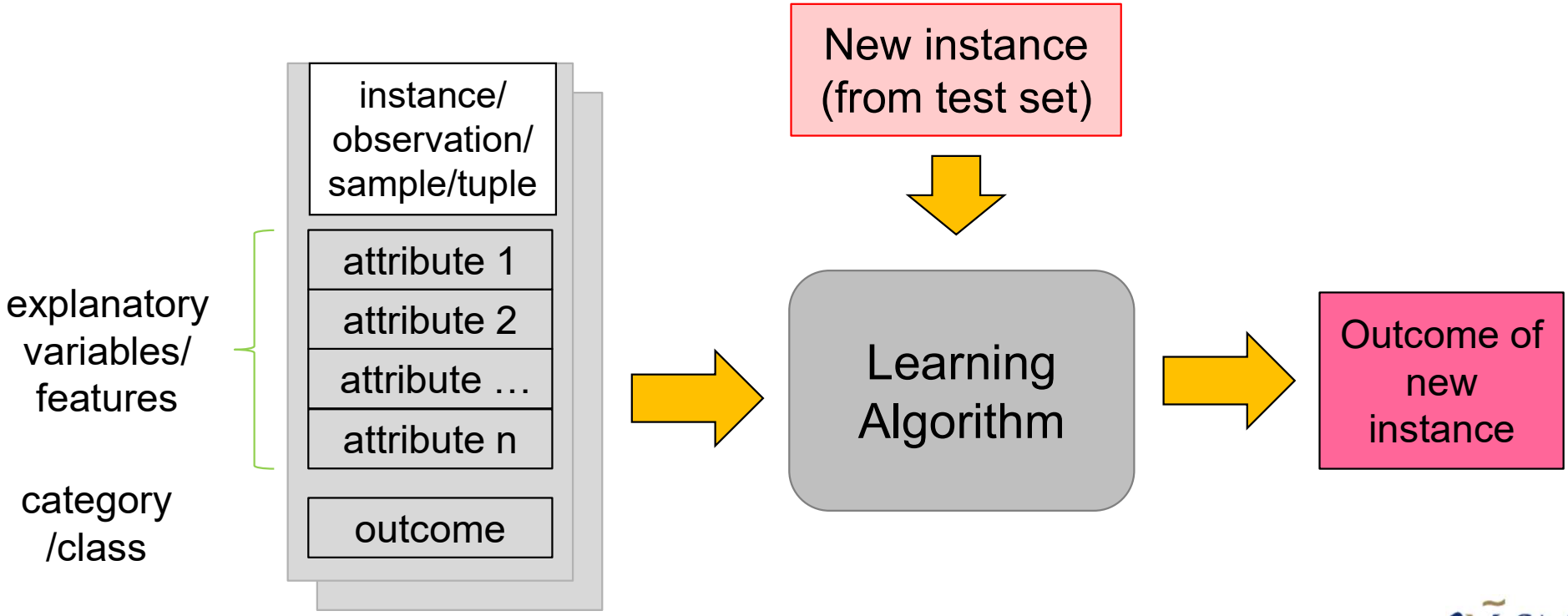
trying to predict a category

Module Outline

1. Introduction to classification
2. Decision Tree
    a. Type of node splitting
    b. Splitting based on entropy
    c. Splitting based on gini index
3. Evaluation metrics

Supervised Learning

- Learning where a training set of actual outcome (target, or label) is available (e.g. Revenue, Accept / Reject, Group 1 to Group N)
- The algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples
- E.g. Classification, regression – data on the target must be available

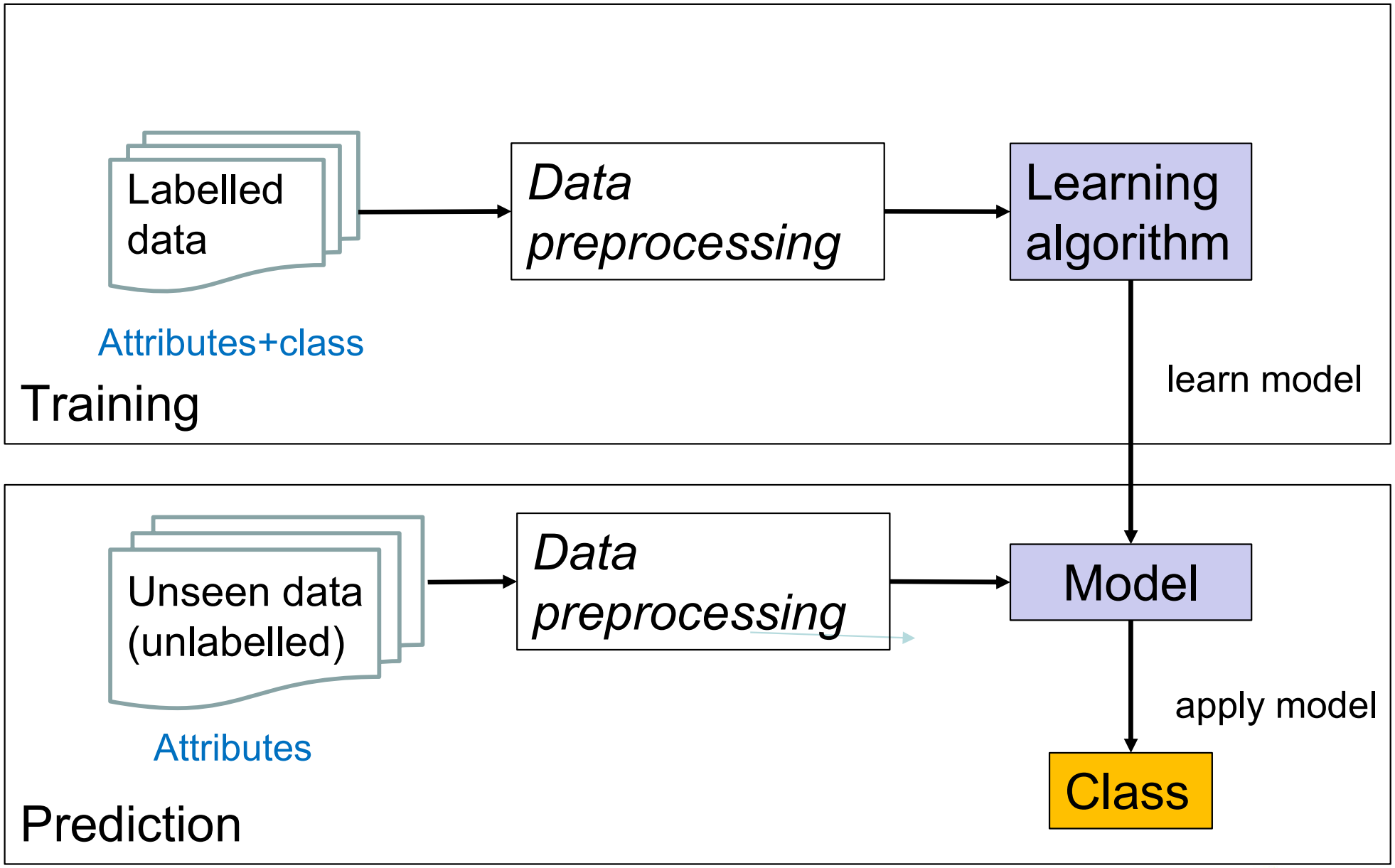


Classification: Definition

- Given a collection of records (*training set*)
  - Each record contains a set of *attributes*, one of the attributes is the *class*
- Builds a *model* for class attribute as a function of the values of other attributes
- Goal: previously unseen records should be assigned a class as **accurately** as possible

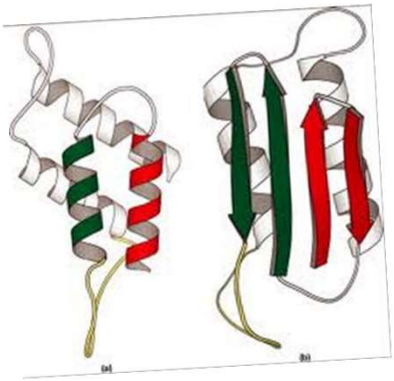
A *test set* is used to determine the **accuracy** of the model. Usually, the given data set is **divided** into **training** and **test** sets, with training set used to build the model, and test set used to evaluate its accuracy.

Classification illustrated



Examples of Classification Task

- Predicting tumor cells as *benign* or *malignant*
- Classifying credit card transactions as *legitimate* or *fraudulent*
- Classifying secondary structures of protein as *alpha-helix, beta-sheet,* or *random coil* in biology
- Categorizing news stories as *finance, weather, entertainment, sports*, etc



Classification Learning Algorithms

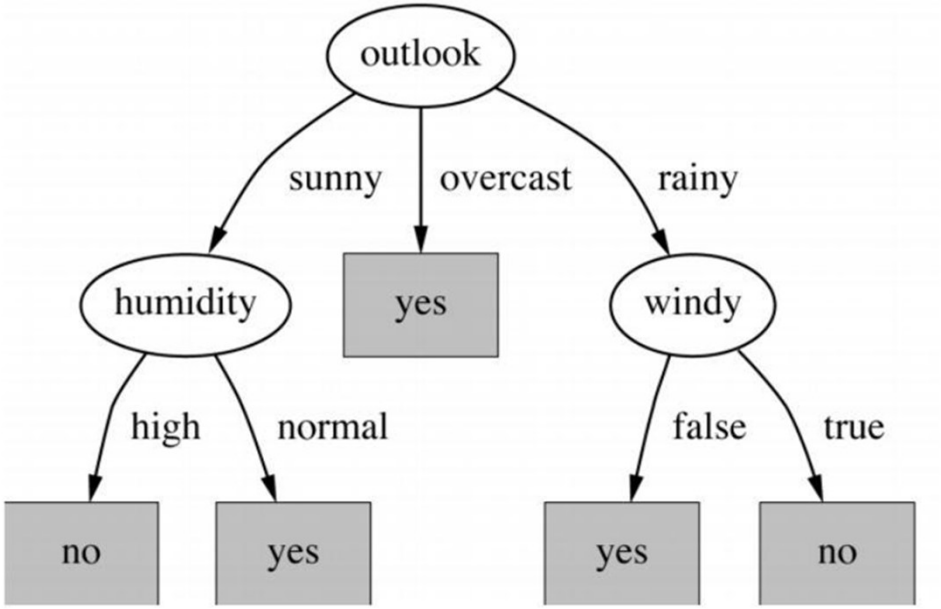
- Decision Tree based Methods
- Naïve Bayes Classifiers
- Support Vector Machines (SVM)
- Neural Networks (NN) and Deep Learning
- Ensemble Classification

Activity: Picking the Perfect Pair of

Find the right pair of glasses for yourself:
https://zingtree.com/host-gallery.php?gallery_id=96

Why Decision Tree?

- Decision tree offers clear representation of decision making steps and helps to understand the data
- Decision tree enables good interpretation of the data with logic and granularity, unlike other algorithms (which are “black box” algorithms)



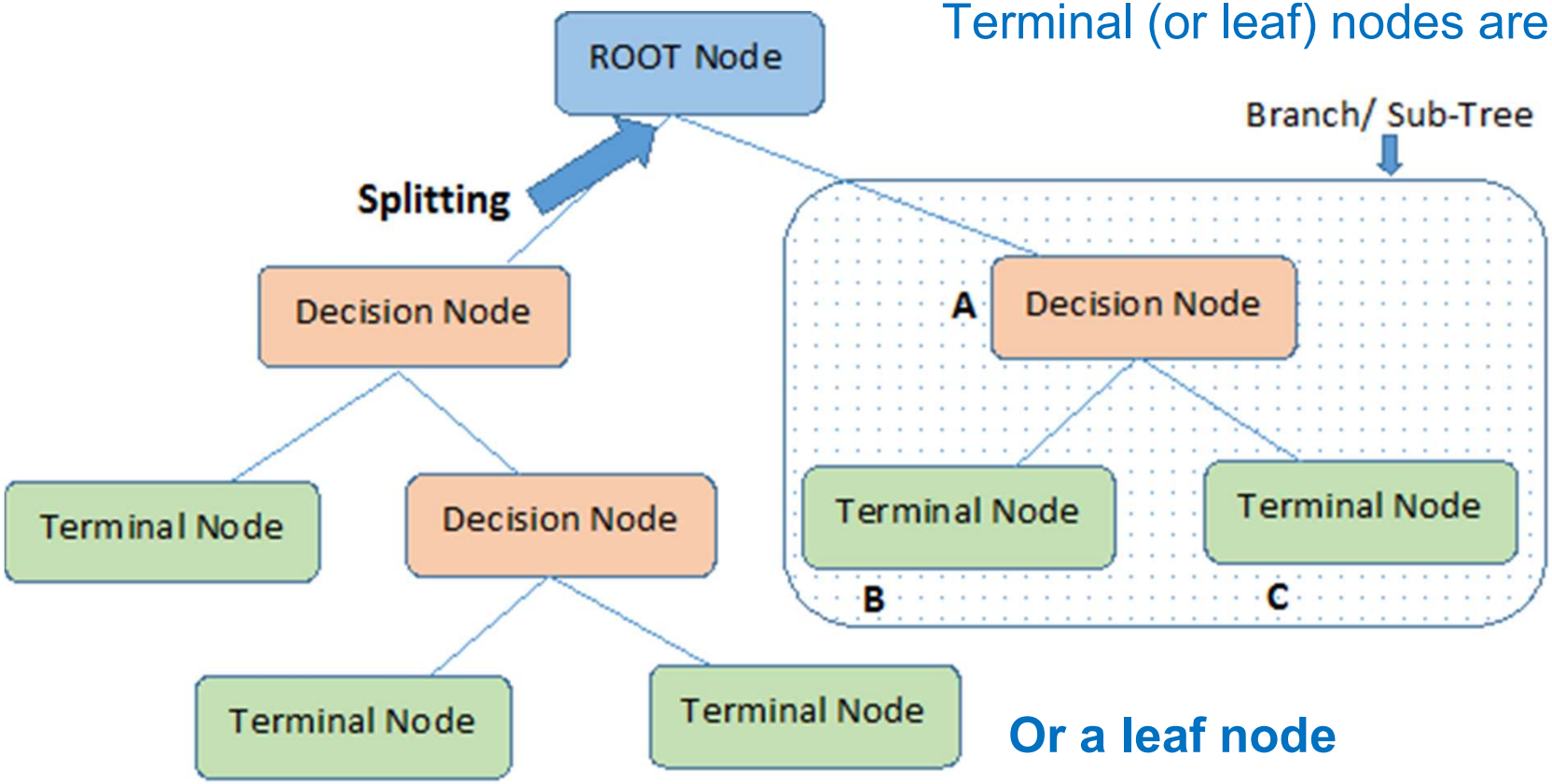
Types of Decision Trees

- **Categorical Variable (or Classification) Decision Tree**
  - Decision Tree which has categorical target variable
  - Example: In above scenario of income tax cheat problem, where the target variable was "if the person will cheat" i.e. YES or NO
- **Continuous Variable (or Regression) Decision Tree**
  - Decision Tree has continuous target variable
  - Example: Decision tree to predict customer income based on occupation, product and various other variables
- Jointly call Classification and Regression Tree (**CART**)

We will cover Categorical Variable (or Classification) Decision Tree in this lesson

Important Terminology

Root and Decision nodes are Attribute
Terminal (or leaf) nodes are class



Note:- A is parent node of B and C.

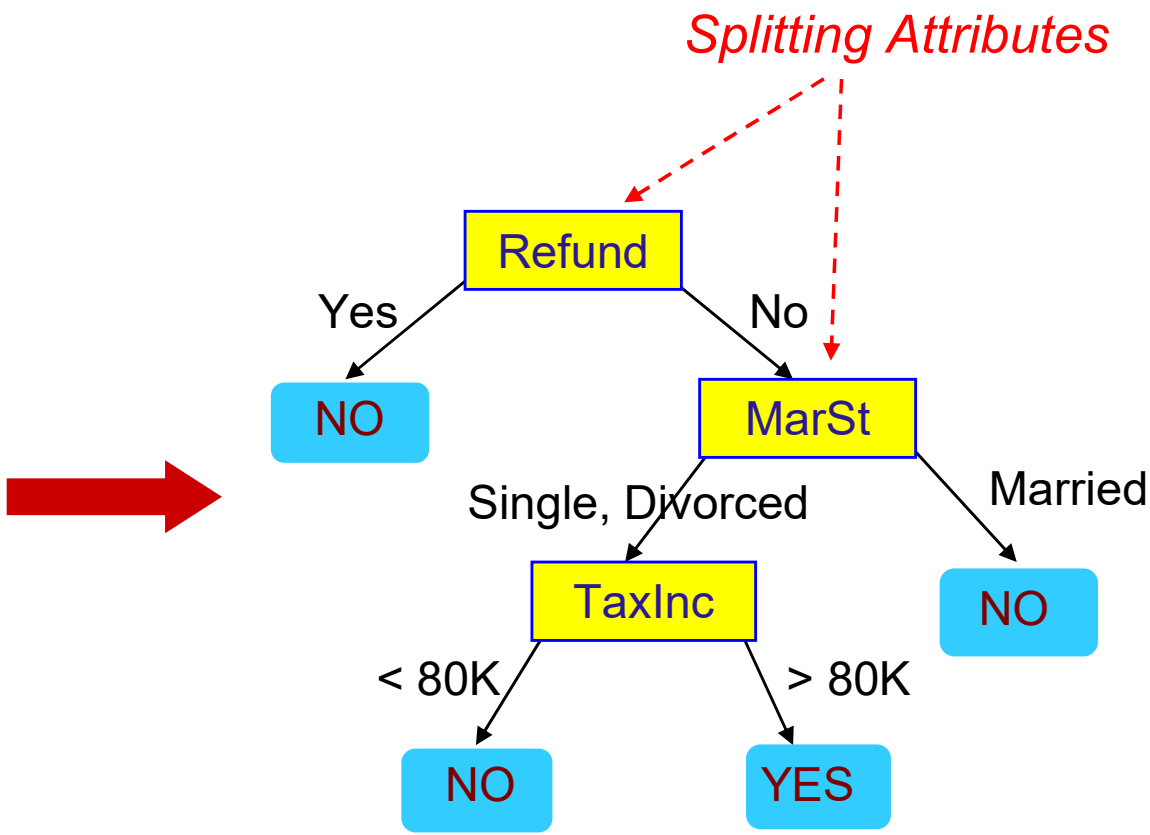
A decision tree is a tree where each node represent a feature (attribute), each link (branch) represents a decision (rule) and each leaf represents an outcome (categorical or continuous value)

https://www.analyticsvidhya.com/blog/2016/04/complete-tutorial-tree-based-modeling-scratch-in-python/

Example of a Decision Tree

Tid	Refund	Marital Status	Taxable Income	Cheat
	categorical	categorical	continuous	class
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

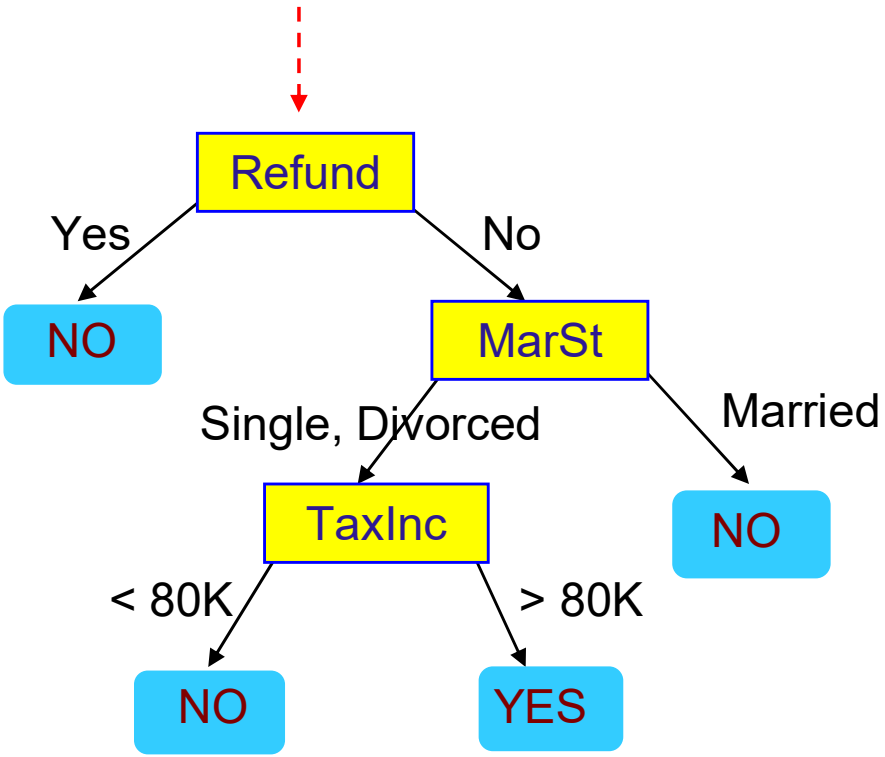


Model: Decision Tree

Every time you hit a decision node, you are splitting up the data set

Apply Model to Test Data

Start from the root of tree.



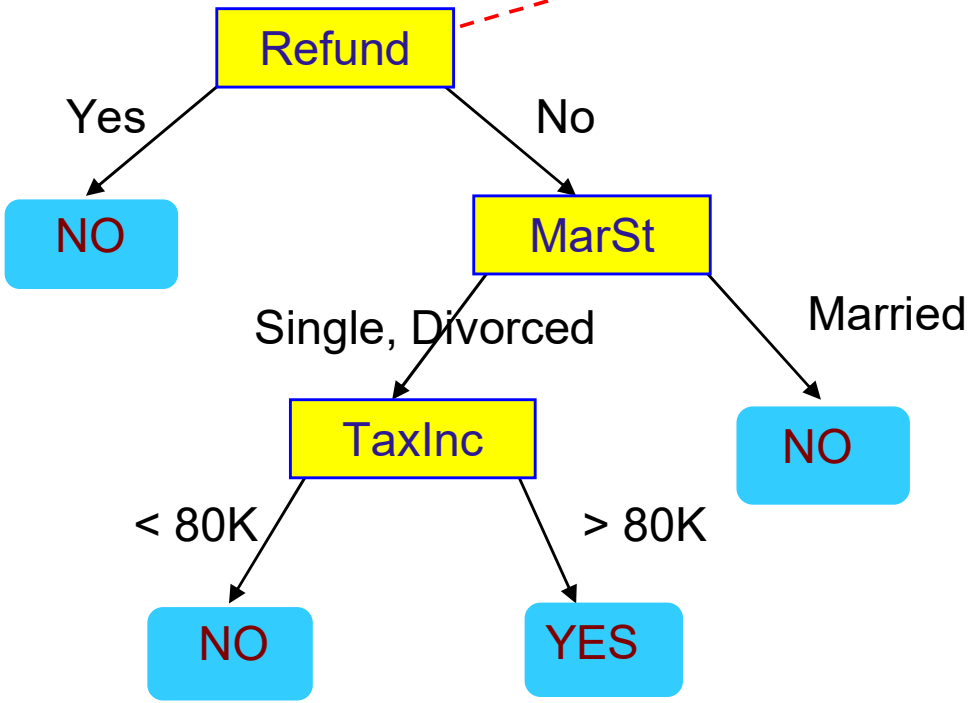
Test Data

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

Apply Model to Test Data

Test Data

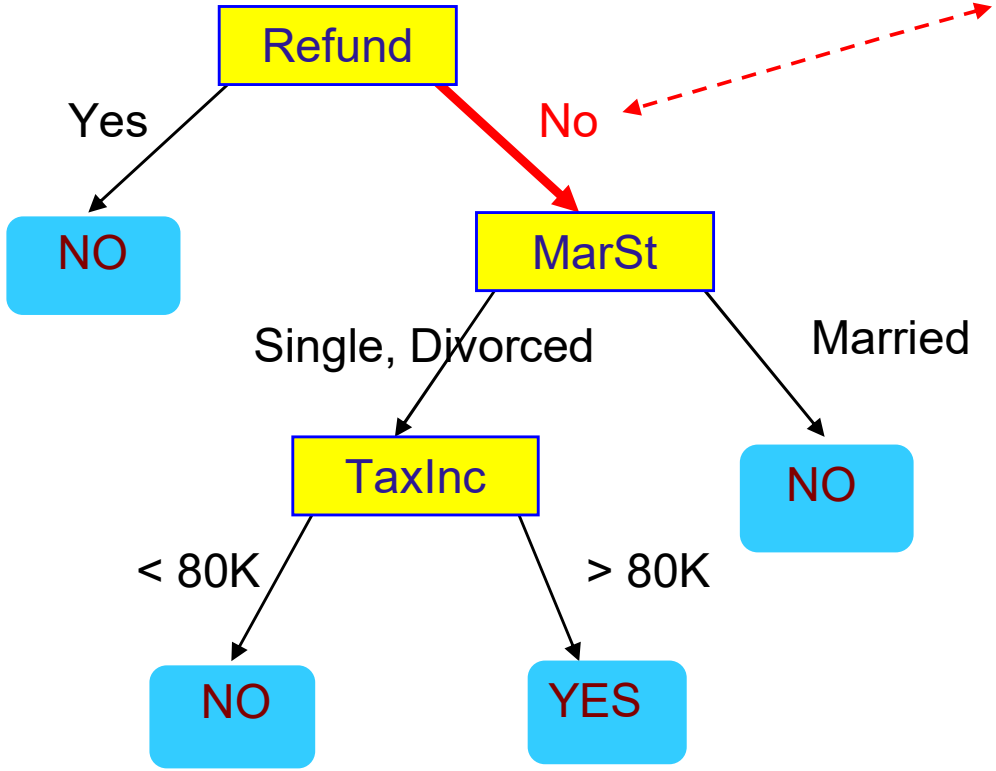
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

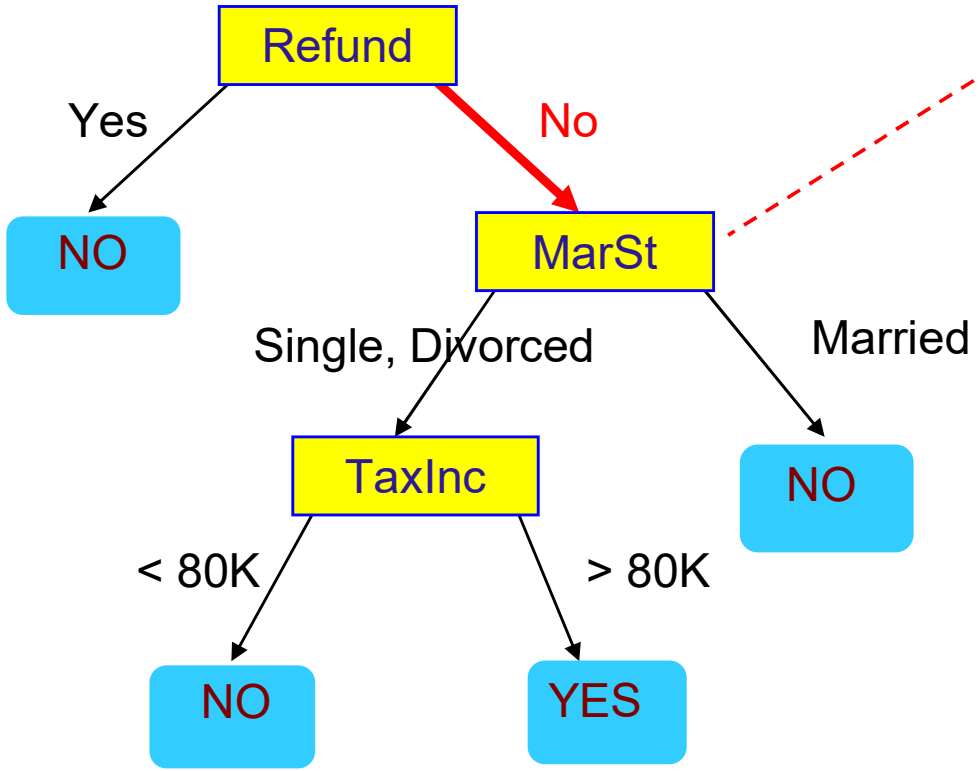
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

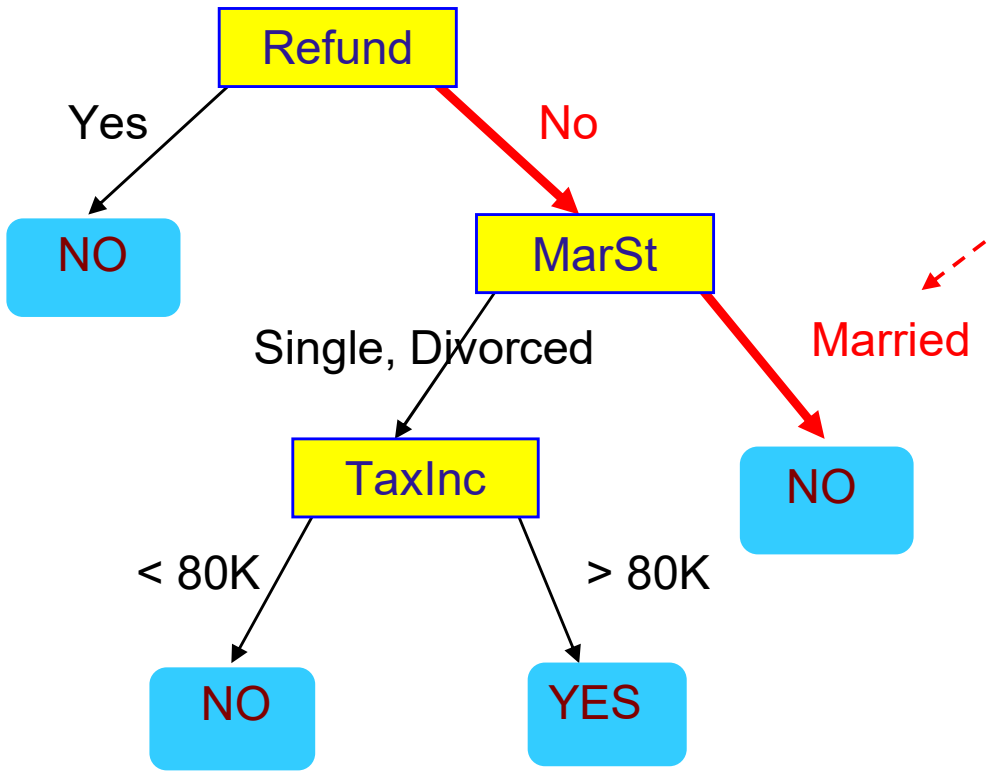
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Another Example of Decision Tree

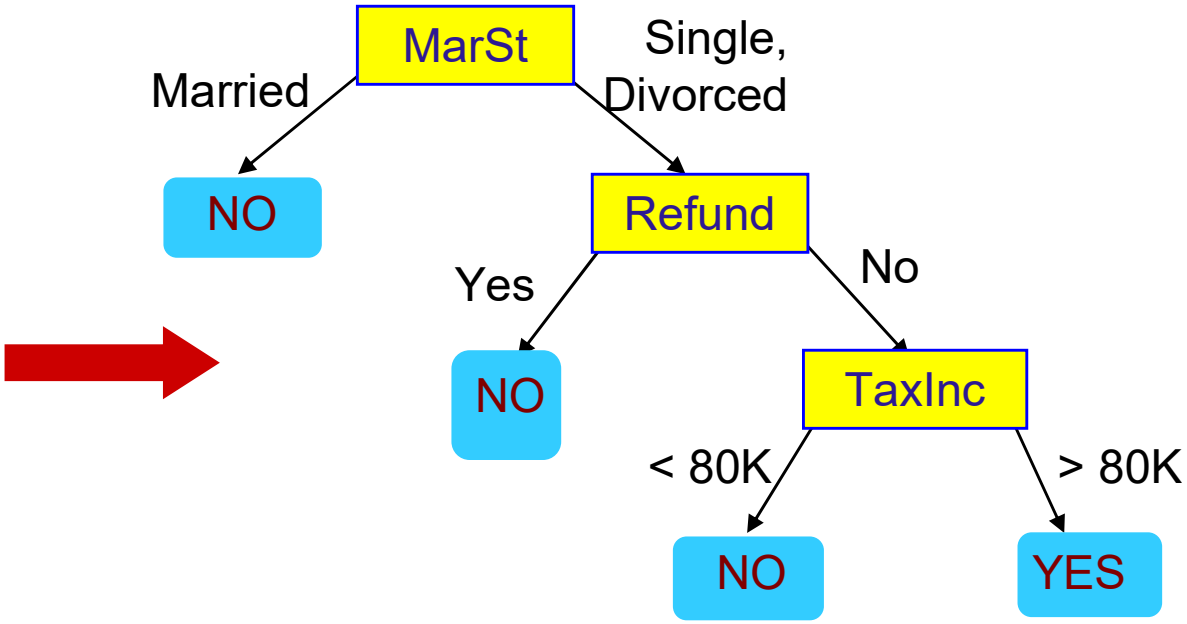
Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

categorical categorical continuous class

Training Data

One dataset can have multiple trees that fits the same data set

Which is the best tree you should pick?



There could be more than one tree that fits the same data!

How to train (or build) a Decision Tree

- A greedy approach

It may not build the best, optimal tree, but it builds a good enough tree
Start at the top and determine which are the best questions to ask
It tries to get to the best result as close as possible mathematically

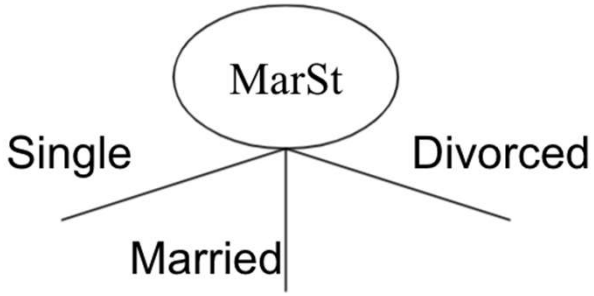
  - Tree is constructed in top-down, greedy manner
    - ***Greedy*** means that each stage, the algorithm chooses what appears to be best at that point, even though it may not be the best decision for the longer term
  - To build a tree, need a root / parent node (attribute that best classifies the training data)
    - ***Split the data*** based on selected attributes (2-way or multi-way)
    - Test attributes selected based on some ***algorithm*** (e.g. entropy)
  - Repeat this for each branch

Stopping conditions for the split

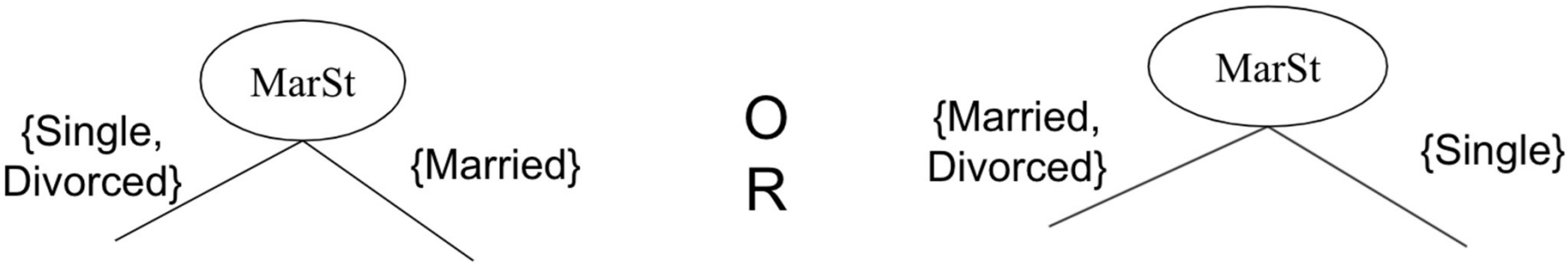
- All data for a given node belong to the same class
- There are no remaining attributes for further splitting
- Number of observations per node is small (e.g. less than 10)

Splitting Based on Categorical Attributes

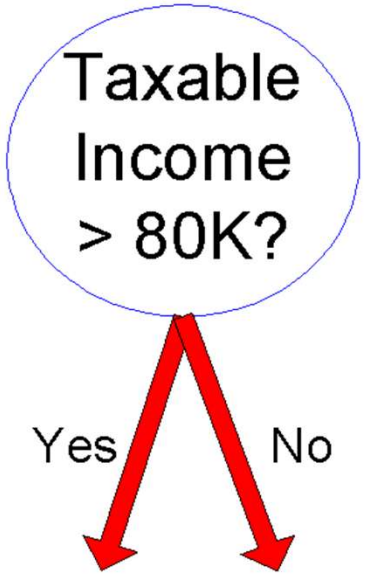
- **Multi-way split:** Use as many partitions as distinct values.



- **Binary split:** Divides values into two subsets. Need to find optimal partitioning.

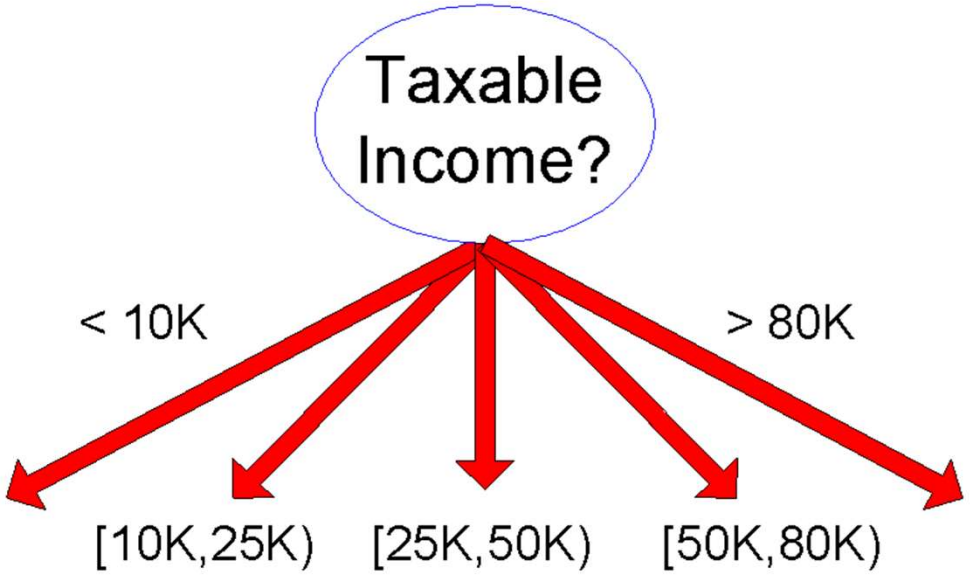


Splitting Based on Continuous Attributes



(i) Binary split

Consider all possible splits and finds the best cut



(ii) Multi-way split

Discretization to form an ordinal categorical attribute

How to determine the Best Split

- Nodes with homogeneous class distribution are preferred
- Need a measure of node **impurity**:

Tree is trying to achieve purity or low purity

C0: 5 C1: 5	C0: 9 C1: 1

Most of them belong to the same class

10 vs 0 is the best

Non-homogeneous,

High degree of impurity

Homogeneous,

Low degree of impurity

Model assumes that the training set is representative of the population and in the future

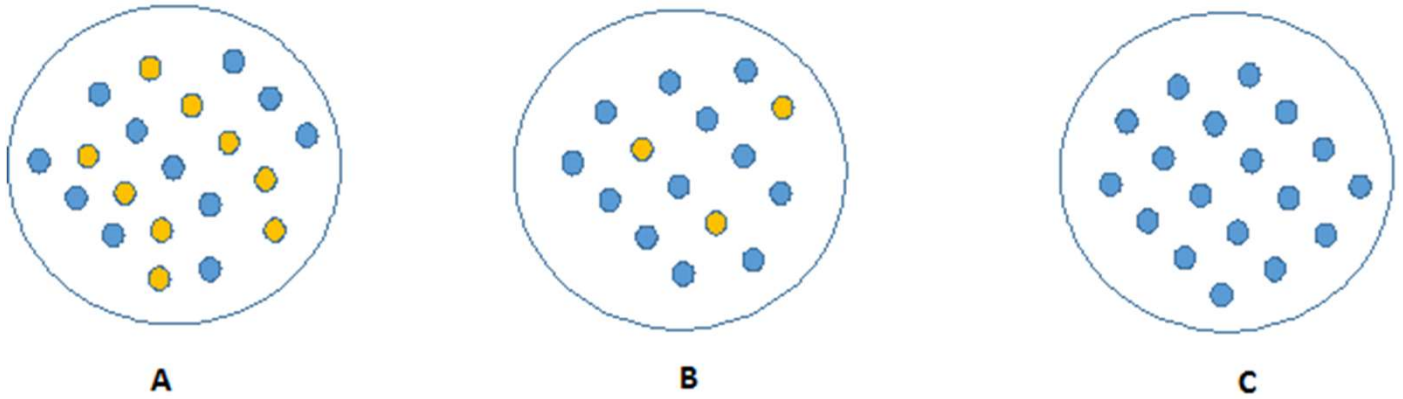
Algorithm to measure the impurity of node

- Entropy

  Entropy is measured in bits (*important!)

  - Measure degree of impurity
    - If the sample (child node) is completely homogeneous, entropy is zero
    - If the sample is equally divided amongst 2 classes, then entropy is one
  - Lower the entropy value, higher the homogeneity
- Gini Index
  - Calculate diversity or heterogeneity from the sum of squared category probabilities
  - Lower Gini value, higher the homogeneity or lower the impurity

Entropy



A – Impure B – Less impure C – Pure node

- How much information is needed to describe a node?
  - C – No information needed since all values same
  - B – A bit more information needed
  - A – A lot more information needed

https://www.analyticsvidhya.com/blog/2016/04/complete-tutorial-tree-based-modeling-scratch-in-python/

Entropy – information gain

- Less impure node → low entropy (closer to zero)
- More impure node → high entropy (closer to one)

- The **information gain** is based on the **decrease in entropy** after a dataset is split on an attribute

- At each point in the decision tree, we choose the attribute that returns the highest information gain (i.e. the most homogeneous branches)

Coin toss to decide who to have dinner with: Entropy = 1
If you had already decided to eat dinner with one of them: Entropy = 0 = You don't need more information
4 people and you have to decide who to give the prize to, toss coin twice, {HH, HT, TH, TT}: Entropy = 2 (2 bits!)
3 people & toss coin: Entropy: Decimal number of bits

How to identify the root node?

- Use the attribute with the largest information gain
- Measure the resulting entropy by using that attribute, which measures the impurity of resulting sets of data

OR

- Use the attribute with the least gini index

Entropy calculation

j = outcome
t = information you just got

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

p(j | t) is probability of class j at (if) node t
log is base 2 ($\log_2$)

– Measures homogeneity of a node

- Maximum ($\log n_c$) when records are equally distributed among all classes, implying the maximum impurity
- Minimum (0.0) when all records belong to one class, implying least impurity

Let's workout special cases of entropy

- Perfectly separated classes
  - Assume there are 2 classes: 0 and 1
  - Since perfectly separated, then using this node, we will get only one class (say 1) and not the other (0)
  - p(0 | t) = 0 and p(1 | t) = 1
  - Entropy = $-0 * \log_2(p(0 \mid t)) - 1 * \log_2(p(1 \mid t))$

    (log base 2 of 0 is undefined; log base 2 of 1 is 0)

    $= 0 - (1 * \log_2 1)$

    $= 0 - (1 * 0)$

    $= 0$

Let's workout special cases of entropy

- Perfectly (evenly) split classes
  - Assume there are 2 classes: 0 and 1
  - Since perfectly split, then using this node, we will get have equal probability for either class
  - p(0 | t) = 0.5 and p(1 | t) = 0.5
  - Entropy = $-0.5 * \log_2(p(0 \mid t)) - 0.5 * \log_2(p(1 \mid t))$
    $= -0.5 * \log_2(0.5) - (0.5 * \log_2(0.5))$
    $= (-0.5 * -1) - (0.5 * -1)$
    $= 0.5 + 0.5$
    $= 1$

2 to the power of -1 gives 0.5

Splitting based on Information Gain

- Information Gain:

$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)$$

Parent Node, p is split into k partitions;

n_i is number of records in partition i

  - Measures Reduction in Entropy achieved because of the split
  - Choose the split that achieves most reduction (maximizes information gain)

Example

- Weather dataset → whether to play game based on weather condition
- Four attributes
  - Outlook
  - Temperature
  - Humidity
  - Windy
- Which attribute should be the root?

What is the total number of records?
Under Outlook, how many 'sunny', 'overcast', 'rainy'?

outlook	temp.	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

You want to try to get entropy as close to 0, reduces the need to toss coin

Info Gain for 'outlook'

j = {'yes' , 'no' }

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

probability of playing tennis + probability of not playing tennis

- Entropy(outlook=sunny) = – 2/5 $\log_2$ (2/5) – 3/5 $\log_2$ (3/5) = 0.971
- Entropy(outlook=overcast) = – 1 $\log_2$ (1) – 0 $\log_2$ (0) = 0
- Entropy(outlook=rainy) = – 3/5 $\log_2$ (3/5) – 2/5 $\log_2$ (2/5) = 0.971

60vs40 is close to 50-50

- Entropy(parent) = – 9/14 $\log_2$ (9/14) – 5/14 $\log_2$ (5/14) = 0.94

a balance of 9v5 is close to 50-50

$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)$$

Weighted Entropy Information for outlook

Info(outlook) = 5/14*0.971 + 4/14*0 + 5/14*0.971 = 0.693

sum of probability of each possible outlook * entropy of each outlook

this is the entropy you will gain tomorrow, even though it is not tomorrow yet.

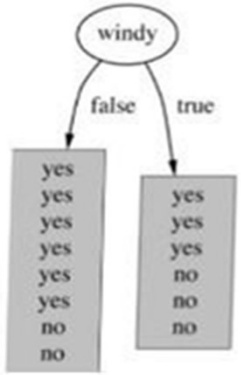
Gain(outlook) = Entropy(parent) – Info(outlook) = 0.247

you will need about 0.247 coin toss to know if you are playing tomorrow

When you know the outlook tomorrow, you will have gained 0.247 bits of information (closer to knowing if you are playing tennis or not tomorrow).

Info Gain for 'windy'

j = {'yes' , 'no' }



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

- Entropy(windy=false) = – 6/8 $\log_2$ (6/8) – 2/8 $\log_2$ (2/8) = 0.811
- Entropy(windy=true) = – 3/6 $\log_2$ (3/6) – 3/6 $\log_2$ (3/6) = 1
- Entropy(parent) = – 9/14 $\log_2$ (9/14) – 5/14 $\log_2$ (5/14) = 0.94

$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)$$

Weighted Entropy Information for windy
Info(windy) = 8/14*0.811 + 6/14*1 = 0.892

Gain(windy) = Entropy(parent) – Info(windy) = 0.048

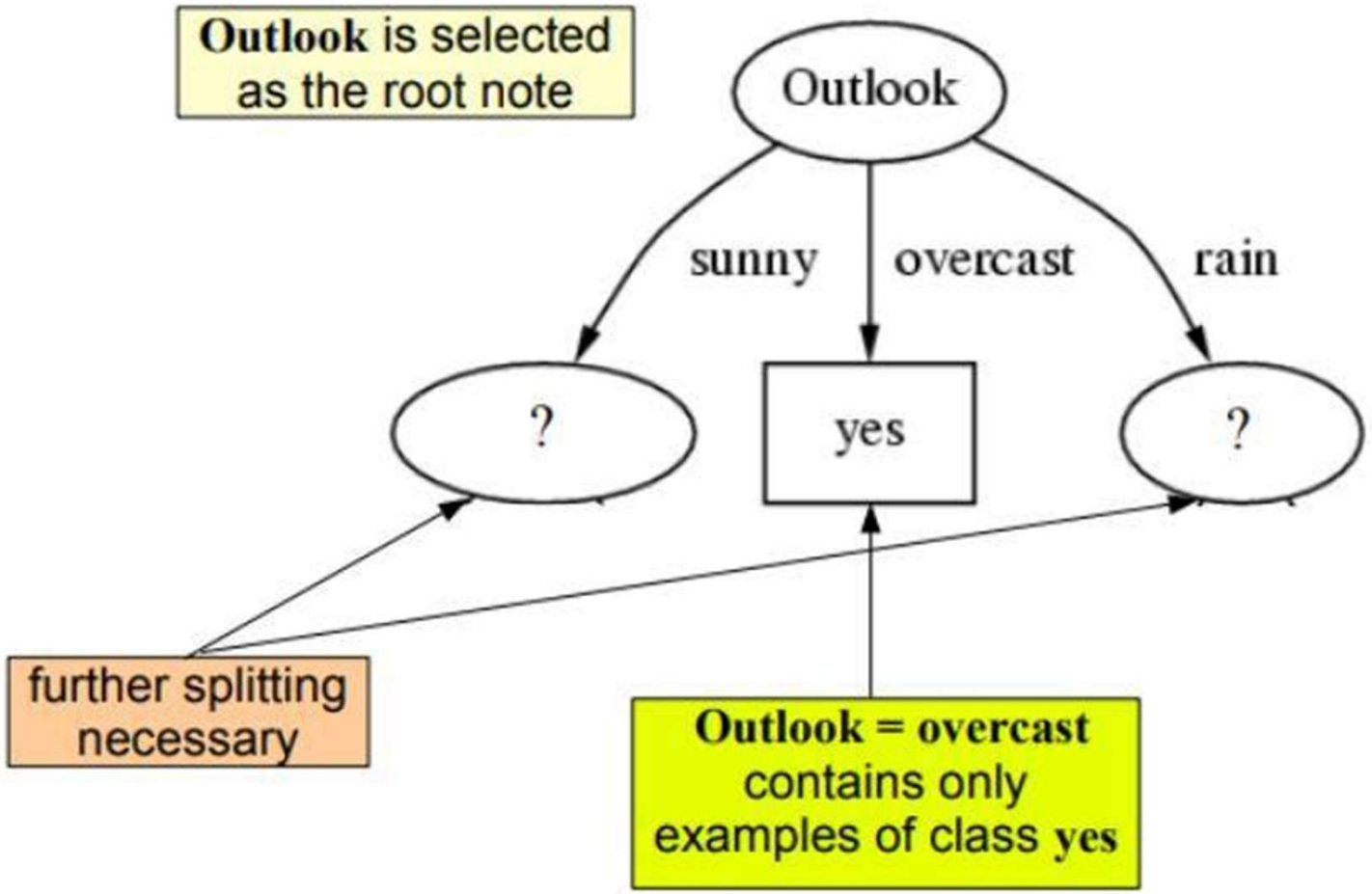
Which should be the root node?

- Calculate the information gain for humidity and temperature

Outlook		Temperature	
Info:	0.693	Info:	0.911
Gain: 0.940-0.693	0.247	Gain: 0.940-0.911	0.029
Humidity		**Windy**	
Info:	0.788	Info:	0.892
Gain: 0.940-0.788	0.152	Gain: 0.940-0.892	0.048

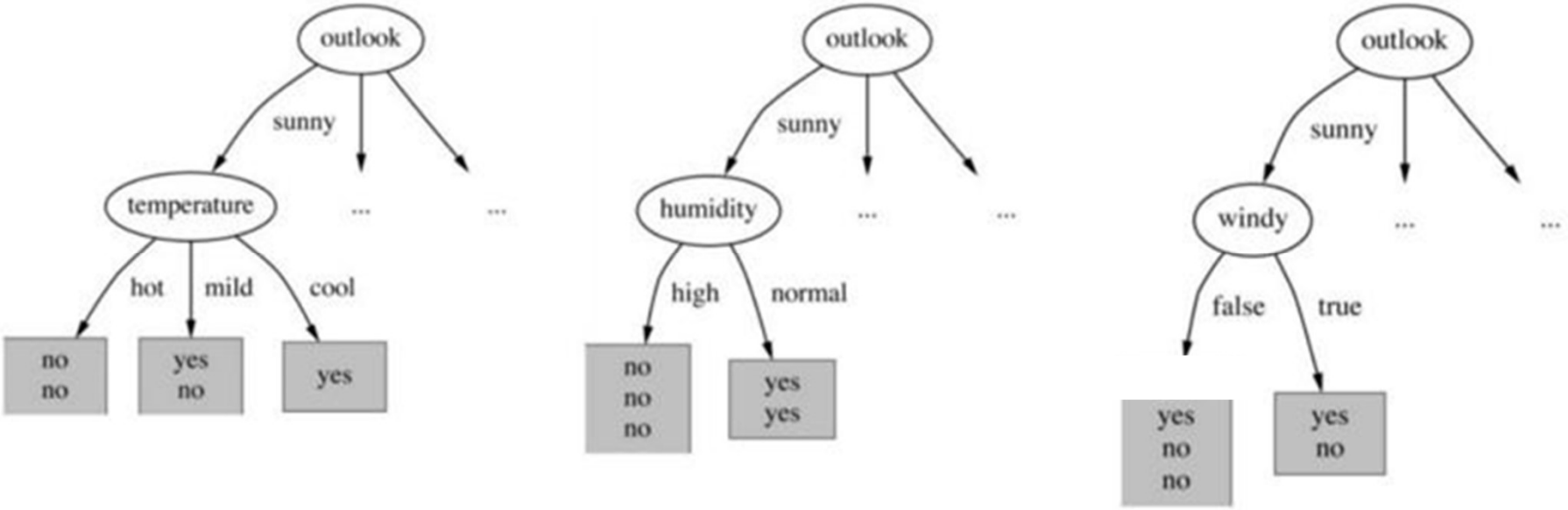
source: https://medium.com/deep-math-machine-learning-ai/chapter-4-decision-trees-algorithms-b93975f7a1f1

The tree with root node



source: https://medium.com/deep-math-machine-learning-ai/chapter-4-decision-trees-algorithms-b93975f7a1f1

Activity – find the node on the left

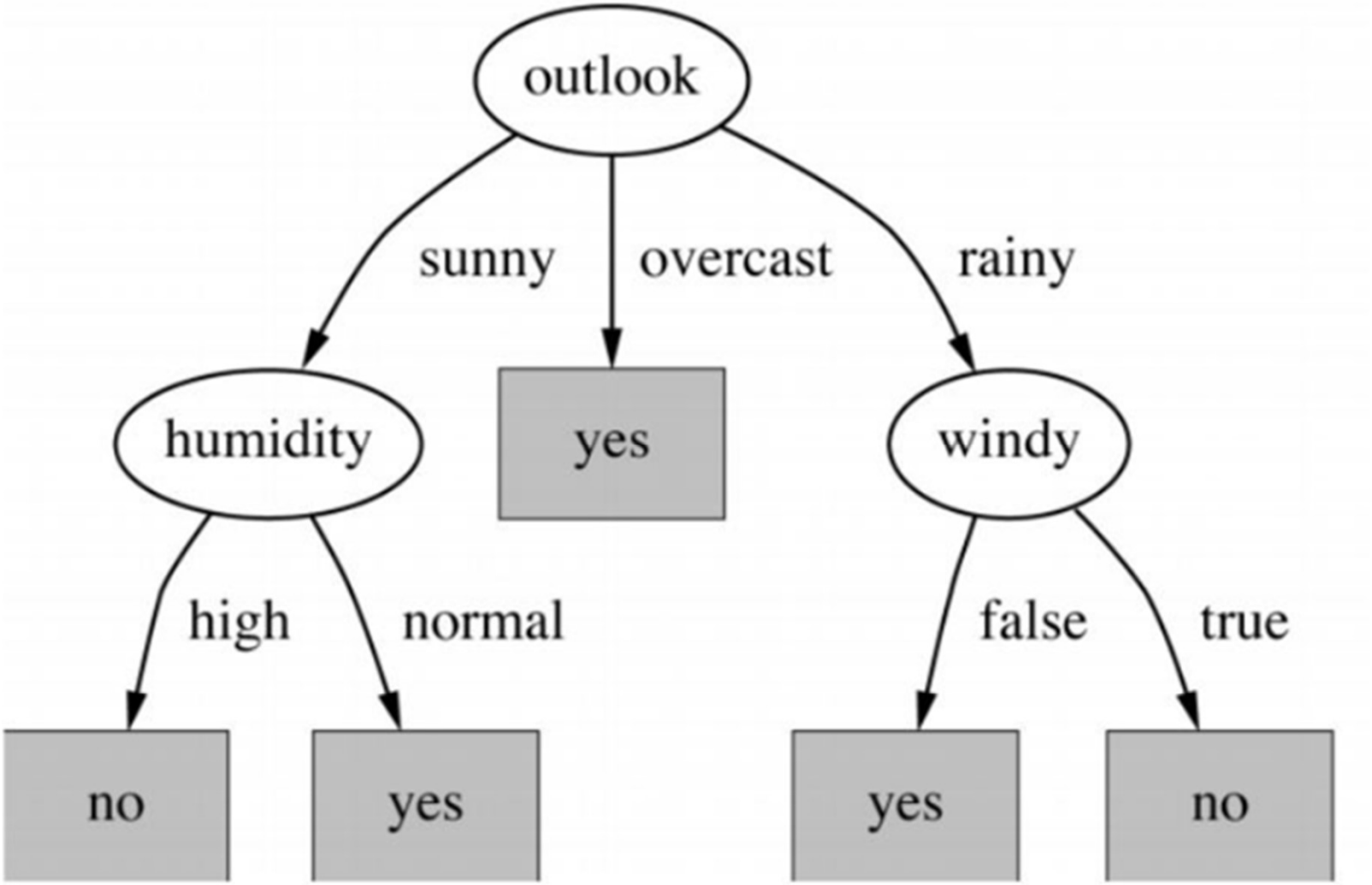


- Which should be selected?
- Is further split required?

Let's work on "InfoGain_calculation - Student.xlsx"

source: https://medium.com/deep-math-machine-learning-ai/chapter-4-decision-trees-algorithms-b93975f7a1f1

Final tree



source: https://medium.com/deep-math-machine-learning-ai/chapter-4-decision-trees-algorithms-b93975f7a1f1

Information Gain Calculation

Steps to calculate information gain for a split:

1. Calculate entropy of parent node
2. For each attribute (node)
    1. Calculate entropy for each sub-nodes
    2. Take weighted information entropy for the node
    3. Calculate information gain
3. Select the highest gain attribute to split
4. Repeat until stopping condition is met

Stopping conditions for the split

Inherent features:

- All observations in the node belong to the same class
- No remaining attributes for splitting

Parameter settings:

- Number of observations per node is small enough
  (e.g. min 10 observation per node)
- The depth of the tree is deep enough
  (e.g. If depth = 3, the tree has maximum 3 levels of splitting)
- The improvement of class impurity is less than a specified minimum

you're still gaining information but it is so small it is not worth it

When to stop splitting?

- When stopping condition is met
- Instead of stopping, it is possible to split until 1 leaf for each observation (100% accuracy), but this causes overfitting
- To avoid overfitting:
  - Set constraints on tree size
  - Tree pruning

you just don't create the initial parts to begin with (in computer science)

Advantage of Decision Tree

- **Easy to Understand**: DT output is very easy to understand even for people from non-analytical background
- **Useful in Data exploration:** DTs easily identify most significant variables and relation between two or more variables
- **Handle outliers:** Not heavily influenced by outliers and missing values

not sensitive to outliers: e.g. it was 123455/1000000 rows, with outlier 123456/1000000 rows
what it should be: 123455/1000000 rows. adding an outlier wouldn't change the results much

- **Data type is not a constraint:** Can handle both numerical and categorical variables

e.g. slide 22. Computer will randomly pick 80k and calculate entropy. Now it picks 60k and entropy is higher, so won't change. Then it picks 100k and entropy is higher, so won't change. Convert numeric variable to categorical then selects the lowest entropy.

- **Non Parametric Method:** Decision tree is considered to be a non-parametric method. i.e. DTs have no assumptions about the space distribution and the classifier structure

https://www.analyticsvidhya.com/blog/2016/04/complete-tutorial-tree-based-modeling-scratch-in-python/

Disadvantage of Decision Tree

tell the decision tree to NOT use all the variables

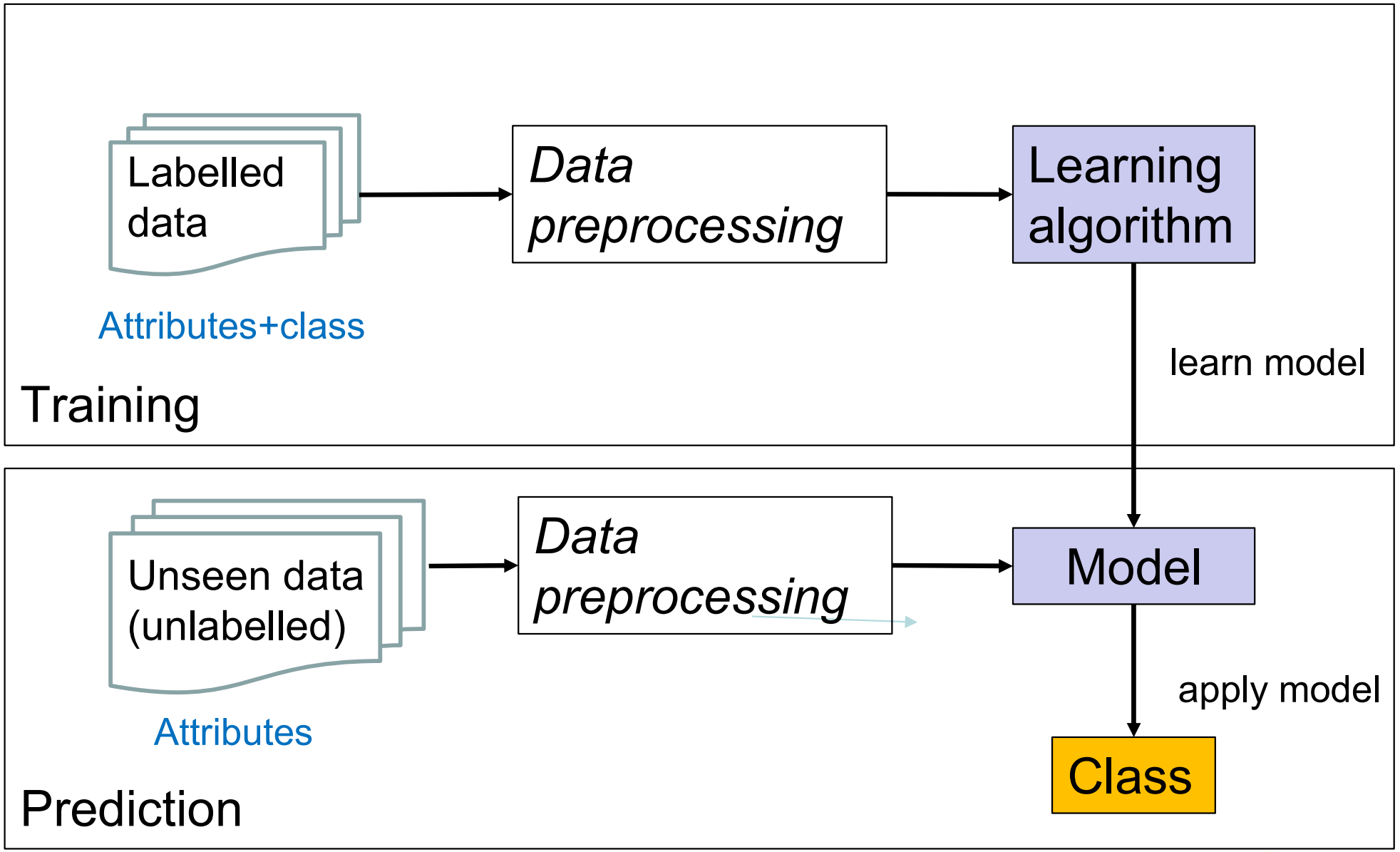
- **Overfitting**: Can easily overfit (to avoid overfitting, use constraints on model parameters and pruning)
- **Limitation for continuous variables:** Continuous numerical variables need to be discretized into categories, thus losing some information

e.g. slide 22: people who earn 70k and 79k are still put into the same category of "No"
Variables are continuous, then yes or no output are categorical and then treated the same, losing some information about what the variable data was

https://www.analyticsvidhya.com/blog/2016/04/complete-tutorial-tree-based-modeling-scratch-in-python/

How to evaluate the prediction?

applies to any classifier, not just decision trees



Confusion Matrix

- A table to describe the performance of a classification model

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

91% correct

- Accuracy: (TP+TN)/total = (100+50) / 165 = 0.91

true positives (TP): Actual 'Yes' record predicted correctly, i.e. 'Yes'
true negatives (TN): Actual 'No' record predicted correctly, i.e. 'No'
false positives (FP): Actual 'No' record predicted as 'Yes'. ("Type I error")
false negatives (FN): Actual 'Yes' record predicted as 'No'. ("Type II error")

FN: thought there is no landmine there, but there is one
FP: thought there is a landmine there, but there is not
you will need to evaluate which outcome is more expensive

FN: guilty of crime, thought not guilty
FP: not guilty of crime, thought guilty
FP is more expensive here

Issue with Accuracy

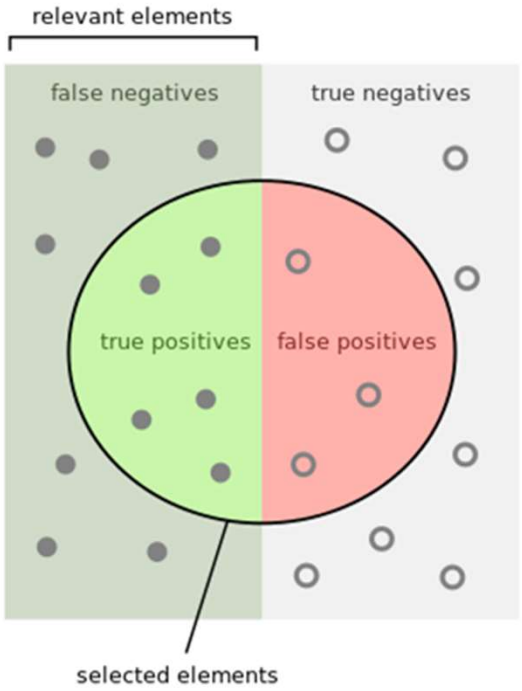
	Predicted Negative	Predicted Positive
Actual Negative	998	0
Actual Positive	1	1

What is the accuracy? 99.9%!

what if the positive is actually someone who is sick and carrying a virus that can spread very quickly? Or the positive here represent a fraud case?

The cost of misclassifying is very expensive!

Evaluation Metrics



How many selected items are relevant?

Precision =

Recall =

	Predicted Negative	Predicted Positive
Actual Negative	998	0
Actual Positive	1	1

People who are predicted positive

- Precision: TP/ (TP+FP) = 1/1 = 1.0

People who are actually positive

- Recall: TP/ (TP+FN) = 1/2 = 0.5

Precision: When it predicts positive, how often is it correct?
Precision is a good measure to use when the costs of False Positive is high and False Negative is low. e.g. email spam detection.
Recall: When it's actually positive, how often does it predict positive?
Recall is a good measure to use when the costs of False Negative is high. e.g. fraud detection, cancer detection.

Evaluation Metrics

- Balanced F-score or F-measure or F1
  - Harmonic mean of precision and recall
  - Defined as

$$2 \times \frac{Precision \times Recall}{Precision + Recall}$$

- F-score provides a measure of accuracy without bias toward precision or recall
- The higher the F-score, the better the classification model

	Predicted Negative	Predicted Positive
Actual Negative	998	0
Actual Positive	1	1

F score is 0.67
(as compare to
accuracy – 0.999)

Practice Question on Evaluation

Analyse the output below and answer the questions for a Classification model using Decision Tree for 10,000 data points

	Predicted Negative	Predicted Positive
Negative Cases	TN: 9,760	FP: 140
Positive Cases	FN: 40	TP: 60

For each question, state the appropriate metric
Accuracy, Precision, Recall
and provide the answer.

What percent of your predictions were correct?

TN + TP

Accuracy: (9,760+60) out of 10,000 = 98.2%

What percent of the positive cases did you catch?

Recall: TP / (TP + FN)

Recall = 60 out of 100 = 60%

What percent of positive predictions were correct?

Precision: TP / (TP + FP)

Precision= 60 out of 200 = 30%

Key Learning Points

1. Classification is a supervised learning approach where samples are labelled by class or category.
2. Decision tree is one example of classification.
3. Decision tree is a upside down tree with root node at the top and leaf nodes (indicating assigned class) at the end.
4. Entropy method can be used to decide how best to split a node.
5. Confusion matrix is used to describe performance of a classification model and F-score is a balanced metric to evaluate a classification model.