Supervised Learning: Classification with Decision Tree

Dr. Yuzana Win (Nagasaki University, Japan)
Lecturer
Department of Computer Engineering and Information Technology

Lecture Objectives

- To introduce
  - Classification Techniques or Methods
  - What is Decision Tree?
  - How does the Decision Tree Works?
  - Pros and Cons

Classification Techniques or Methods

- Naïve Bayes Classifier *
- Support Vector Machines *
- Neural Networks (Deep Learning)
- Decision Tree based Methods
- Rule-based Methods
- K-Nearest Neighbors (kNN) *

Classification Techniques or Methods

- Naïve Bayes Classifier *
- Support Vector Machines *
- Neural Networks (Deep Learning)
- Decision Tree based Methods
- Rule-based Methods
- K-Nearest Neighbors (KNN) *

What is Decision Tree?

- Decision trees are supervised learning algorithms used for both classification, prediction and regression tasks
- The decision tree is an important data structure to solve many computational problems
- The main idea of decision trees is to find the most "information".
- A decision tree mainly contains of three types of nodes:
  - Root node
  - Internal node
  - Leaf or terminal node

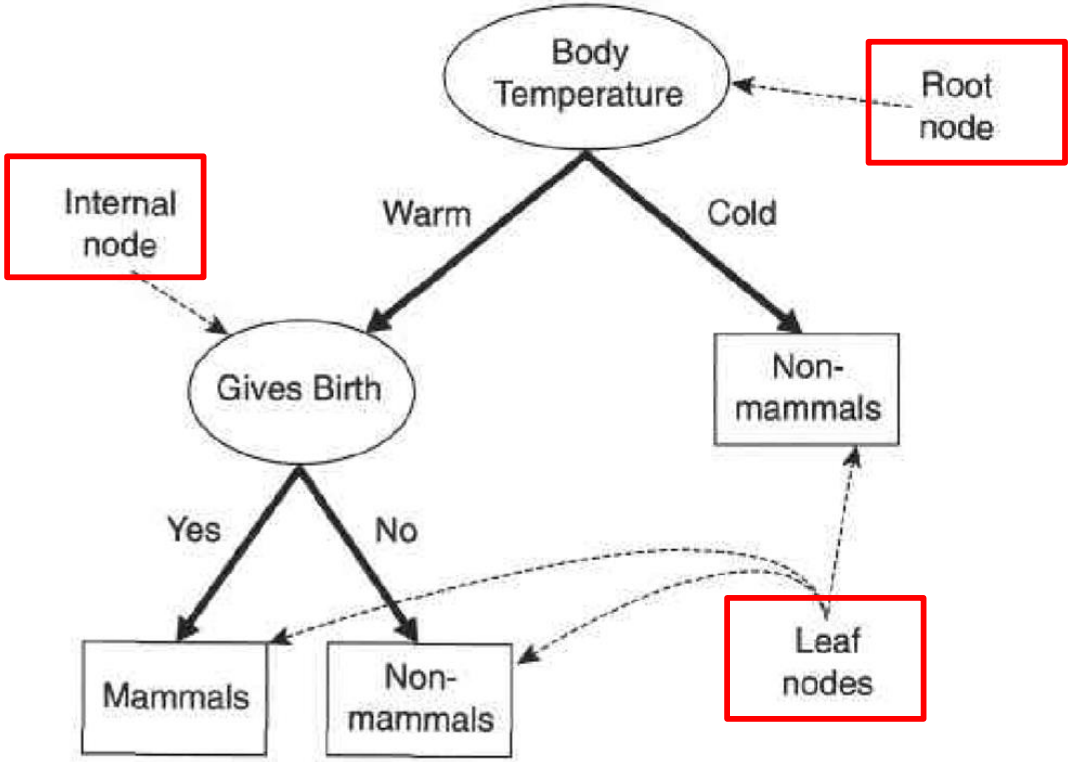
Definition of Decision Tree

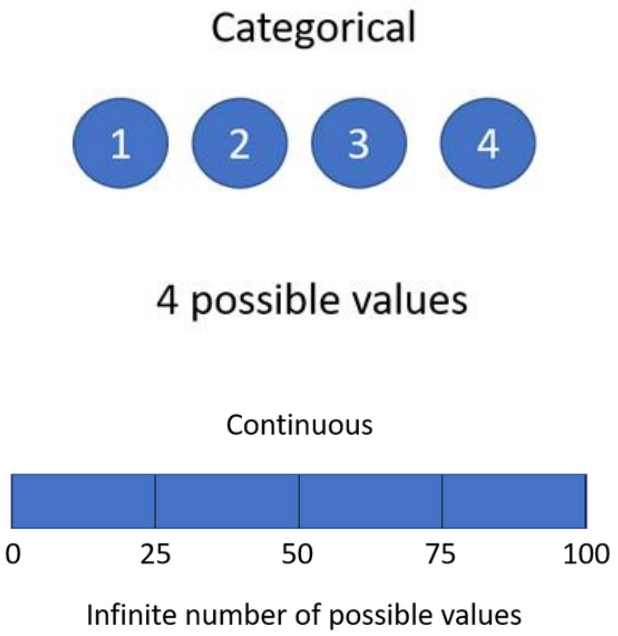
- Given a database $D = \{t_1, t_2, \ldots\ldots, t_n\}$, where t_i denotes a tuple, which is defined by a set of attribute $A = \{A_1, A_2, \ldots\ldots, A_m\}$.
  Also, given a set of classes $C = \{c_1, c_2, \ldots\ldots, c_k\}$.

- A decision tree $\boldsymbol{T}$ is a tree associated with D that has the following properties:
  - Each internal node is labeled with an attribute A_i
  - Each edges is labeled with predicate that can be applied to the attribute associated with the parent node of it
  - Each leaf node is labeled with class c_j

How Decision Tree work?

- Decision trees classify instances or examples by starting at the root of the tree and moving through it until a leaf node.



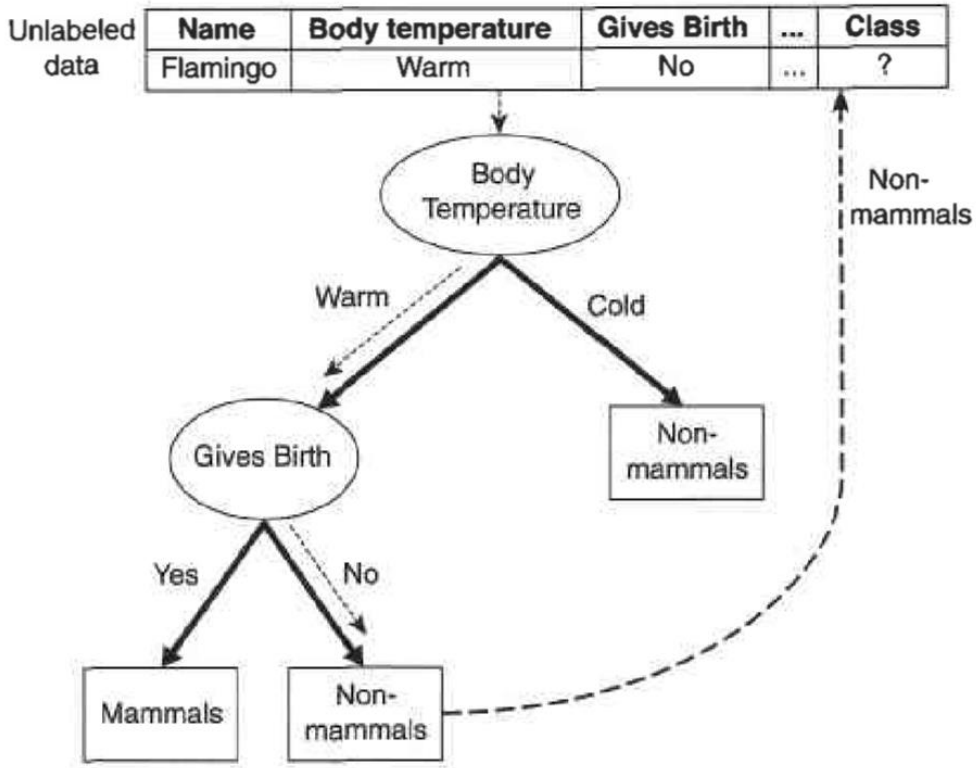


Decision Tree and Classification Task I

Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hibernates	Class
Human	Warm	hair	yes	no	no	yes	no	**Mammal**
Python	Cold	scales	no	no	no	no	yes	**Reptile**
Salmon	Cold	scales	no	yes	no	no	no	**Fish**
Whale	Warm	hair	yes	yes	no	no	no	**Mammal**
Frog	Cold	none	no	semi	no	yes	yes	**Amphibian**
Komodo	Cold	scales	no	no	no	yes	no	**Reptile**
Bat	Warm	hair	yes	no	yes	yes	yes	**Mammal**
Pigeon	Warm	feathers	no	no	yes	yes	no	**Bird**
Cat	Warm	fur	yes	no	no	yes	no	**Mammal**
Leopard	Cold	scales	yes	yes	no	no	no	**Fish**
Turtle	Cold	scales	no	semi	no	yes	no	**Reptile**
Penguin	Warm	feathers	no	semi	no	yes	no	**Bird**
Porcupine	Warm	quills	yes	no	no	yes	yes	**Mammal**
Eel	Cold	scales	no	yes	no	no	no	**Fish**
Salamander	Cold	none	no	semi	no	yes	yes	**Amphibian**

Decision Tree and Classification Task I

- Suppose, a new species is discovered and Decision Tree that can be inducted based on the data is as follows



Once a decision tree is built, it is applied to any test to classify it.

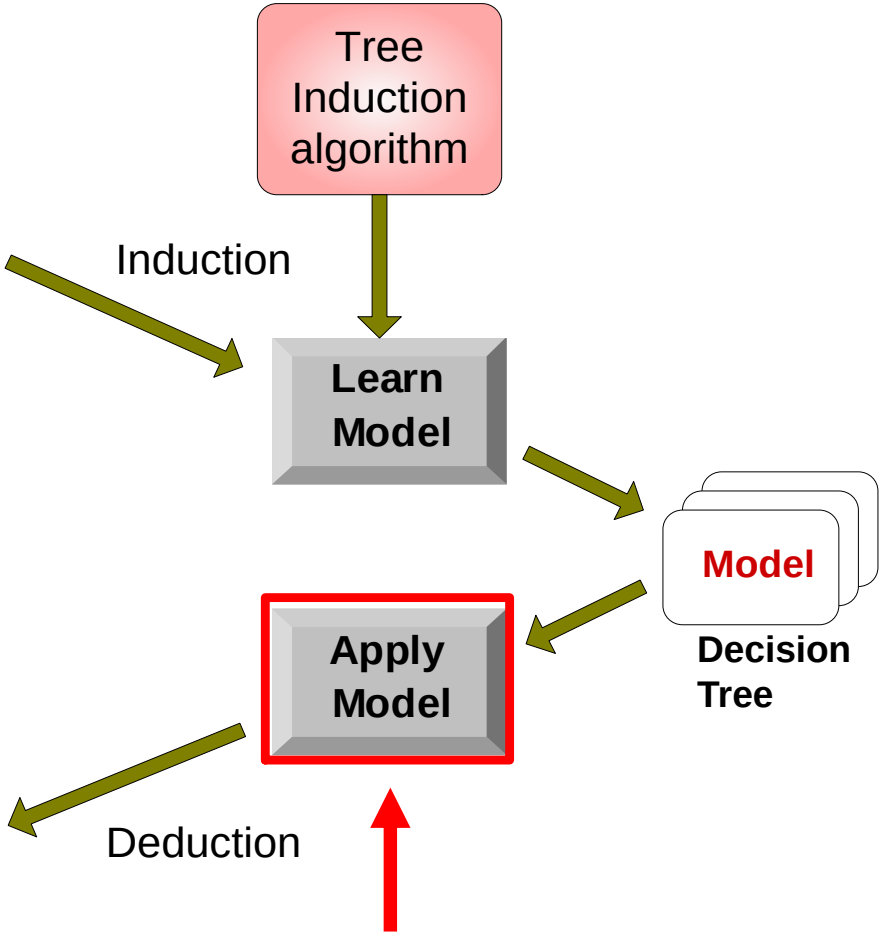
Another Example of Decision Tree Classification Task II

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

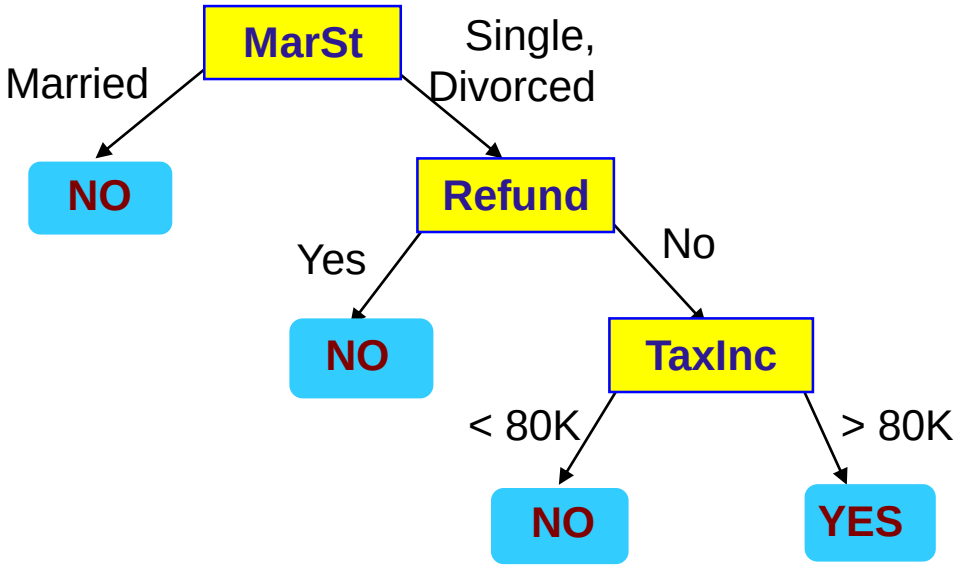
Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

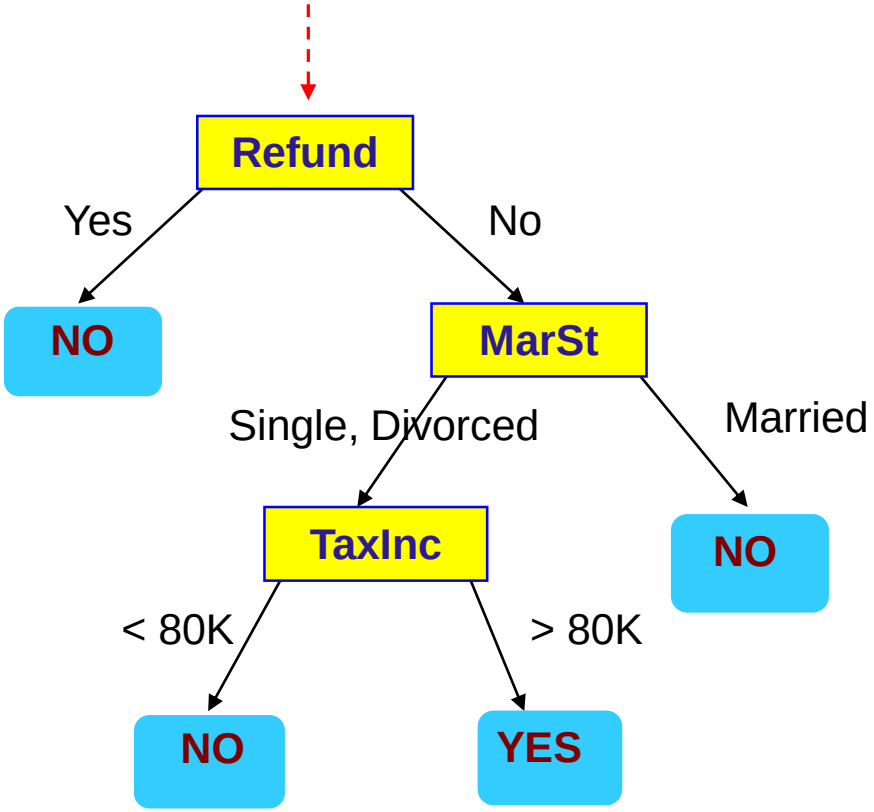


Tid	Refund	Marital Status	Taxable Income	Cheat
	categorical	*categorical*	*continuous*	*class*
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



Apply Model to Test Data

Start from the root of tree.



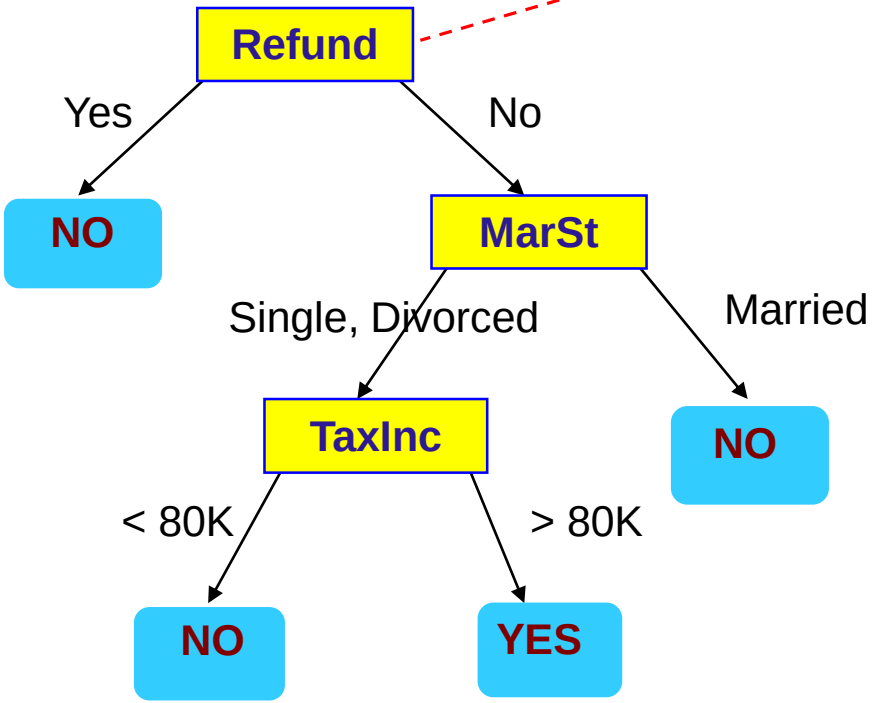
Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Apply Model to Test Data

Test Data

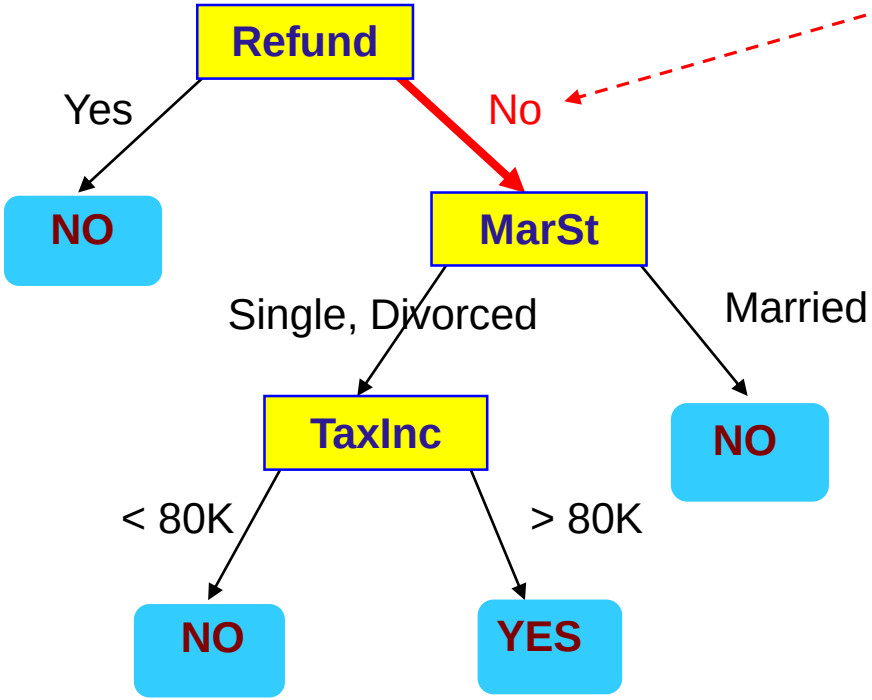
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

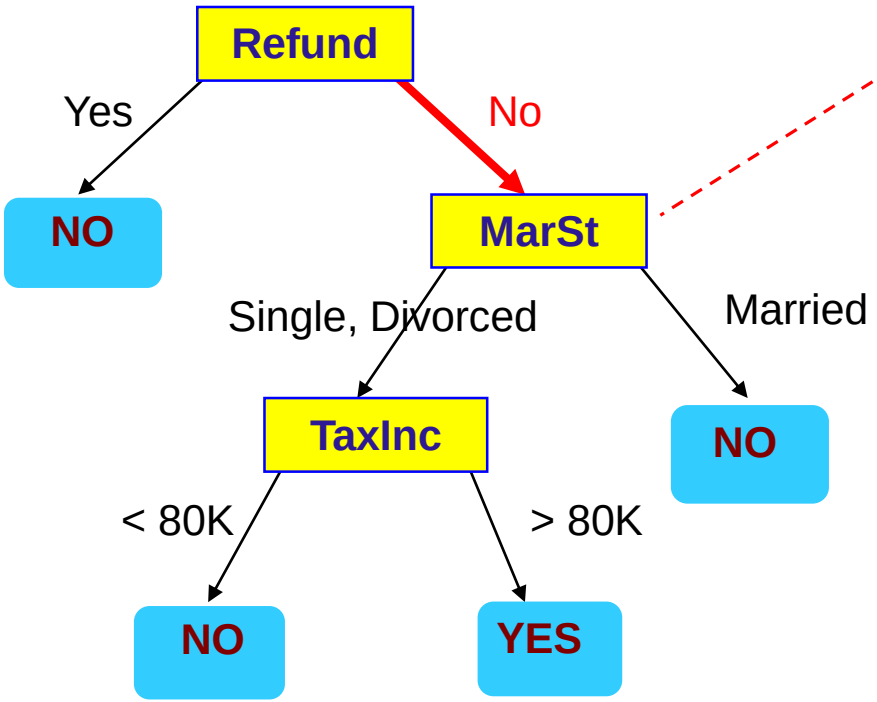
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

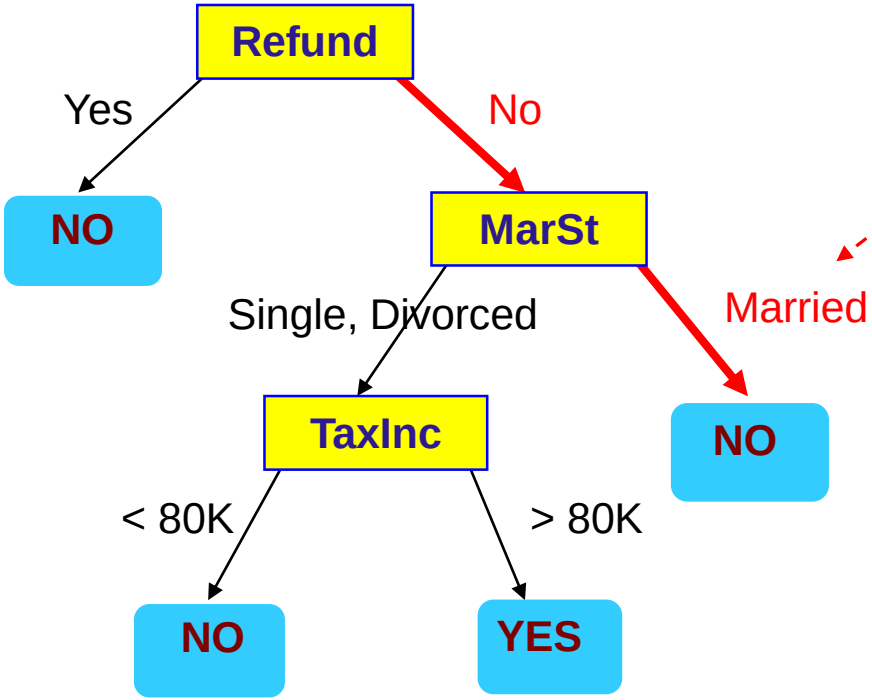
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

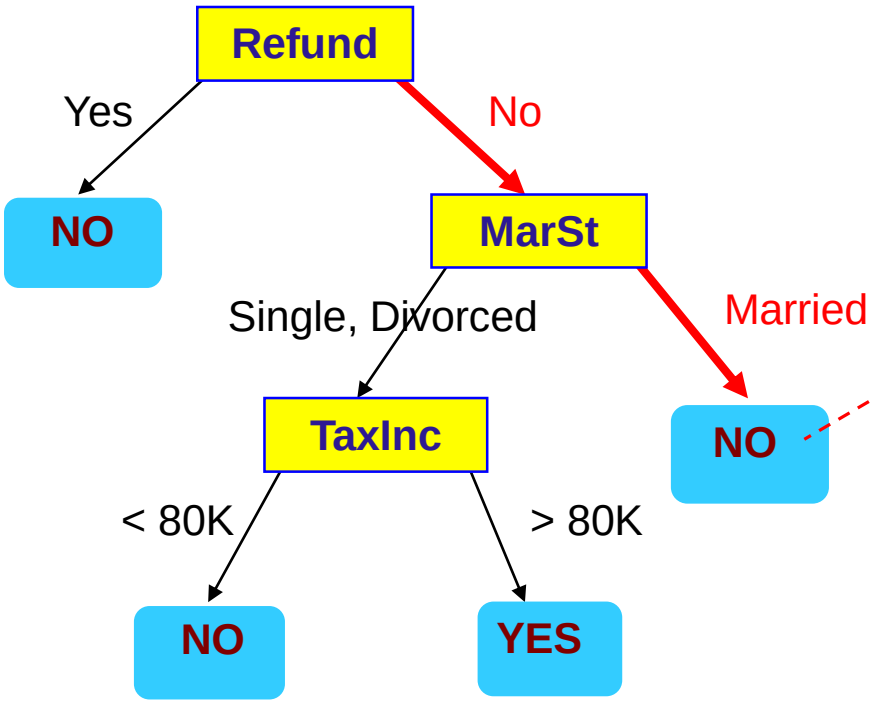
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Apply Model to Test Data

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



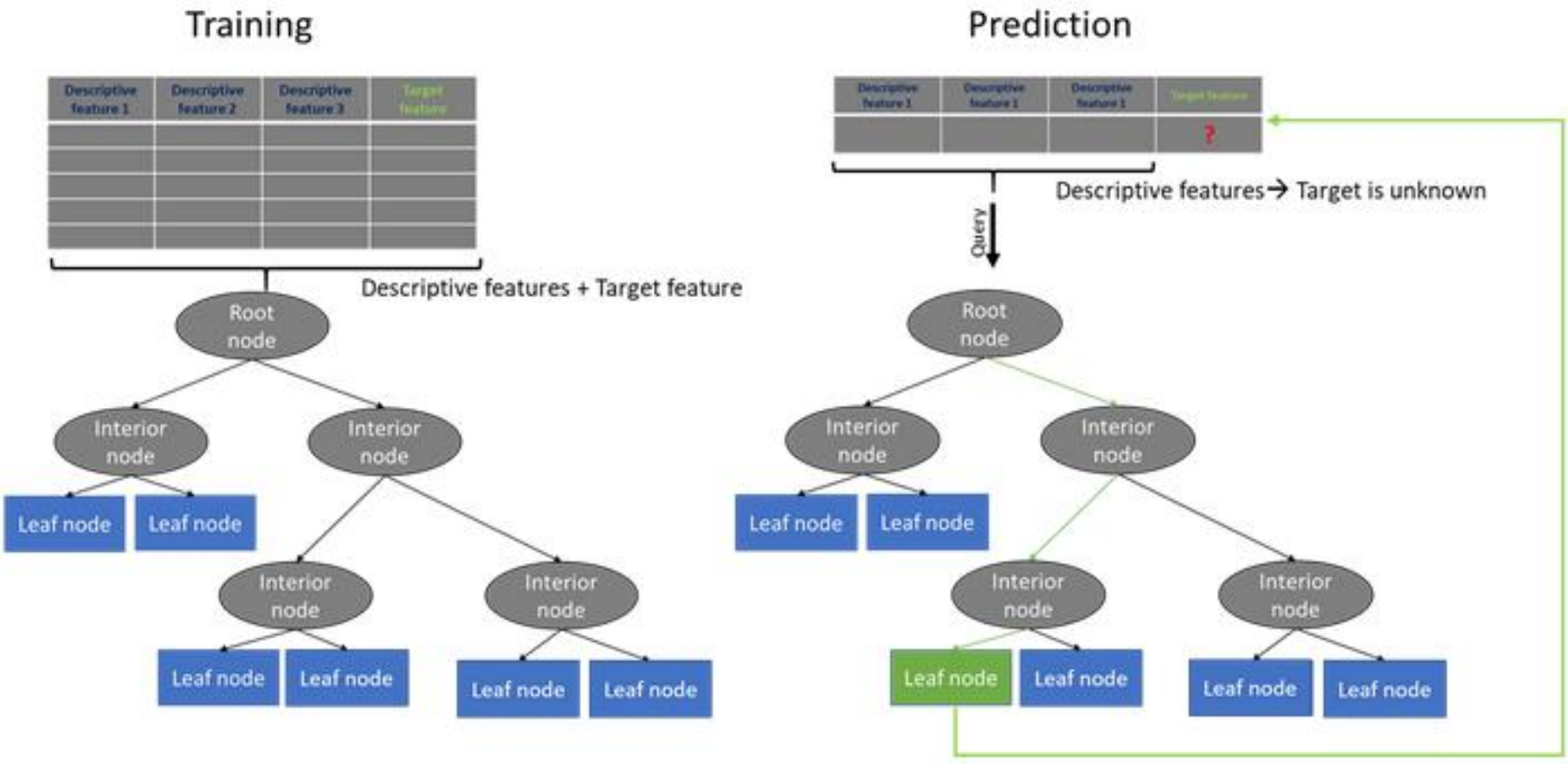
Assign Cheat to "No"

Once a decision tree is built, it is applied to any test to classify it.

How to build a Decision Tree?

1. Present a dataset containing of a number of training instances characterized by a number of descriptive features and a target feature
2. Train the decision tree model by using a measure of information gain during the training process
3. Grow the tree until a stopping criteria --> create leaf nodes which represent the *predictions* we want to make for new query instances
4. Show query instances to the tree and run down the tree until we arrive at leaf nodes
5. DONE

How to build a Decision Tree?



Building Decision Tree

- There are exponentially many decision tree that can be constructed from a given database (also called training data)

- Two approaches are known
  - **Greedy strategy**
    - A top-down recursive divide-and-conquer
  - **Modification of greedy strategy**
    - ID3
    - C4.5
    - CART, etc.

ID3: Decision Tree Induction Algorithms

- Quinlan [1986] introduced the ID3, a popular short form of Iterative Dichotomizer 3 for decision trees from a set of training data
- In ID3, each node corresponds to a splitting attribute and each arc is a possible value of that attribute
- At each node, the splitting attribute is selected to be the most informative among the attributes not yet considered in the path starting from the root

ID3: Decision Tree Induction Algorithms

- In ID3, entropy is used to measure how informative a node is
- ID3 algorithm defines a measurement of a splitting called Information Gain to determine the goodness of a split

Concept of Entropy

- The entropy of a dataset is used to measure the the "information content" in messages

- The most prominent ones are the:
  - *Gini Index, Chi-Square, Information gain ratio, Variance*

$$H(x) = -\sum_{for\ k\ \in target} (P(x=k) * log_2(P(x=k)))$$

where,

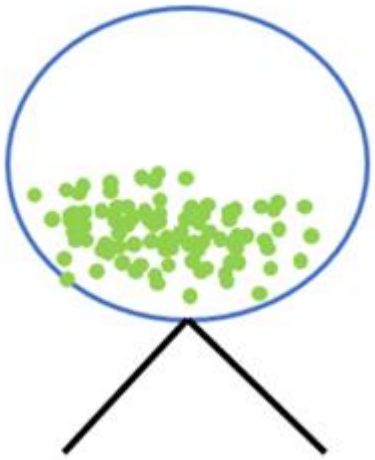
P(x=k) is the probability, that the target feature takes a specific value k.

By using *Entropy concept*, it is to build a better decision tree.

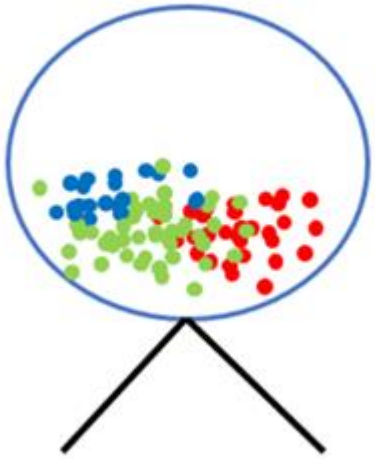
CEIT
Think smart, be different.

Concept of Entropy

Totally pure



More impure



How you can measure the same?

$$H(x) = -\sum_{for\ k\ \in target} (P(x=k) * log_2(P(x=k)))$$

Green balls: $H(x = green) = 0.5 * log_2(0.5) = -0.5$
Blue balls: $H(x = blue) = 0.2 * log_2(0.2) = -0.464$
Red balls: $H(x = red) = 0.3 * log_2(0.3) = -0.521$

H(x): $H(x) = -((-0.5) + (-0.464) + (-0.521)) = 1.485$

CEIT
Think smart, be different.

Information Gain

- The information gain of a feature is the measure of how good a descriptive feature is suited to split a dataset on.
- The term **information gain** as a measure of "informativeness" of a feature.

$$InfoGain(feature_d) = Entropy(D) - Entropy(feature_d)$$

Information Gain Calculation

Training set: $D_1(\text{Age} = 1)$

Age	Eye-sight	Astigmatism	Use type	Class
1	1	1	1	3
1	1	1	2	2
1	1	2	1	3
1	1	2	2	1
1	2	1	1	3
1	2	1	2	2
1	2	2	1	3
1	2	2	2	1

$$E(D_1) = -\frac{2}{8}log_2(\frac{2}{8}) - \frac{2}{8}log_2(\frac{2}{8}) - \frac{4}{8}log_2(\frac{4}{8}) = \mathbf{1.5}$$

$$E_{Age}(D_1) = \frac{8}{24} \times 1.5 = \mathbf{0.5000}$$

Information Gain Calculation

Training set: $D_2(\text{Age} = 2)$

Age	Eye-sight	Astigmatism	Use type	Class
2	1	1	1	3
2	1	1	2	2
2	1	2	1	3
2	1	2	2	1
2	2	1	1	3
2	2	1	2	2
2	2	2	1	3
2	2	2	2	3

$$E(D_2) = -\frac{1}{8}log_2(\frac{1}{8}) - \frac{2}{8}log_2(\frac{2}{8}) - \frac{5}{8}log_2(\frac{5}{8})$$
$$= 1.2988$$

$$E_{Age}(D_2) = \frac{8}{24} \times 1.2988 = 0.4329$$

Information Gain Calculation

Training set: D_3(Age = 3)

Age	Eye-sight	Astigmatism	Use type	Class
3	1	1	1	3
3	1	1	2	3
3	1	2	1	3
3	1	2	2	1
3	2	1	1	3
3	2	1	2	2
3	2	2	1	3
3	2	2	2	3

$$E(D_3) = -\frac{1}{8}log_2(\frac{1}{8}) - \frac{1}{8}log_2(\frac{1}{8}) - \frac{6}{8}log_2(\frac{6}{8}) = \mathbf{1.0613}$$

$$E_{Age}(D_3) = \frac{8}{24} \times 1.0613 = \mathbf{0.3504}$$

$$\alpha\,(Age, D) = 1.3261 - (0.5000 + 0.4329 + 0.3504) = \mathbf{0.0394}$$

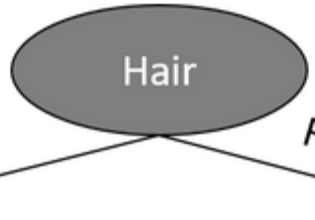
CEIT
Think smart, be different.

Example: Decision Tree Method using ID3

Zoo data set

	toothed	hair	breathes	legs	species
0	True	True	True	True	Mammal
1	True	True	True	True	Mammal
2	True	False	True	False	Reptile
3	False	True	True	True	Mammal
4	True	True	True	True	Mammal
5	True	True	True	True	Mammal
6	True	False	False	False	Reptile
7	True	False	True	False	Reptile
8	True	True	True	True	Mammal
9	False	False	True	True	Reptile

Our data set should only contain "Mammals" or "Reptiles“ (target variable)
How can find the best "way" to split the dataset?



	toothed	hair	breathes	legs	species
0	True	True	True	True	Mammal
1	True	True	True	True	Mammal
3	False	True	True	True	Mammal
4	True	True	True	True	Mammal
5	True	True	True	True	Mammal
8	True	True	True	True	Mammal

	toothed	hair	breathes	legs	species
2	True	False	True	False	Reptile
6	True	False	False	False	Reptile
7	True	False	True	False	Reptile
9	False	False	True	True	Reptile

Lets apply this approach to our original dataset where we want to predict the animal species. Our dataset has two target feature values in its target feature value space {Mammal, Reptile}. Where $P(x = Mammal) = 0.6$ and $P(x = Reptile) = 0.4$ Hence the entropy of our dataset regarding the target feature is calculated with:

$$H(x) = -((0.6 * log_2(0.6)) + (0.4 * log_2(0.4))) = 0.971$$

So where are we now creating a tree model?

We have now determined the total impurity/purity (≈ entropy) of our dataset which equals to approximately 0.971.

Now our task is to find the best feature in terms of information gain which serves as root node.

The formula for the Information Gain calculation per feature is:

$$InforGain(feature_d, D) = Entropy(D) - \sum_{t \in feature} (\frac{|feature_d = t|}{|D|} * H(feature_d = t))$$

$$=$$

$$Entropy(D) - \sum_{t \in feature} (\frac{|feature_d = t|}{|D|} * (- \sum_{k \in target} (P(target = k, feature_d = t) * log_2(P(target = k, feature_d = t)))))$$

	toothed	breathes	legs	species
0	True	True	True	Mammal
1	True	True	True	Mammal
2	True	True	False	Reptile
3	False	True	True	Mammal
4	True	True	True	Mammal
5	True	True	True	Mammal
6	True	False	False	Reptile
7	True	True	False	Reptile
8	True	True	True	Mammal
9	False	True	True	Reptile

toothed == True

toothed == False

After computing the IG of feature *toothed* do this for features *breathes* and *legs*

	toothed	breathes	legs	species
0	True	True	True	Mammal
1	True	True	True	Mammal
2	True	True	False	Reptile
4	True	True	True	Mammal
5	True	True	True	Mammal
6	True	False	False	Reptile
7	True	True	False	Reptile
8	True	True	True	Mammal

	toothed	breathes	legs	species
3	False	True	True	Mammal
9	False	True	True	Reptile

1. Calculate the entropy for toothed == True
2. Calculate the entropy for toothed == False
3. Sum up the entropies of 1. and 2.
4. Subtract this sum from the whole datasets entropy → InfoGain

Now we will calculate the Information gain for each descriptive feature:

toothed:

Drawing= **0.963547**

$InfoGain(toothed) = 0.971 - 0.963547 =$ **0.00745**

breathes:

$H(breathes) = (\frac{9}{10} * -((\frac{6}{9} * log_2(\frac{6}{9})) + (\frac{3}{9} * log_2(\frac{3}{9}))) + \frac{1}{10} * -((0) + (1 * log_2(1)))) =$ **0.82647**

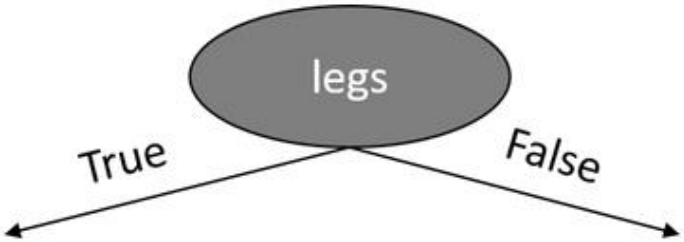
$InfoGain(breathes) = 0.971 - 0.82647 =$ **0.1445**

legs:

$H(legs) = \frac{7}{10} * -((\frac{6}{7} * log_2(\frac{6}{7})) + (\frac{1}{7} * log_2(\frac{1}{7}))) + \frac{3}{10} * -((0) + (1 * log_2(1))) =$**0.41417**

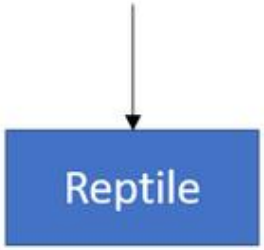
$InfoGain(legs) = 0.971 - 0.41417 =$ **0.5568**

Hence, the splitting the dataset along the feature *legs* results in the largest information gain.
We should use this feature for our root node.



	toothed	breathes	species
0	True	True	Mammal
1	True	True	Mammal
3	False	True	Mammal
4	True	True	Mammal
5	True	True	Mammal
8	True	True	Mammal
9	False	True	Reptile

	toothed	breathes	species
2	True	True	Reptile
6	True	False	Reptile
7	True	True	Reptile



We see that for legs == False, the target feature values of the remaining dataset are all *Reptile* and hence we set this as leaf node because we have a pure dataset.
Until now we have found the feature for the root node as well as a leaf node for *legs == False*. The same steps for information gain calculation must now be accomplished also for the remaining dataset for *legs == True* since here we still have a mixture of different target feature values.

Information gain calculation for the features *toothed* and *breathes* for the remaining dataset *legs* == *True*:

Entropy of the (new) sub data set after first split:

Information gain calculation for the features *toothed* and *breathes* for the remaining dataset *legs* == *True*:

Entropy of the (new) sub data set after first split:

$$H(D) = -((\frac{6}{7} * log_2(\frac{6}{7})) + (\frac{1}{7} * log_2(\frac{1}{7}))) = \mathbf{0.5917}$$

toothed:

$$H(toothed) = \frac{5}{7} * -((1 * log_2(1)) + (0)) + \frac{2}{7} * -((\frac{1}{2} * log_2(\frac{1}{2})) + (\frac{1}{2} * log_2(\frac{1}{2}))) = \mathbf{0.285}$$

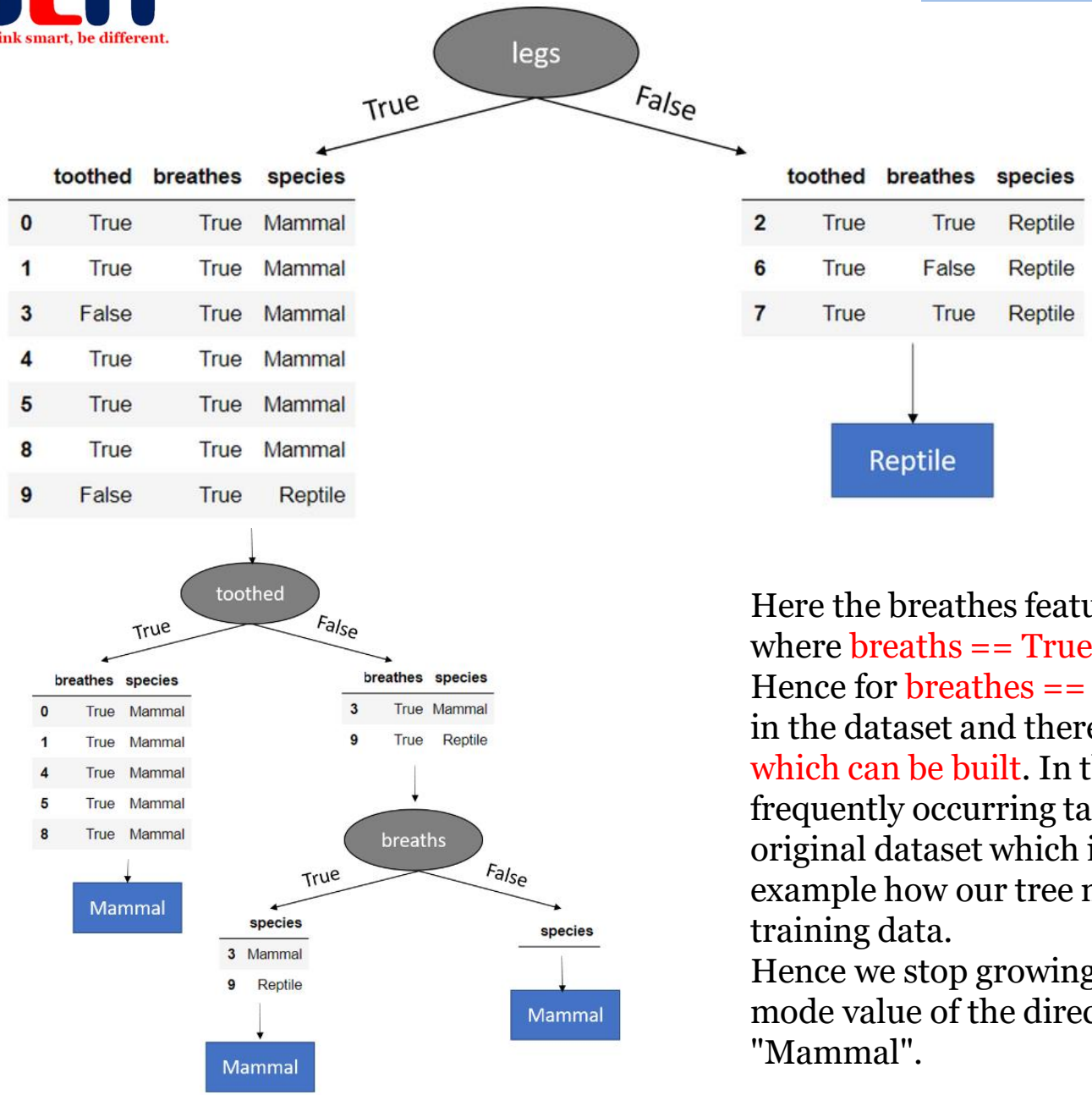
$$InfoGain(toothed) = 0.5917 - 0.285 = \mathbf{0.3067}$$

breathes:

$$H(breathes) = \frac{7}{7} * -((\frac{6}{7} * log_2(\frac{6}{7})) + (\frac{1}{7} * log_2(\frac{1}{7}))) + 0 = \mathbf{5917}$$

$$InfoGain(toothed) = 0.5917 - 0.5917 = \mathbf{0}$$

The dataset for toothed == False still contains a mixture of different target feature values why we proceed partitioning on the last left feature (== breathes)

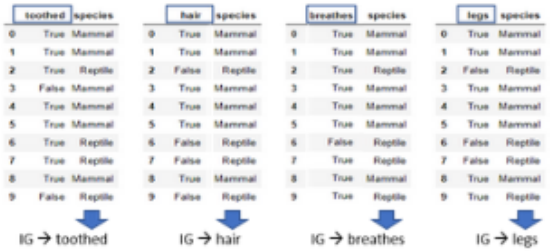


Here the breathes feature solely contains data where breaths == True.
Hence for breathes == False there are no instances in the dataset and therewith there is no *sub-Dataset* which can be built. In that case we return the most frequently occurring target feature value in the original dataset which is *Mammal*. This is an example how our tree model generalizes behind the training data.
Hence we stop growing the tree and return the mode value of the direct parent node which is "Mammal".

1. Create a root node

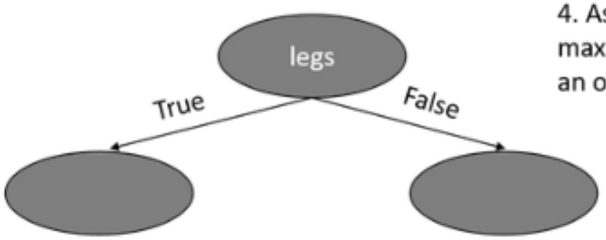
Entropy

2. Calculate the entropy of the whole (sub) dataset



Select: max(IG_feature)

3. Calculate the Information gain of each single feature and pick that feature with the larges Information gain



4. Assign the (root) node the label of the feature with the maximum information gain. Grow for each feature value an outgoing branch and add unlabelled nodes at the end

Legs == True

	toothed	hair	breathes	legs	species
0	True	True	True	True	Mammal
1	True	True	True	True	Mammal
3	False	True	True	True	Mammal
4	True	True	True	True	Mammal
5	True	True	True	True	Mammal
8	True	True	True	True	Mammal
9	False	False	True	True	Reptile

Legs == False

	toothed	hair	breathes	legs	species
2	True	False	True	False	Reptile
6	True	False	False	False	Reptile
7	True	False	True	False	Reptile

5. Split the dataset along the values of the maximum information gain feature and remove this feature from the dataset

First sub tree

Second sub tree

6. For each of the sub_datasets, repeat steps 2 to 5 until a stopping criteria is satisfied →Here the recursion kicks in

Advantages

- Can generate understandable rules
- perform classification without much computation
- can handle continuous and categorical variables
- provide a clear indication of which fields are most important for prediction or classification

Disadvantages

- Not suitable for prediction of continuous attribute
- Perform poorly with many class and small data

Next Week Lecture

- Supervised Learning: Classification with Artificial Neural Network (ANN)

Thank You