McMaster University ENGINEERING

W BOOTH SCHOOL OF ENGINEERING PRACTICE AND TECHNOLOGY

SEP 785: Machine Learning

Lecture 3: Decision Trees

Instructor: Dr. Dalia Mahmoud, PhD
(Mechanical Engineering, McMaster University)
Email: mahmoudd@mcmaster.ca

https://forms.office.com/r/QTeeWHaTPg?origin=lprLink

Assignment

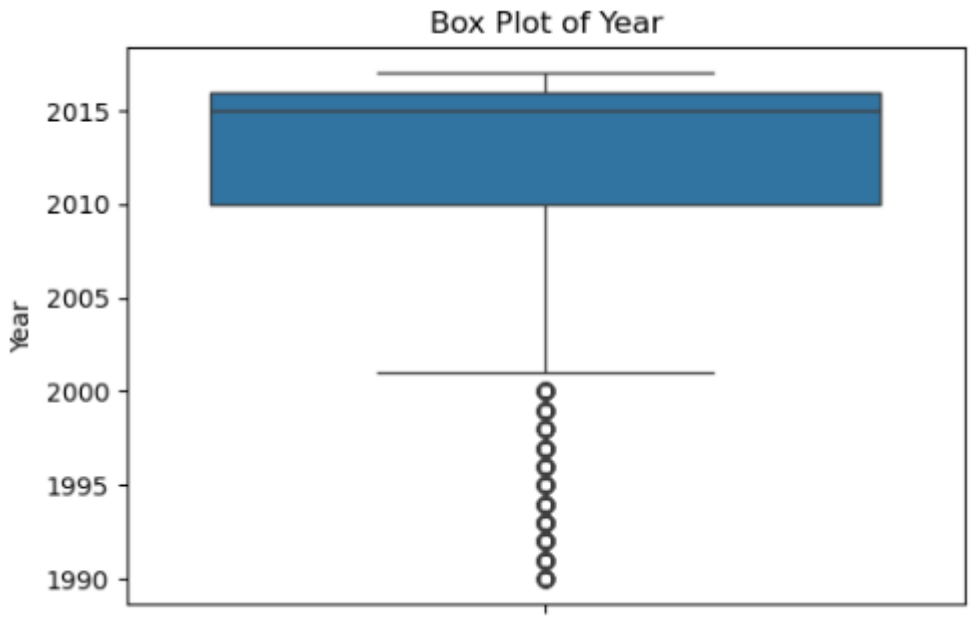

```
from scipy.spatial.distance import pdist, squareform

# Compute pairwise Euclidean distances between rows
distances = pdist(df, metric='euclidean')

# Convert to a square matrix for easier interpretation
distance_matrix = squareform(distances)

print("Euclidean Distance Matrix:\n", distance_matrix)
```

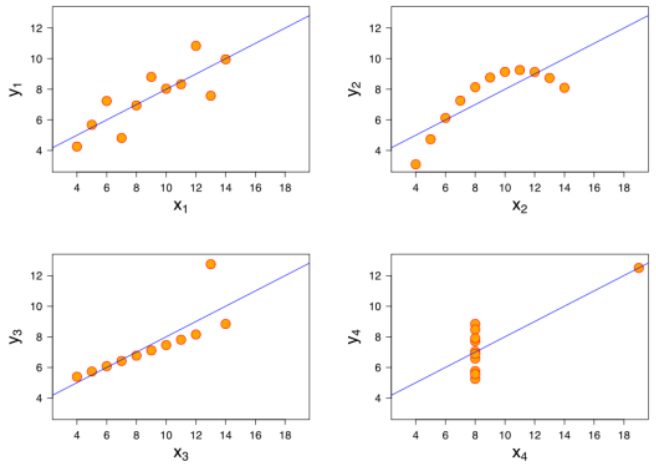
```
Euclidean Distance Matrix:
 [[  0.          50.0999002  100.42410069 ...  47.04253395  91.7224073
  112.70314991]
 [ 50.0999002    0.          50.36864104 ...  66.55073253  54.74486277
   70.38465742]
 [100.42410069  50.36864104   0.         ... 108.21275341  51.14684741
   47.02127178]
 ...
 [ 47.04253395  66.55073253 108.21275341 ...   0.          76.17086057
   99.52386648]
 [ 91.7224073   54.74486277  51.14684741 ...  76.17086057   0.
   23.55843798]
 [112.70314991  70.38465742  47.02127178 ...  99.52386648  23.55843798
    0.        ]]
```



Recap

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

EDA



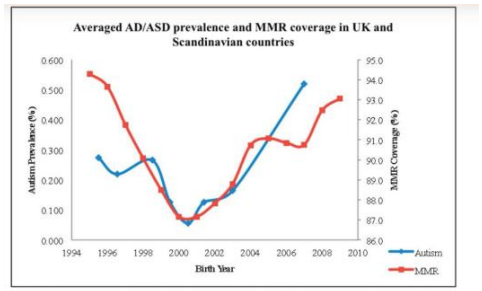
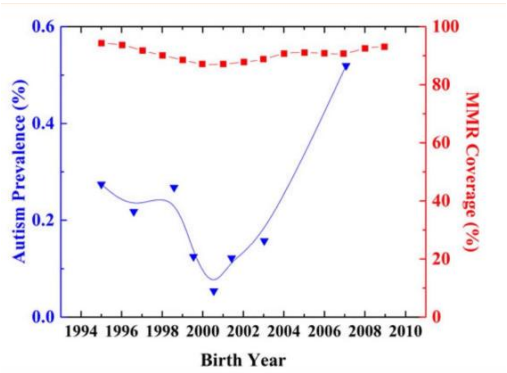


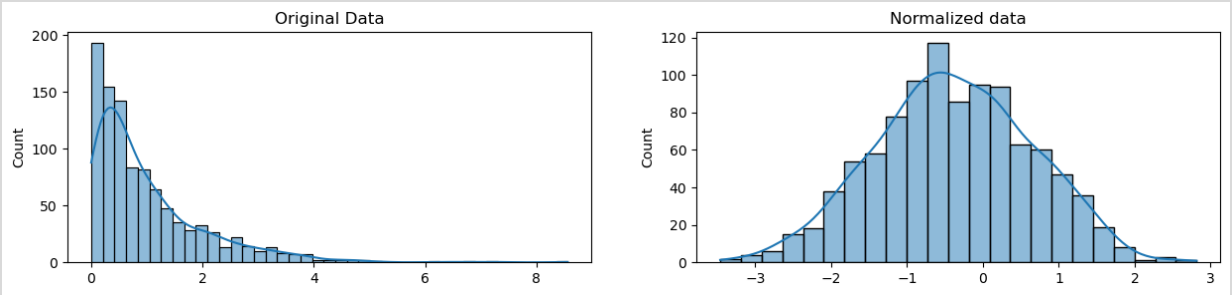
Figure 1-Averaged AD/ASD prevalence and MMR coverage in UK, Norway and Sweden. Both MMR and AD/ASD data are normalized to the maximum coverage/prevalence during the time period of this analysis.

Diesher *et al.* 2015 *Issues in Law and Medicine*



Matt Carey via sciencebasedmedicine.org

Data Preprocessing



Index	Animal
0	Dog
1	Cat
2	Sheep
3	Horse
4	Lion

One-Hot code

Index	Dog	Cat	Sheep	Lion	Horse
0	1	0	0	0	0
1	0	1	0	0	0
2	0	0	1	0	0
3	0	0	0	0	1
4	0	0	0	1	0

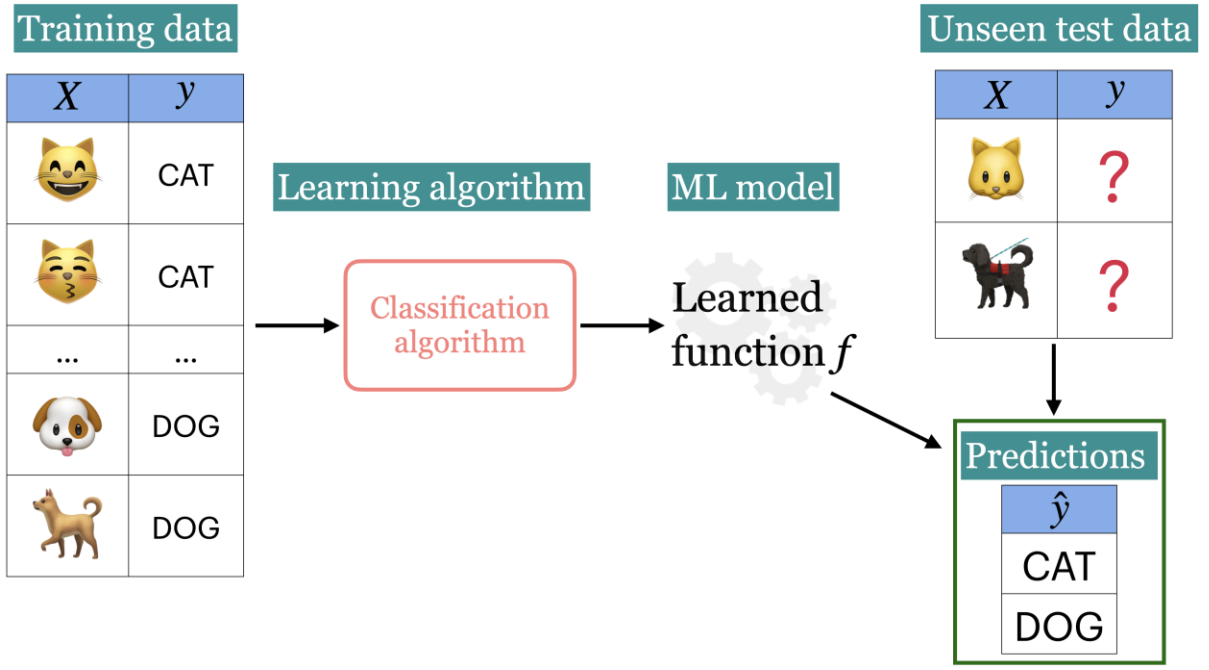
Intended Learning Outcomes

1. Define baseline models and explain their appropriate use cases.
2. Describe the mathematical algorithms underlying decision trees.
3. Utilize functions from the Scikit-learn library to analyze data using decision tree models.
4. Explain the core concept of generalization in machine learning models.
5. Apply data-splitting techniques effectively to prepare datasets for modeling.
6. Highlight the significance of cross-validation in machine learning workflows.
7. Identify and address overfitting and underfitting in machine learning models.

Contents

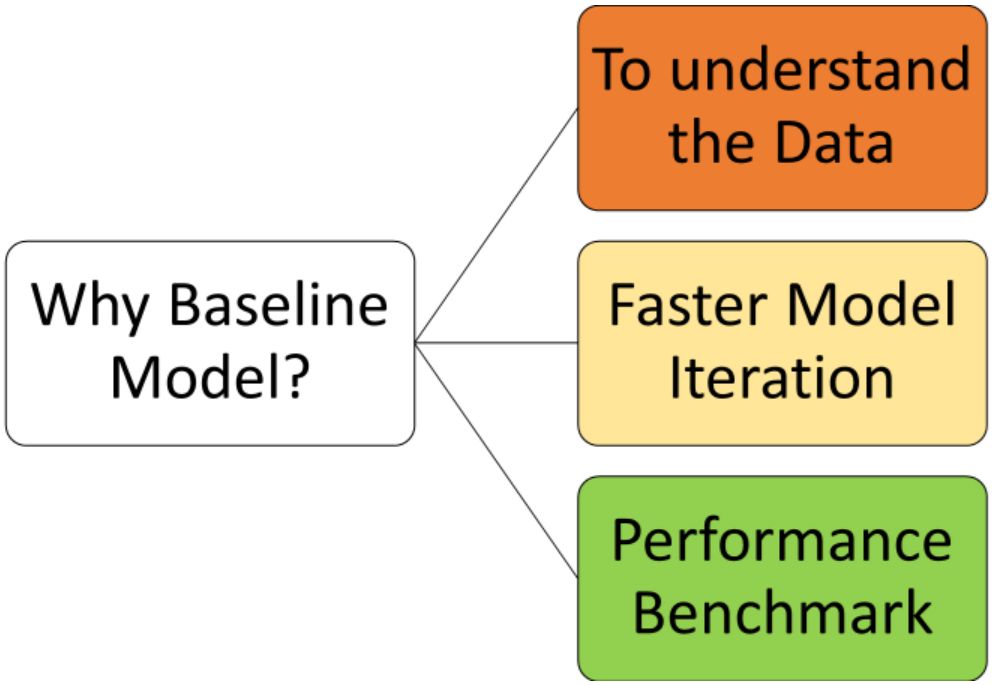
Baseline Models

- Is a simple machine learning algorithm based on simple rules of thumb.
- Baselines provide a way to sanity check your machine learning model.
- Purpose:
  - Establishes the minimum acceptable performance level.
  - Helps evaluate the added value of a sophisticated model.



Why Use Baselines?

- Performance Benchmark: Ensures your model outperforms trivial solutions.
- Insights:
  - Highlights data imbalances.
  - Reveals potential overfitting or underfitting in advanced models.
- Debugging: Provides a simple, reliable result to test data pipelines and preprocessing.



Types of Baseline Models

- Heuristic Baselines: Predict a constant value (e.g., the mean, median, or most frequent class).
- Simple ML Models: Linear regression, logistic regression, or decision trees with minimal tuning.
- Dummy Models: Provided by libraries like Scikit-learn, e.g., DummyClassifier and DummyRegressor.

Baseline Models

- **Advantages of Baseline Models**
  - Simple to implement and interpret.
  - Quick to compute, requiring minimal resources.
  - Highlights whether the dataset provides enough signal for predictive modeling.
- **Limitations of Baselines**
  - Cannot capture complex patterns or relationships.
  - May not always represent realistic use cases (e.g., random predictions).
- **When to Use Baselines**
  - At the start of a machine learning project to set expectations.
  - To validate the performance of advanced models.
  - As part of experimentation and testing pipelines.

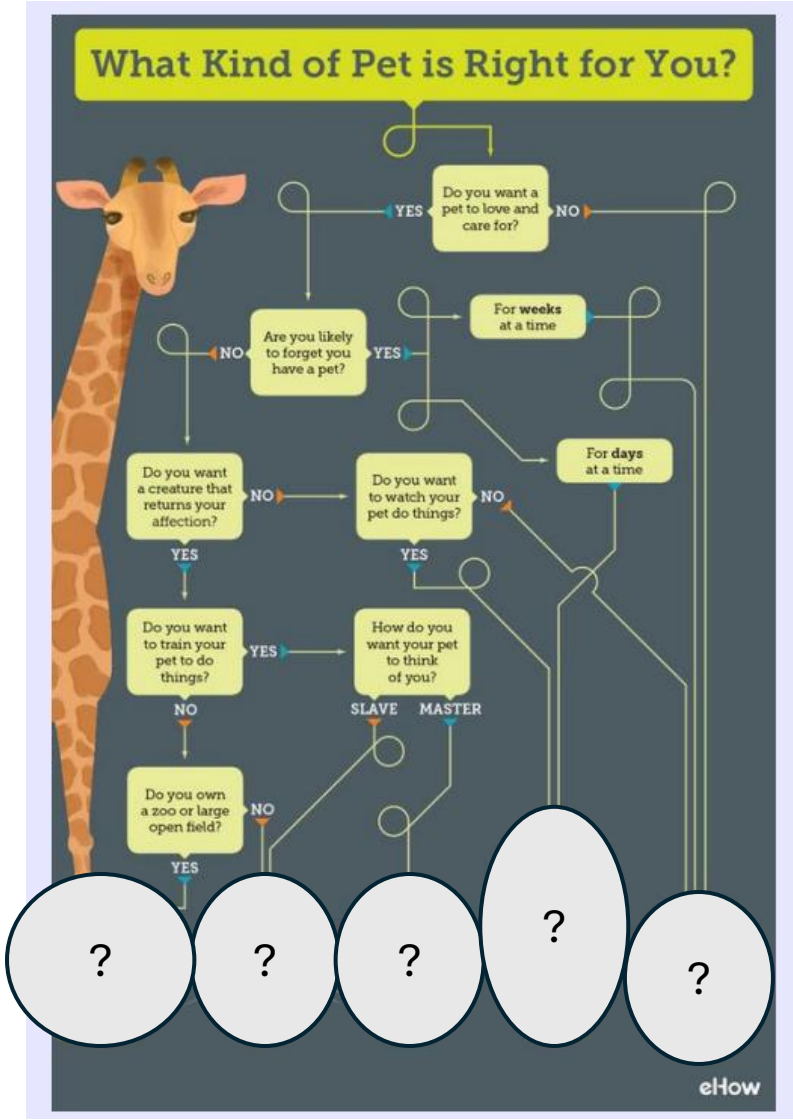
Baseline Models

- Example python script

Contents

What Kind of Pet is Right for you ?

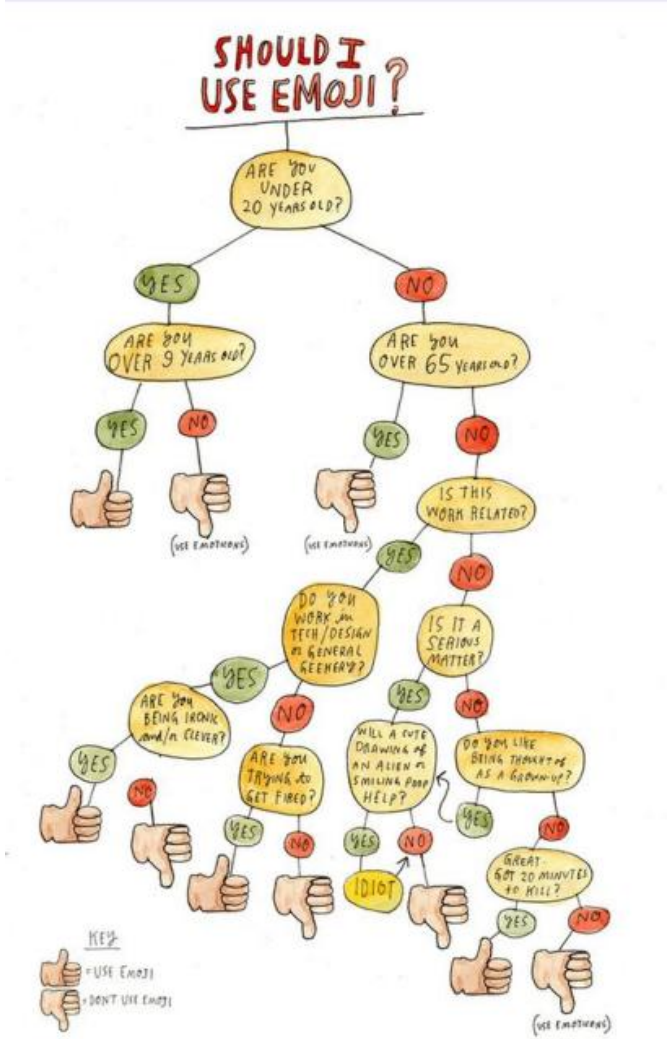
McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology



Volunteer ?

Should you use emoji in a conversation?

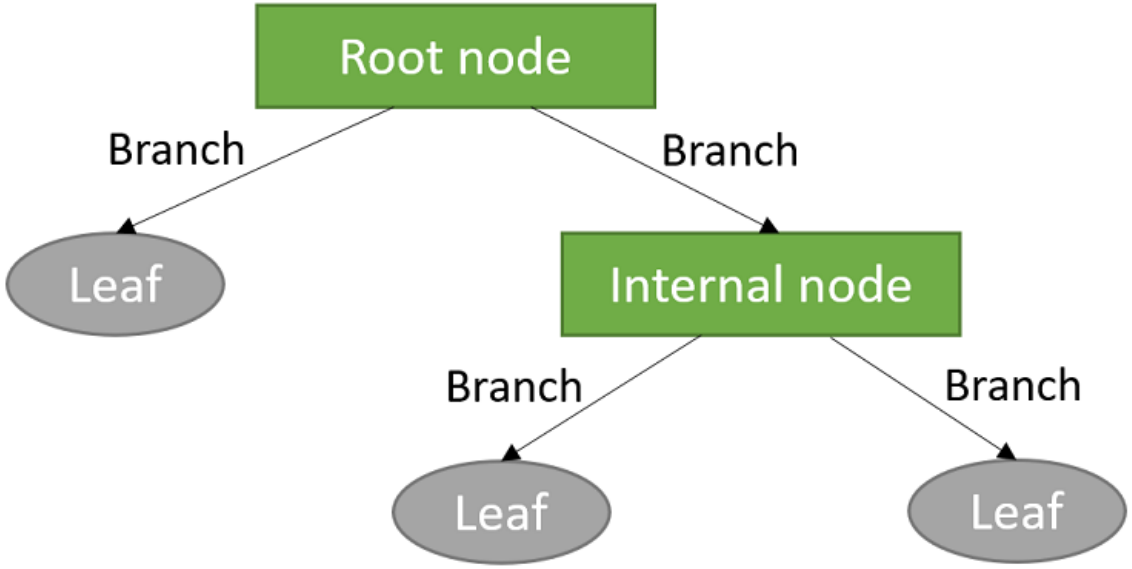
McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology



What is a Decision Tree

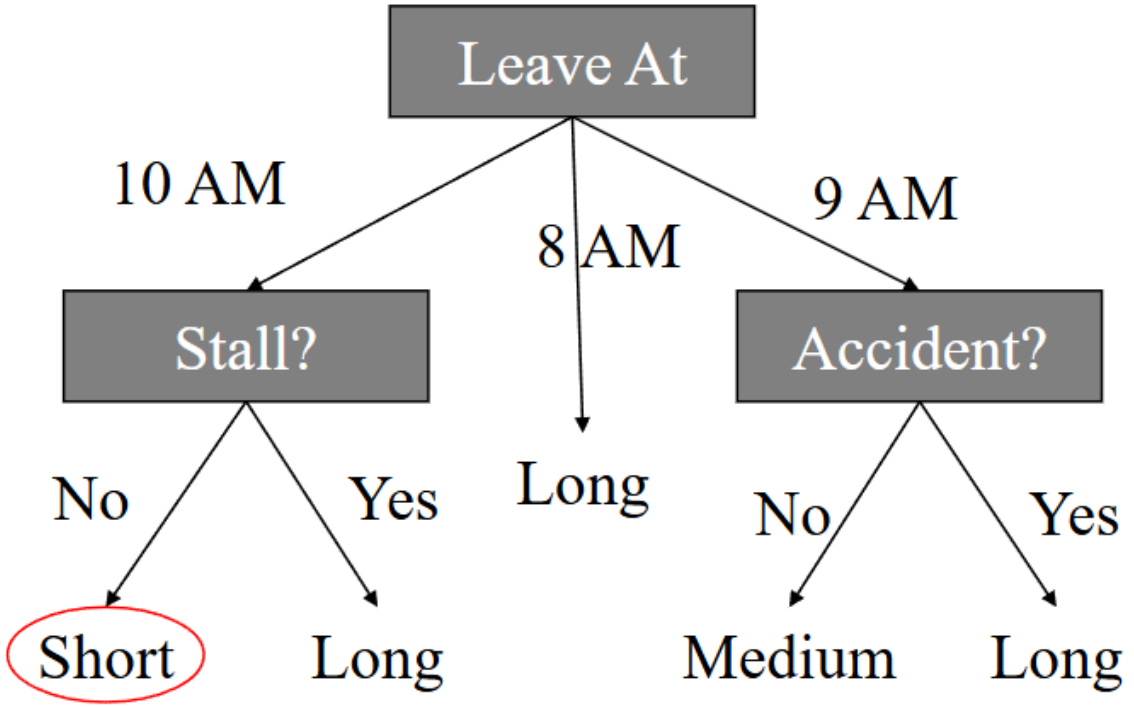
McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

- A decision tree is a **simple model** for **supervised** classification/regression.
- Each internal node performs a **Boolean test** on an input feature.
- Each **leaf node** specifies a value for the target feature.



How to Create a Decision Tree

1. We first make a **list of attributes** that we can measure.
2. These attributes can be **discrete or continuous.**
3. We then choose a **target attribute** that we want to predict.
4. Then **gather data** that lists what we have seen in the past.



If we leave at 10 AM and there are no cars stalled on the road, what will our commute time be?

How to Create a Decision Tree

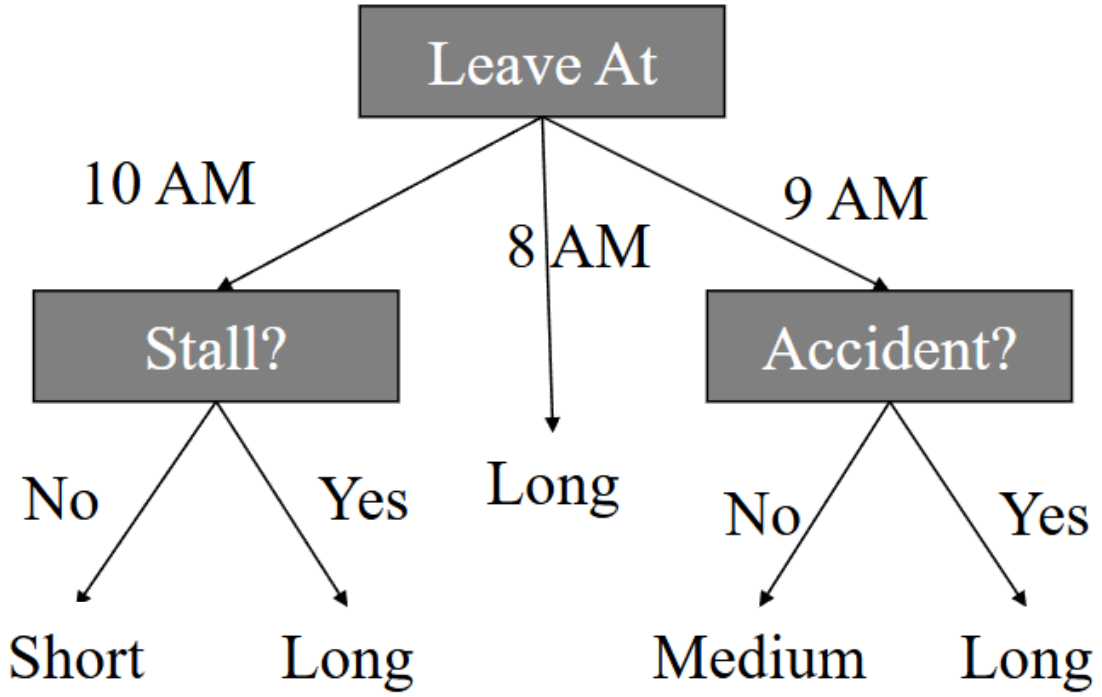
McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

1. We first make a **list of attributes** that we can measure.
2. These attributes can be **discrete or continuous.**
3. We then choose a **target attribute** that we want to predict.
4. Then **gather data** that lists what we have seen in the past.

Example	Attributes				Target
	Hour	Weather	Accident	Stall	Commute
D1	8 AM	Sunny	No	No	Long
D2	8 AM	Cloudy	No	Yes	Long
D3	10 AM	Sunny	No	No	Short
D4	9 AM	Rainy	Yes	No	Long
D5	9 AM	Sunny	Yes	Yes	Long
D6	10 AM	Sunny	No	No	Short
D7	10 AM	Cloudy	No	No	Short
D8	9 AM	Rainy	No	No	Medium
D9	9 AM	Sunny	Yes	No	Long
D10	10 AM	Cloudy	Yes	Yes	Long
D11	10 AM	Rainy	No	No	Short
D12	8 AM	Cloudy	Yes	No	Long
D13	9 AM	Sunny	No	No	Medium

Choosing Attributes

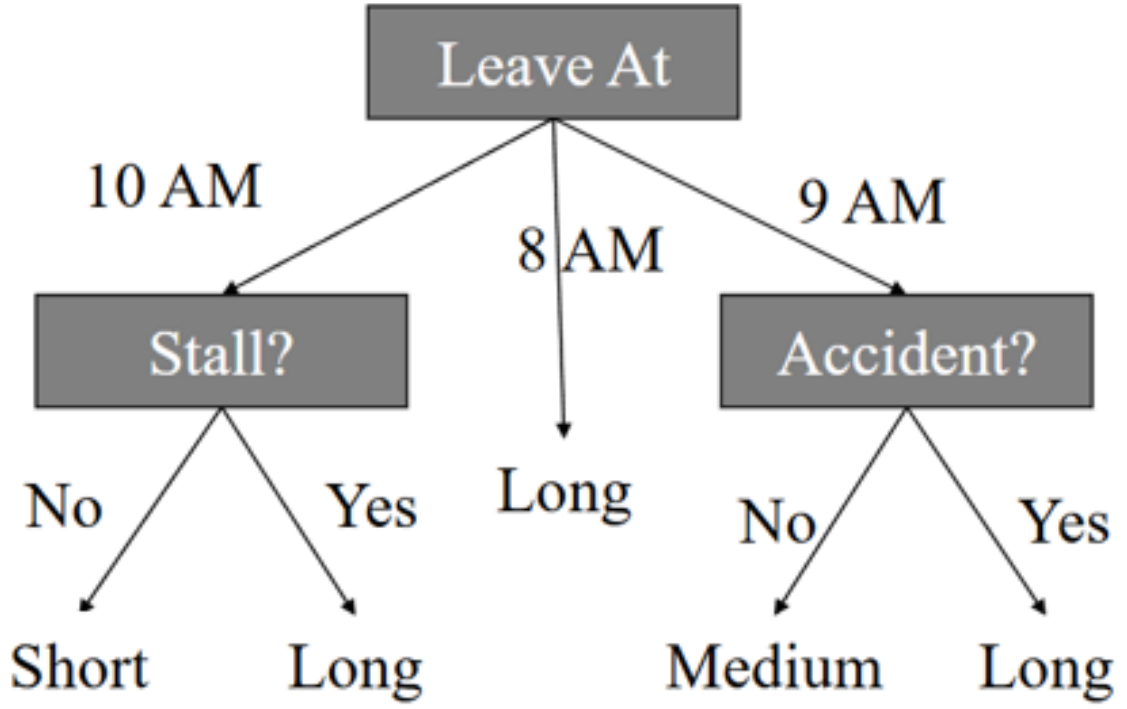
- The previous experience decision table showed **4 attributes**:
  - hour,
  - weather,
  - accident
  - and stall
- But the decision tree only showed **3 attributes**: hour, accident and stall
- Why is that?



Choosing Attributes

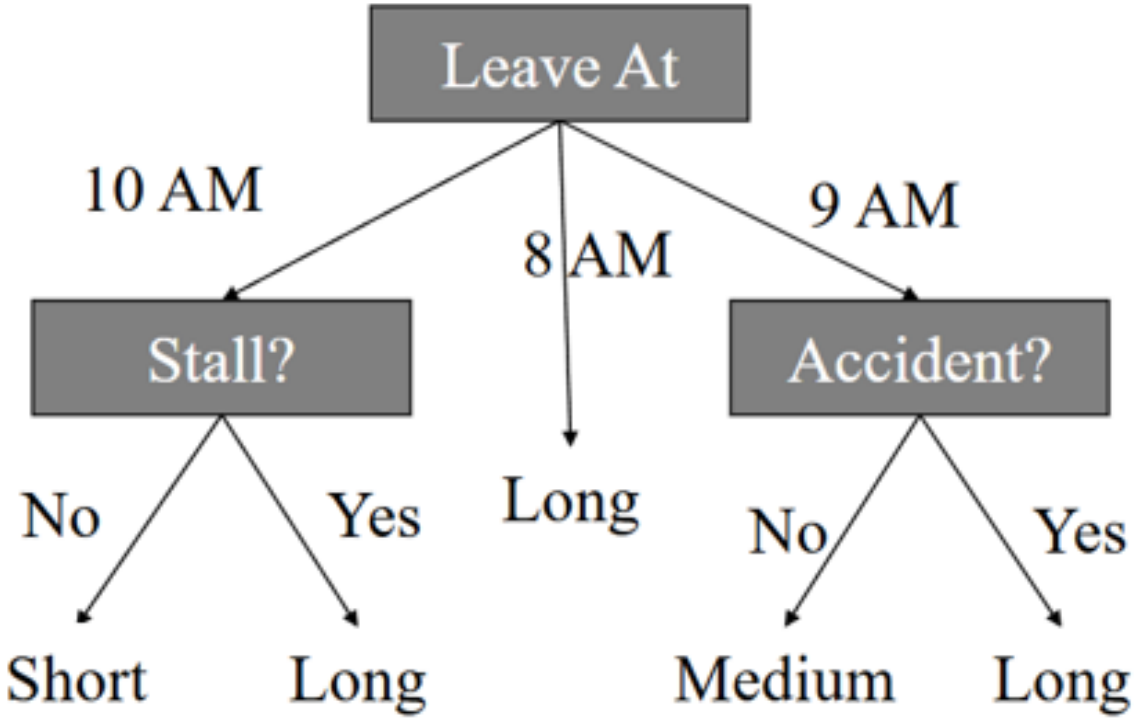
- Methods for selecting attributes (which will be described later) show that **weather** is not a **discriminating** attribute
- We use the principle of **Occam's Razor**: The simplest explanation or solution is often the best.

Occam's Razor is a philosophical principle stating:
"Entities should not be multiplied beyond necessity."



Choosing Attributes

- There are **different algorithms** that can be used in constructing a decision tree.
- The difference lies in **how we select** the attributes for the tree.
- We will focus on the **ID3 algorithm** developed by Ross Quinlan in 1975



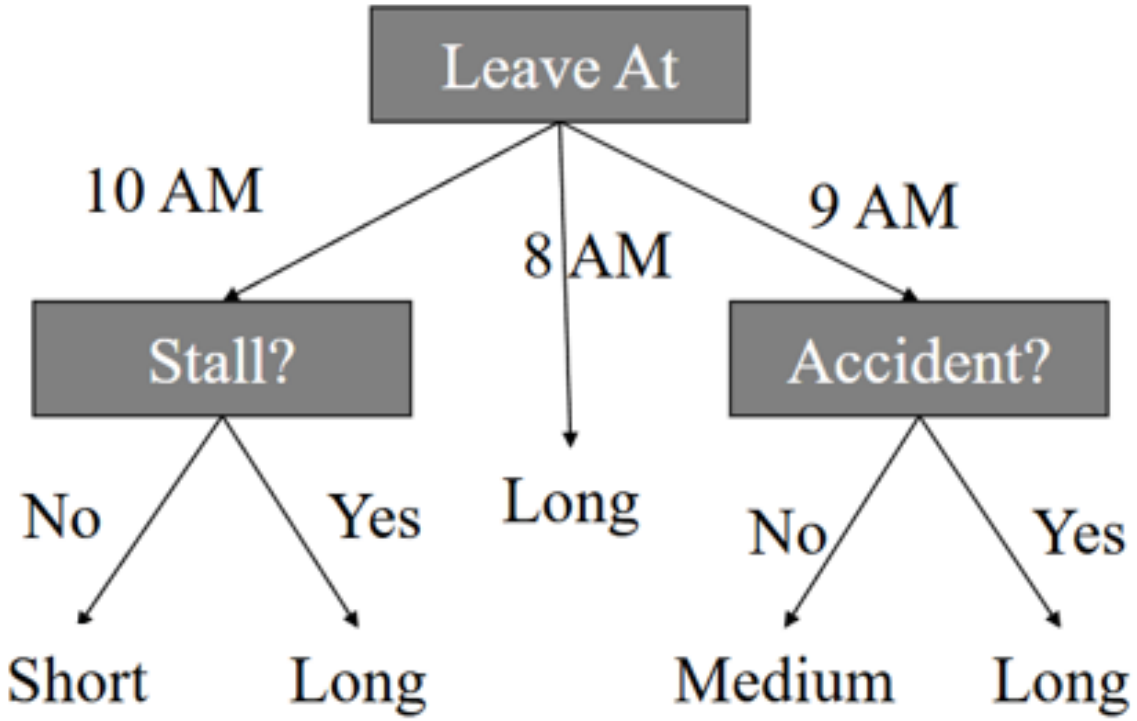
Basic Idea

- Choose the **best attribute(s)** to split the remaining instances and make that attribute a decision node.
- **Repeat** this process for recursively for **each child**.
- Stop when:
  - All the **instances** have the same **target attribute** value.
  - There are **no more attributes.**
  - There are **no more instances.**

Example	Attributes				Target
	Hour	Weather	Accident	Stall	Commute
D1	8 AM	Sunny	No	No	Long
D2	8 AM	Cloudy	No	Yes	Long
D3	10 AM	Sunny	No	No	Short
D4	9 AM	Rainy	Yes	No	Long
D5	9 AM	Sunny	Yes	Yes	Long
D6	10 AM	Sunny	No	No	Short
D7	10 AM	Cloudy	No	No	Short
D8	9 AM	Rainy	No	No	Medium
D9	9 AM	Sunny	Yes	No	Long
D10	10 AM	Cloudy	Yes	Yes	Long
D11	10 AM	Rainy	No	No	Short
D12	8 AM	Cloudy	Yes	No	Long
D13	9 AM	Sunny	No	No	Medium

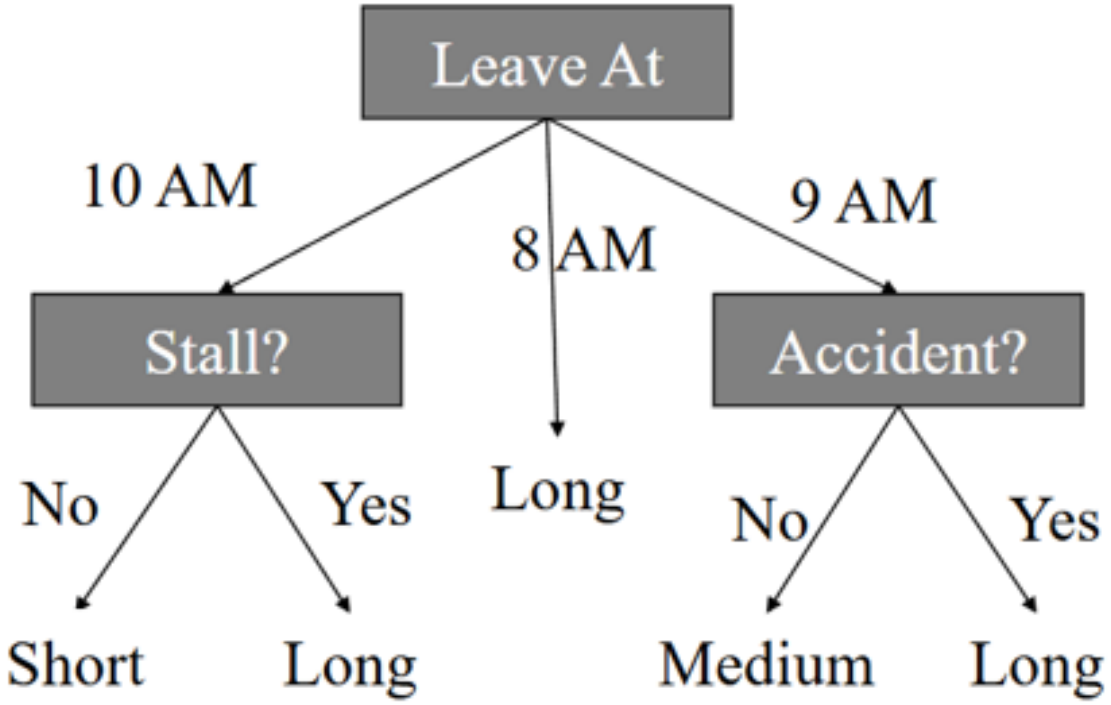
Identifying the Best Attributes

- How did we know to split on leave at and then on stall and accident and not weather?



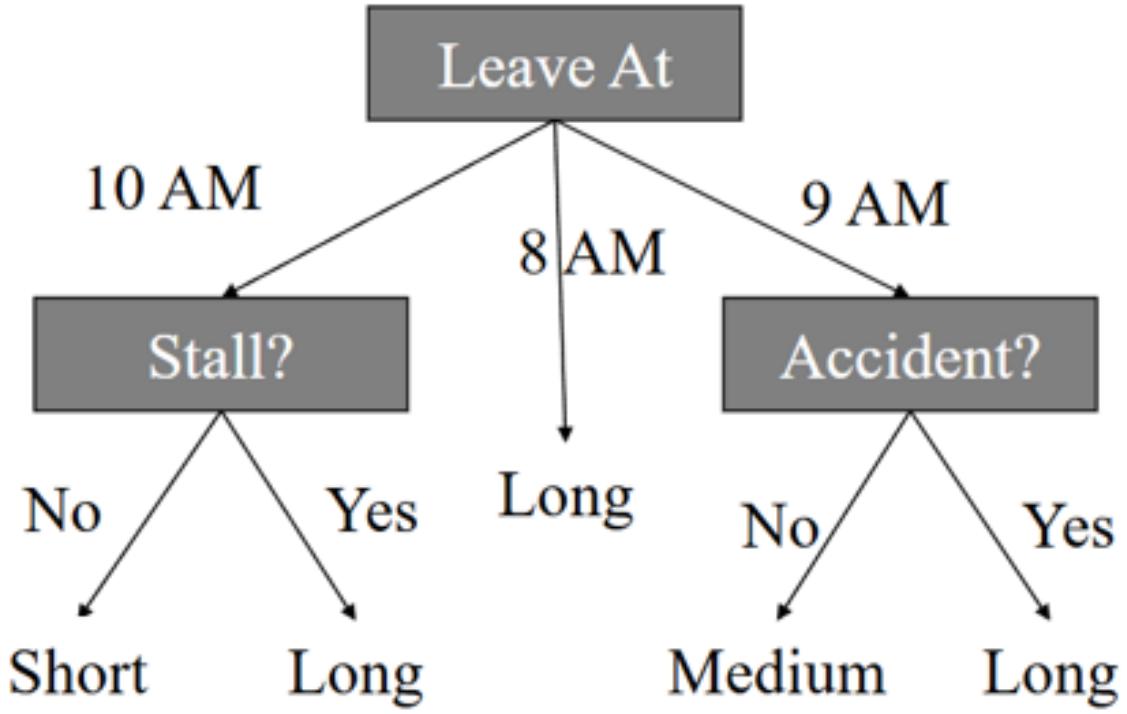
ID3 heuristic

- To determine the best attribute, we look at the **ID3 heuristic** (iterative dichotomizer 3)
- ID3 splits attributes based on their **entropy**.
- Entropy is the measure of **disorder**…
- Entropy is **minimized** when all values of the **target attribute** are the **same**.



ID3 heuristic

- If we know that commute time will always be short, then **entropy = 0.**
- **Entropy** is **maximized** when there is an equal chance of all values for the target attribute (i.e. the result is random)
- If commute time = short in 3 instances, medium in 3 instances and long in 3 instances, **entropy** is **maximized**.



Entropy

$$H(S) = -\sum_{x \in X} p(x) \log_2 p(x)$$

- S the current dataset for which entropy is being calculated.
- x is the current set of classes in S.
- p(x) is the proportion of the number of elements in class x relative to the number of elements in S.

x S

Example	Attributes				Target
	Hour	Weather	Accident	Stall	Commute
D1	8 AM	Sunny	No	No	Long
D2	8 AM	Cloudy	No	Yes	Long
D3	10 AM	Sunny	No	No	Short
D4	9 AM	Rainy	Yes	No	Long
D5	9 AM	Sunny	Yes	Yes	Long
D6	10 AM	Sunny	No	No	Short
D7	10 AM	Cloudy	No	No	Short
D8	9 AM	Rainy	No	No	Medium
D9	9 AM	Sunny	Yes	No	Long
D10	10 AM	Cloudy	Yes	Yes	Long
D11	10 AM	Rainy	No	No	Short
D12	8 AM	Cloudy	Yes	No	Long
D13	9 AM	Sunny	No	No	Medium

Information Gain

- Information Gain is the difference between uncertainty of the starting node and weighted impurity of the two child nodes. Information gain decides which feature should be used to split the data.

Information Gain = Entropy(parent) – [Average entropy(children)]

$$\text{Entropy} = \sum_i -P_i(\log_2 P_i)$$

P_i is probability of class i

For Our Example

Entropy of S

$$H(S) = -(4/13)\log_2(4/13) - (2/13)\log_2(2/13) - (7/13)\log_2(7/13)$$
$$= (0.532 + 0.415 + 0.481)$$
$$= 1.42$$

$$H(Hour = 8) = -(3/3)\log_2(3/3) = 0$$
$$H(Hour = 9) = -(2/5)\log_2(2/5) - (3/5)\log_2(3/5) = 0.971$$
$$H(Hour = 10) = -(4/5)\log_2(4/5) + (1/5)\log_2(1/5) = 0.729$$

Change in Entropy

$$IG = 1.420 - (5/13)(0.971) - (5/13)(0.729) = 1.420 - 0.651 = 0.769$$

x

S

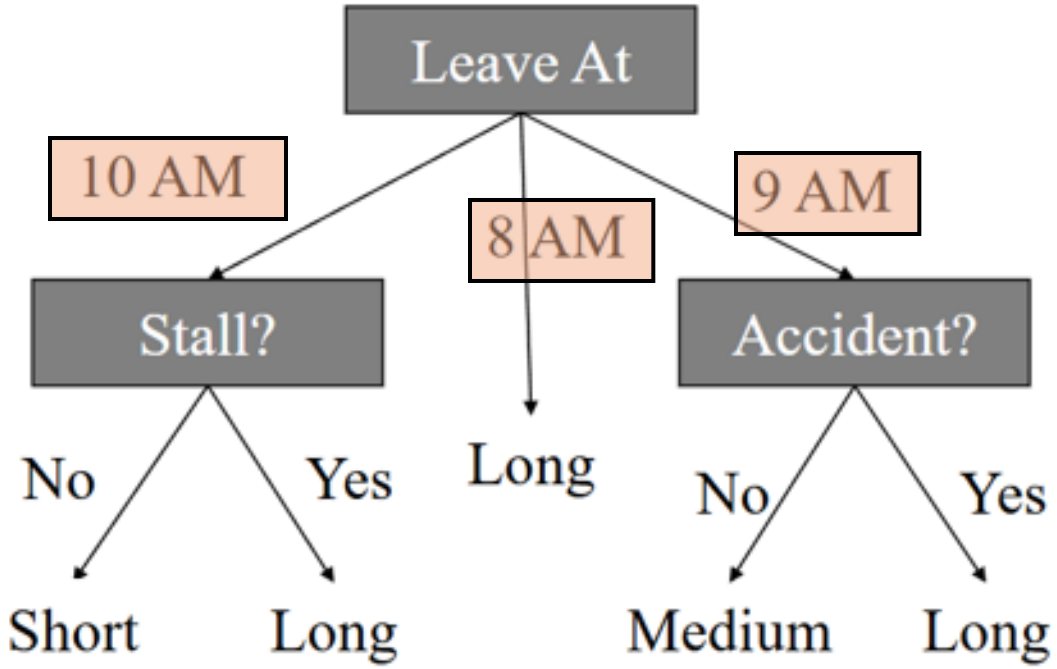
Example	Attributes				Target
	Hour	Weather	Accident	Stall	Commute
D1	8 AM	Sunny	No	No	Long
D2	8 AM	Cloudy	No	Yes	Long
D3	10 AM	Sunny	No	No	Short
D4	9 AM	Rainy	Yes	No	Long
D5	9 AM	Sunny	Yes	Yes	Long
D6	10 AM	Sunny	No	No	Short
D7	10 AM	Cloudy	No	No	Short
D8	9 AM	Rainy	No	No	Medium
D9	9 AM	Sunny	Yes	No	Long
D10	10 AM	Cloudy	Yes	Yes	Long
D11	10 AM	Rainy	No	No	Short
D12	8 AM	Cloudy	Yes	No	Long
D13	9 AM	Sunny	No	No	Medium

Results for the Root Node

- Given our commute time sample set, we can calculate the entropy of each attribute at the root node.

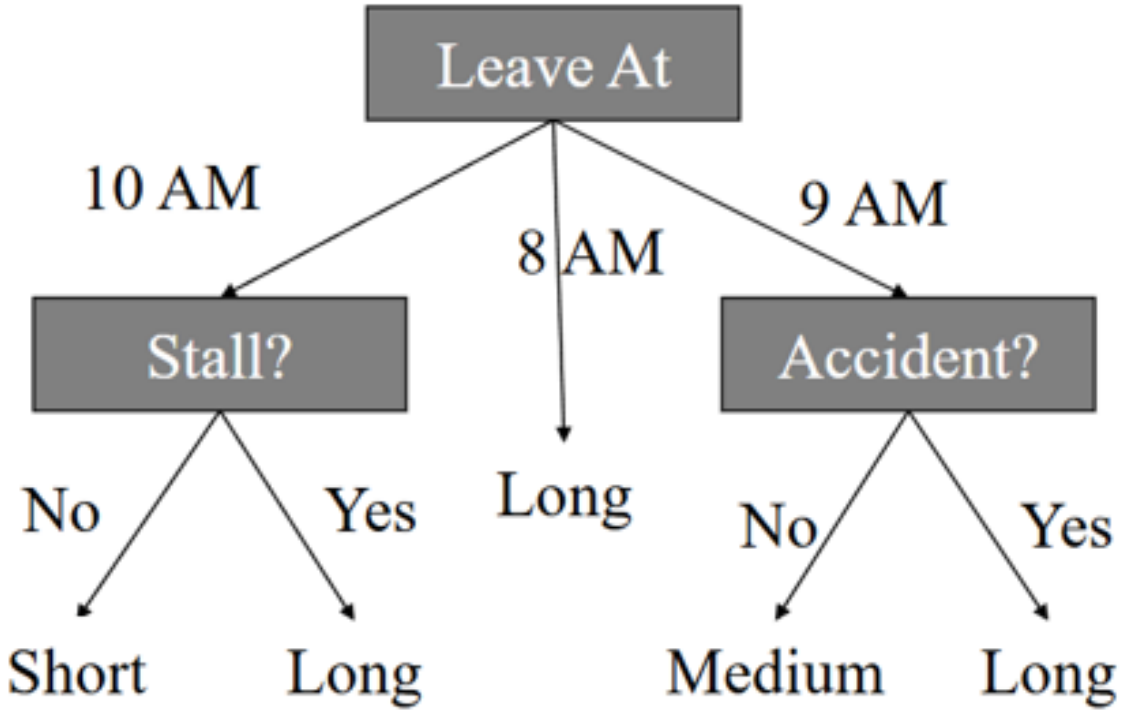
Attribute	Expected Entropy	Information Gain
Hour	0.6511	0.768449
Weather	1.28884	0.130719
Accident	0.92307	0.496479
Stall	1.17071	0.248842

- From this we conclude that the decision made at the root node is based on the Hour that we leave.



For the next level

- We consider each child of the root node one at a time recursively.
- Therefore, we consider Hour=8 first. In this case, all results have the same value – long commute. Since this is the case, we are done with that branch.
- Next, we consider Hour=9. In this case, we have to choose an attribute to split on.



For Hour=9

- Our choices are from the red part of the table

Example	Attributes				Target
	Hour	Weather	Accident	Stall	Commute
D4	9 AM	Rainy	Yes	No	Long
D5	9 AM	Sunny	Yes	Yes	Long
D8	9 AM	Rainy	No	No	Medium
D9	9 AM	Sunny	Yes	No	Long
D13	9 AM	Sunny	No	No	Medium

Entropy of S for Weather

$$H(S) = -(2/5)\log_2(2/5) - (3/5)\log_2(3/5) = 0.971$$

$$H(Weather = rain) = -(1/2)\log_2(1/2) - (1/2)\log_2(1/2) = 1$$

$$H(Weather = sunny) = -(2/3)\log_2(2/3) - (1/3)\log_2(1/3) = 0.918$$

Change in Entropy

$$IG = 0.971 - (2/5)(1) - (3/5)(0.918) = 0.0202$$

Calculating change in entropy

Entropy of S for Accident

$$H(S) = -(2/5)\log_2(2/5) - (3/5)\log_2(3/5) = 0.971$$

$$H(Accident = no) = 0$$

$$H(Accident = yes) = 0$$

Example	Attributes				Target
	Hour	Weather	Accident	Stall	Commute
D4	9 AM	Rainy	Yes	No	Long
D5	9 AM	Sunny	Yes	Yes	Long
D8	9 AM	Rainy	No	No	Medium
D9	9 AM	Sunny	Yes	No	Long
D13	9 AM	Sunny	No	No	Medium

Change in Entropy

$$IG = 0.971$$

Calculating change in entropy

Entropy of S for Stall

$$H(S) = -(2/5)\log_2(2/5) - (3/5)\log_2(3/5) = 0.971$$

$$H(Stall = no) = -(2/4)\log_2(2/4) - (2/4)\log_2(2/4) = 1$$

$$H(Stall = yes) = -(1/1)\log_2(1/1) = 0$$

Change in Entropy

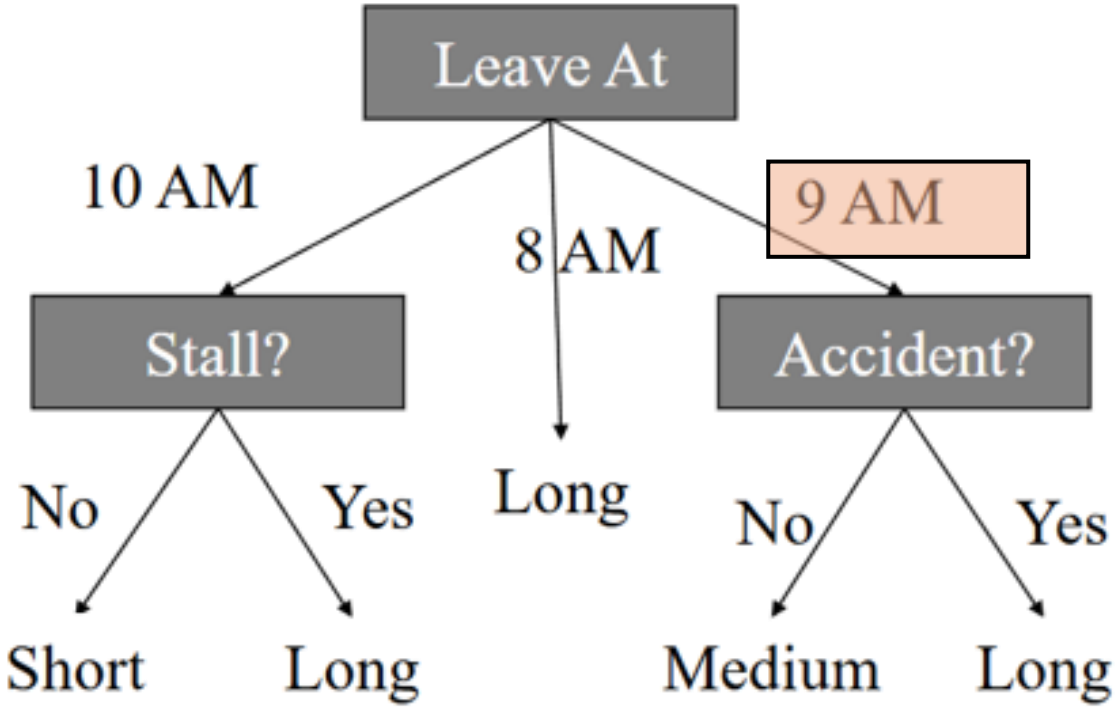
$$IG = 0.971 - (4/5)(1) = 0.170$$

Example	Attributes				Target
	Hour	Weather	Accident	Stall	Commute
D4	9 AM	Rainy	Yes	No	Long
D5	9 AM	Sunny	Yes	Yes	Long
D8	9 AM	Rainy	No	No	Medium
D9	9 AM	Sunny	Yes	No	Long
D13	9 AM	Sunny	No	No	Medium

Results for the node Hour=9

- From this we conclude that the decision made at Hour=9 node is that we should use the Accident attribute

Attribute	Expected Entropy	Information Gain
Weather	0.951	0.02
Accident	0	0.971
Stall	0.8	0.170



We must now consider
Accident = N. For this case all results are the same, so we are done with this branch.
Accident = Y. For this case all results are the same, so we are done with this branch

For Hour=10

- Our choices are from the red part of the table

Entropy of S for Weather

$$H(S) = -(4/5)\log_2(4/5) - (1/5)\log_2(1/5) = 0.729$$

$$H(Weather = rain) = -(1/1)\log_2(1/1) = 0$$

$$H(Weather = cloudy) = -(1/2)\log_2(1/2) - (1/2)\log_2(1/2) = 1$$

$$H(Weather = sunny) = -(2/2)\log_2(2/2) = 0$$

Change in Entropy

$$IG = 0.729 - (2/5)(1) = 0.729 - 0.4 = 0.329$$

Example	Attributes				Target
	Hour	Weather	Accident	Stall	Commute
D3	10 AM	Sunny	No	No	Short
D6	10 AM	Sunny	No	No	Short
D7	10 AM	Cloudy	No	No	Short
D10	10 AM	Cloudy	Yes	Yes	Long
D11	10 AM	Rainy	No	No	Short

Calculating change in entropy

Entropy of S for Stall

$$H(S) = -(4/5)\log_2(4/5) - (1/5)\log_2(1/5) = 0.729$$

$$H(Stall = no) = 0$$

$$H(Stall = yes) = 0$$

Change in Entropy

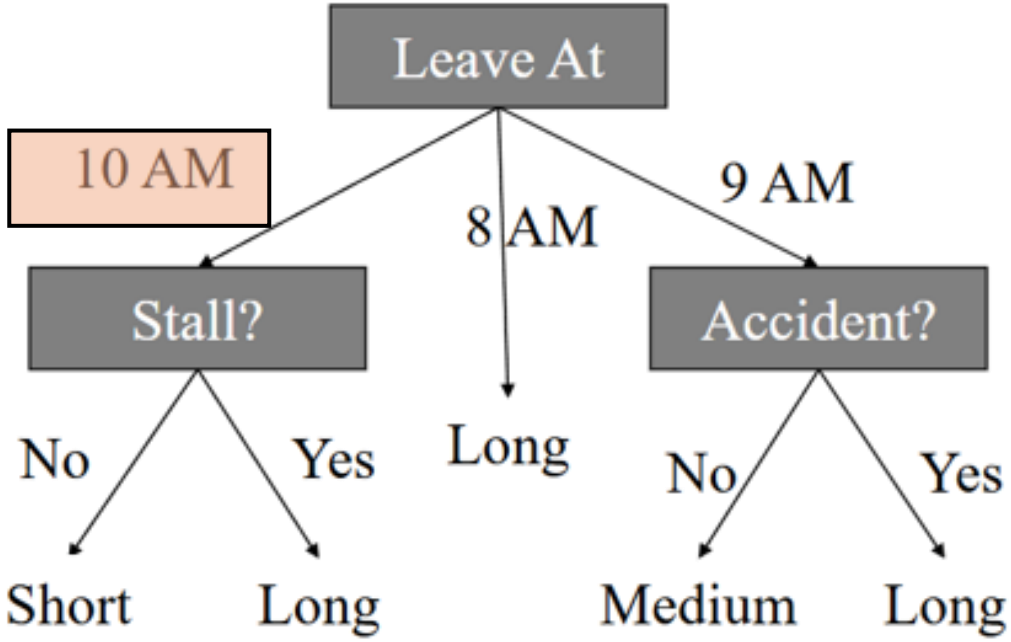
$$IG = 0.729$$

Example	Attributes				Target
	Hour	Weather	Accident	Stall	Commute
D3	10 AM	Sunny	No	No	Short
D6	10 AM	Sunny	No	No	Short
D7	10 AM	Cloudy	No	No	Short
D10	10 AM	Cloudy	Yes	Yes	Long
D11	10 AM	Rainy	No	No	Short

Results for the node Hour=10

- From this we conclude that the decision made at
- Hour=10 node is that we should use the Stall attribute

Attribute	Expected Entropy	Information Gain
Weather	0.4	0.329
Stall	0	0.729



We must now consider
Stall = N. For this case all results are the same, so we are done with this branch.
Stall = Y. For this case all results are the same, so we are done with this branch

Using ID3

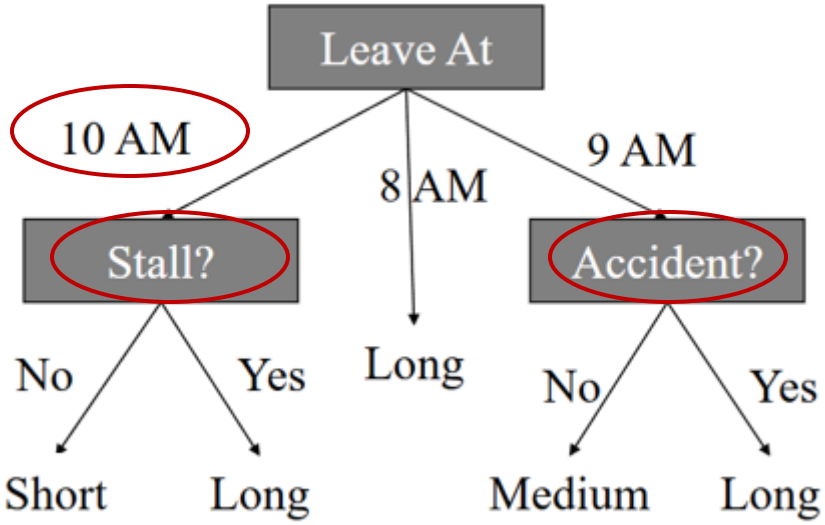
- Advantages
  - **Understandable prediction** rules are created from the training data.
  - Builds a **short tree**.
  - Only need to **test enough attributes** until all data is classified.
  - **Whole dataset** is searched to create tree.
- Disadvantages
  - Data may be **over-fitted** or **over-classified**, if a small sample is tested.
  - Only **one attribute** at a time is tested for making a decision.
  - **Classifying continuous** data may be **computationally expensive**, as many trees must be generated to see where to break the continuum.

Algorithms in Decision Trees

- ID3 (Iterative Dichotomiser 3)
  - classification tasks.
  - Splits nodes based on **Information Gain** (reduction in entropy).
  - Works best with categorical data.
- CART (Classification and Regression Trees)
  - classification and regression tasks.
  - Splits nodes using Gini Impurity for classification or Mean Squared Error (MSE) for regression.
- C4.5
  - classification tasks.
  - Splits nodes using **Gain Ratio**, which normalizes Information Gain to address its bias toward attributes with many values.
  - Handles **continuous attributes** by dynamically creating threshold splits (e.g., $X>5X > 5X>5$).

Parameters

- Parameters are the internal values or characteristics learned by the model during training.
- They are determined from the data and represent the structure of the trained decision tree.
- Key Point: Parameters are not set by the user; they are learned from the training data.

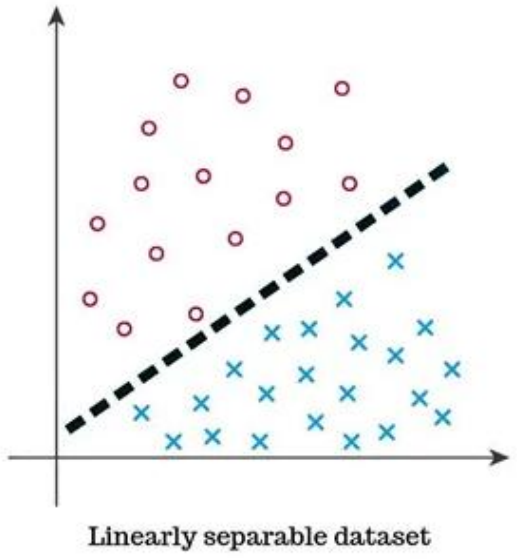


Hyperparameters

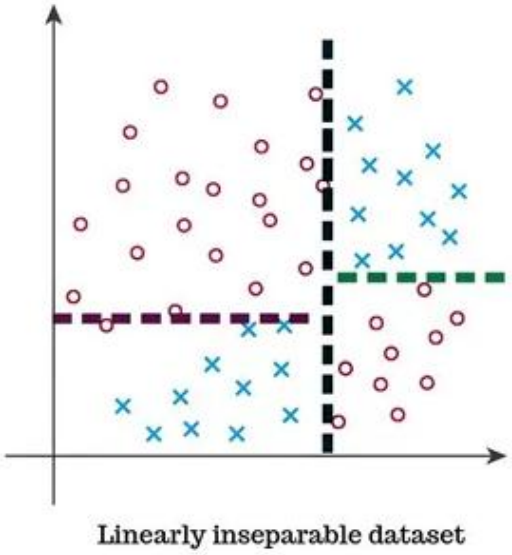
- Hyperparameters are the settings or configurations specified before training the model.
- They control how the decision tree is built and its complexity.
- Key Point: Hyperparameters are set by the user or via optimization techniques like grid search or random search.
- Some Examples:
  - Depth
  - Sample to form a leaf node
  - Criteria to split on (entropy/ gini/MSE)

Decision Boundary in Decision Trees

- In a decision tree, a "decision boundary" refers to the line or dividing point in the feature space that separates different classes.
- It defines the border between regions where the model predicts one class versus another, effectively creating distinct decision regions within the data space.



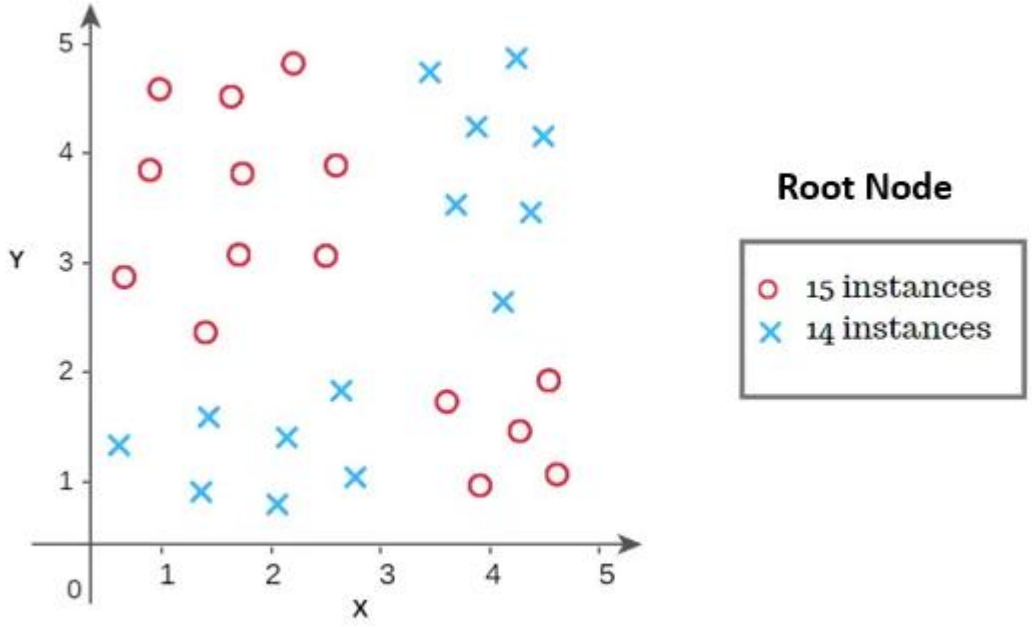
Linear Classifier



Decision tree Classifier

Decision Boundary in Decision Trees

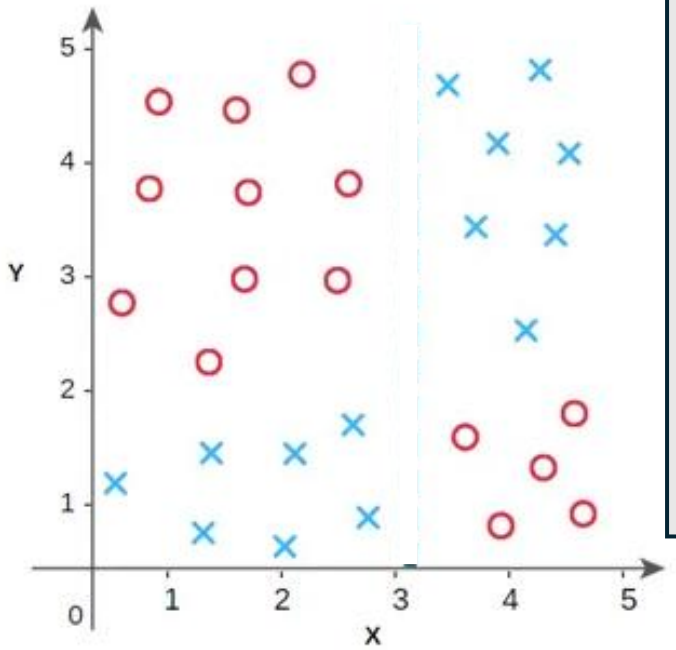
McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology



Root Node

15 instances
14 instances

Dataset to fit the decision tree on along with its corresponding tree structure

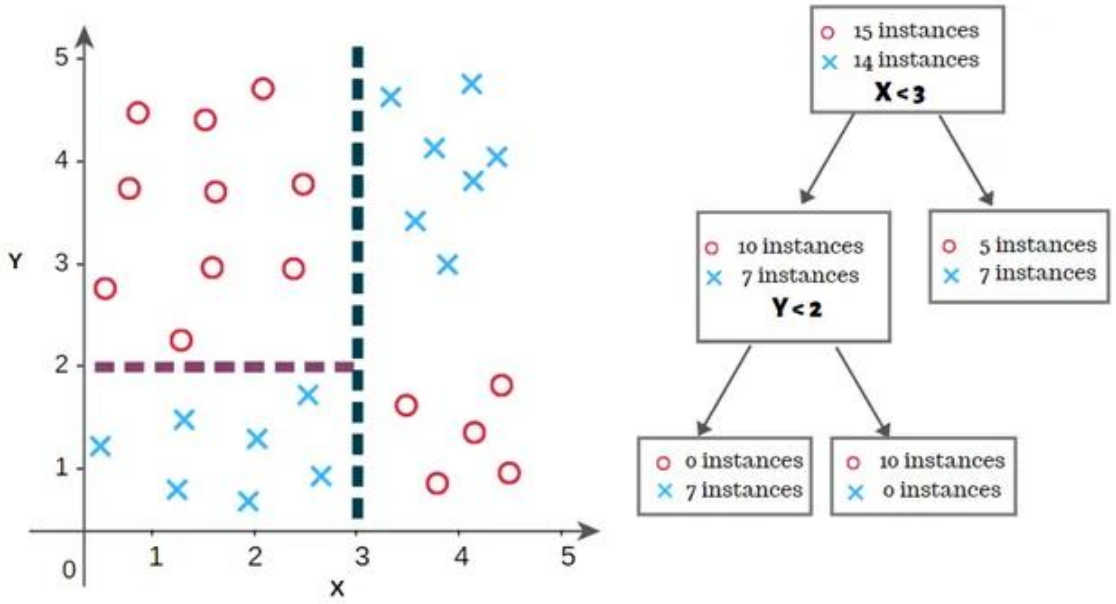


???

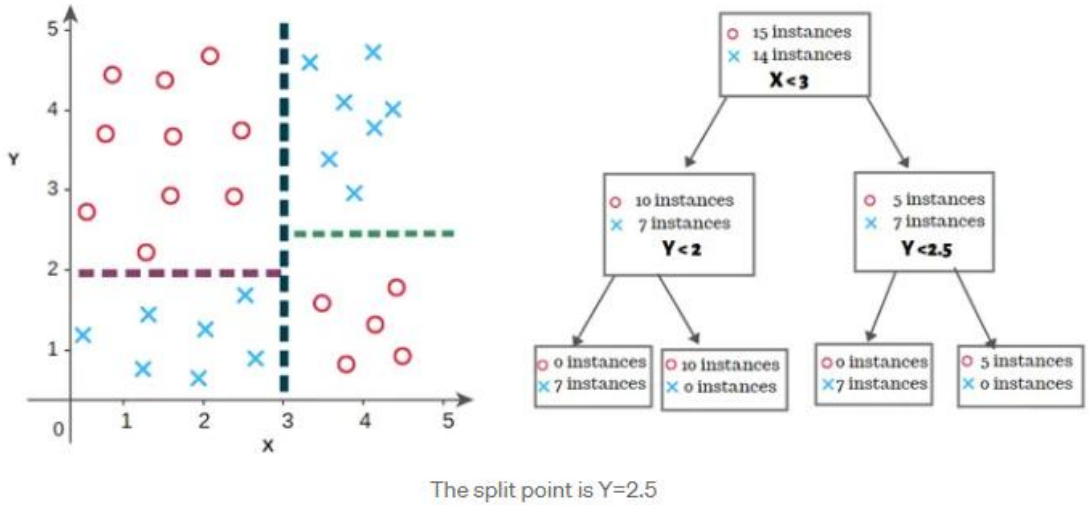
Split point is X=3

Decision Boundary in Decision Trees

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology



Split point is Y=2



The split point is Y=2.5

Classification Vs Regression

- Python Example

Contents

Generalization

- Fundamental Goal : To generalize beyond what we see in the training examples.
- New examples should be representative of the training data.
- They should have the same characteristics as the training data.

Training data

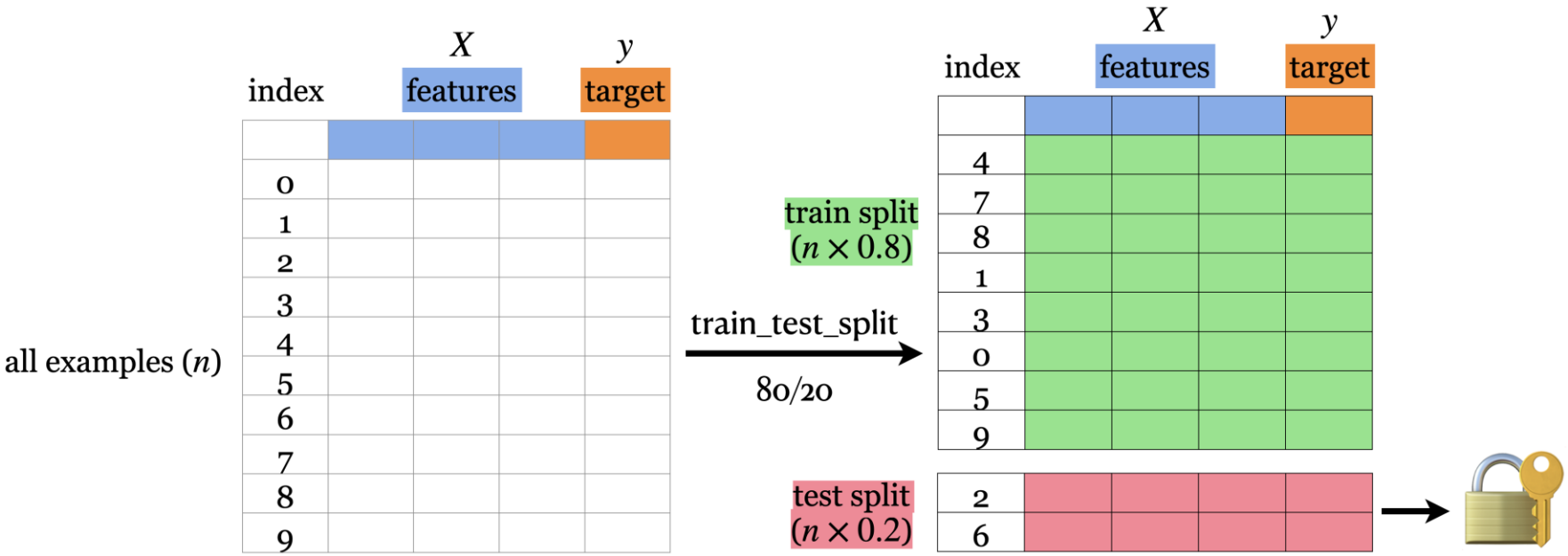
	CAT
	CAT
	DOG
	DOG

New examples for prediction

1		?
2		?
3		?
4		?

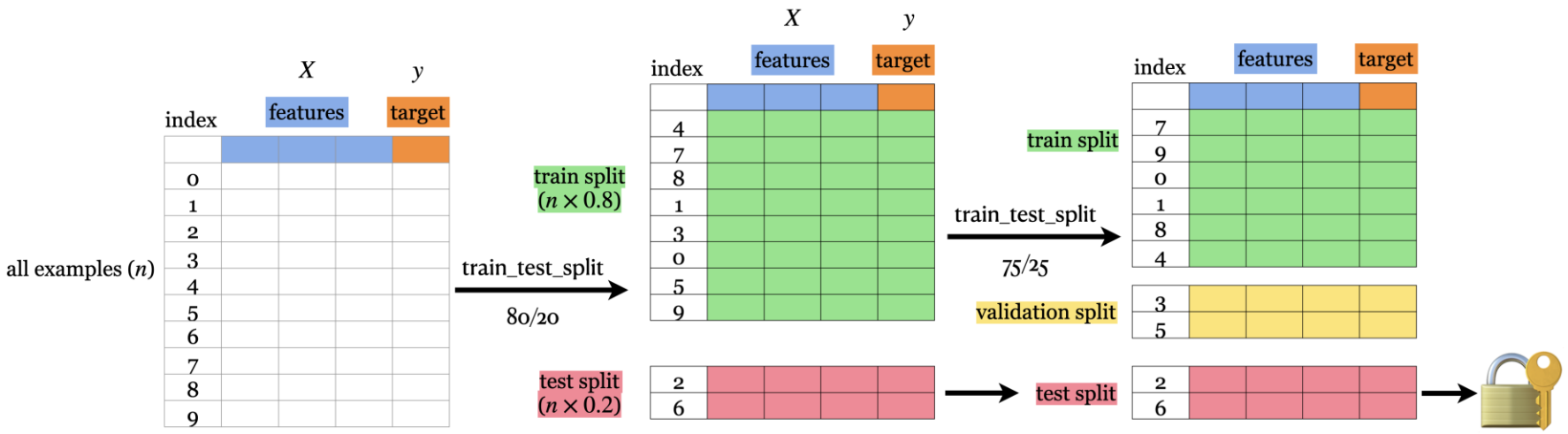
Data Splitting

- The picture shows an 80%-20% split of a toy dataset with 10 examples.
- The data is shuffled before splitting.
- Usually when we do machine learning we split the data before doing anything and put the test data in an imaginary chest lock.



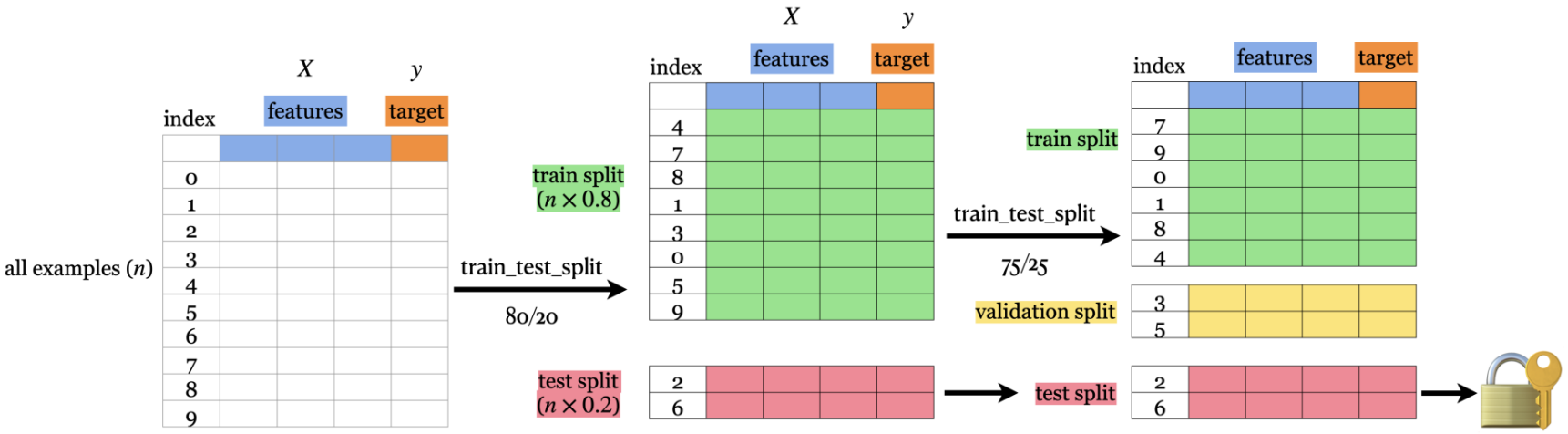
Train/validation/test split

- Training Dataset: The sample of data used to fit the model.
- Validation Dataset: The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyperparameters.
- Test Dataset: The sample of data used to provide an unbiased evaluation of a final model fit on the training dataset.



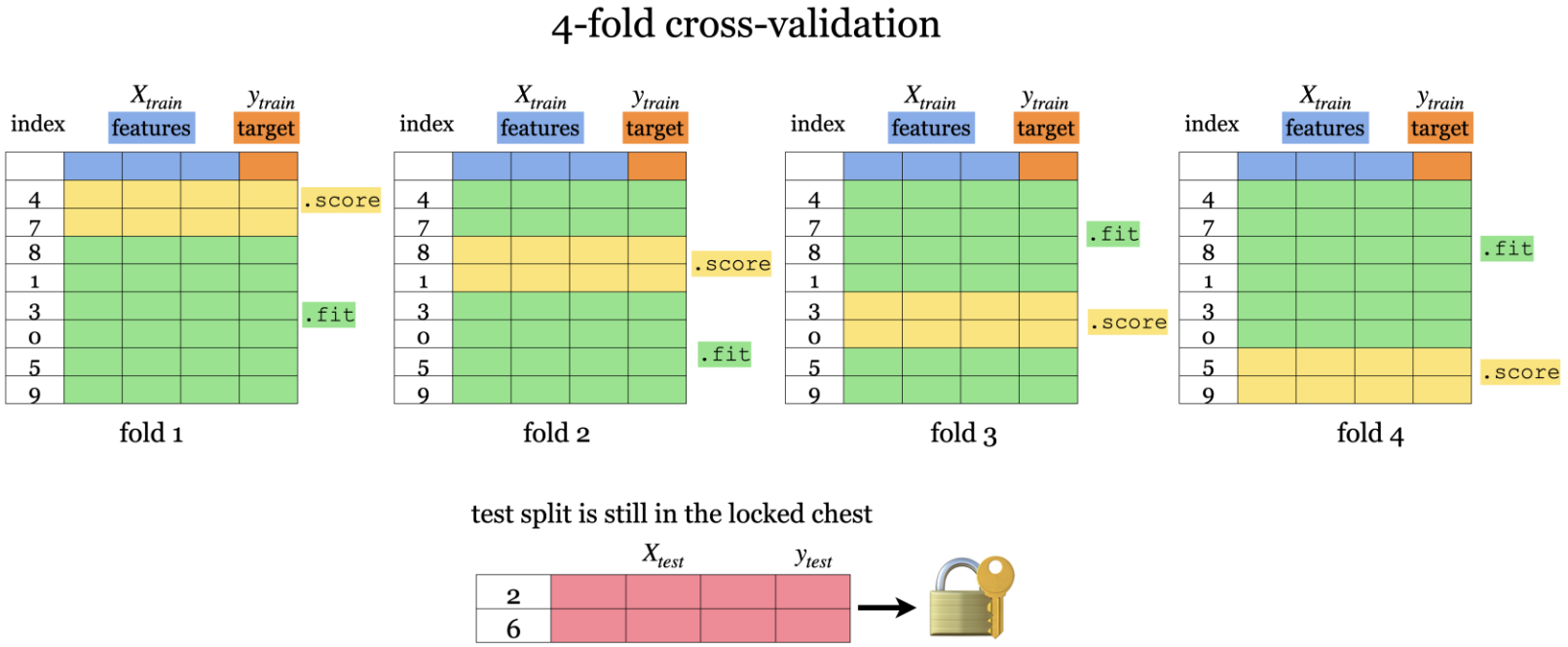
Problems with single train/validation split

- Only using a portion of your data for training and only a portion for validation.
- If your dataset is small you might end up with a tiny training and/or validation set.
- You might be unlucky with your splits such that they don't align well or don't well represent your test data.



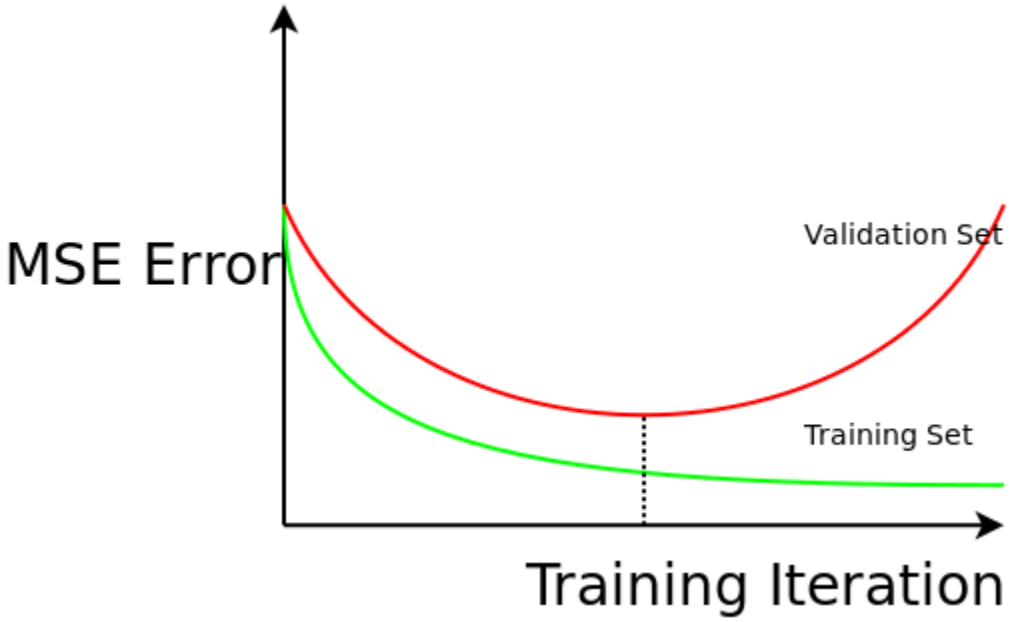
Cross Validation

- Cross-validation provides a solution to this problem.
- Split the data into k folds (k>2 often k=10).
- Each “fold” gets a turn at being the validation set.



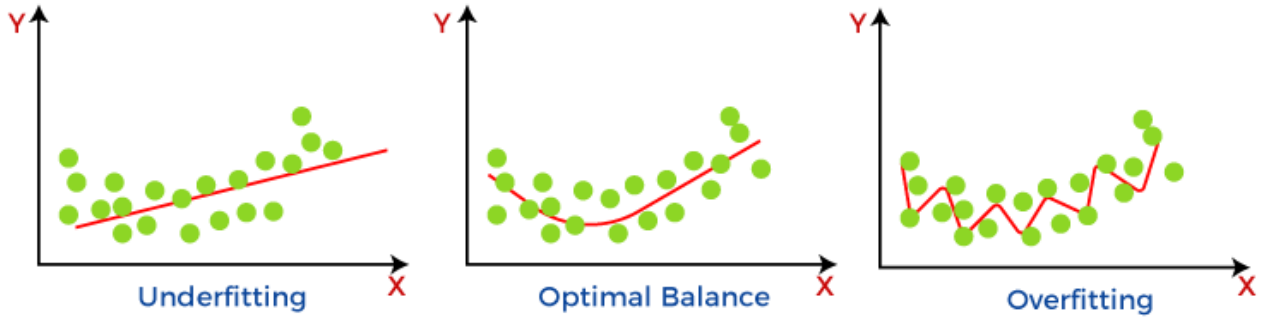
Types of errors

- E_{train} is your training error (or mean train error from cross-validation).
- E_{valid} is your validation error (or mean validation error from cross-validation).
- E_{test} is your test error.



Underfitting Vs Overfitting

- Overfitting: The model learns the training data too well, performing perfectly on it but poorly on new, unseen data. This is non-generalizable and not ideal.
- Underfitting: The model fails to learn the training data properly, performing poorly on both the training data and new inputs. This is also not desirable.
- Good Fit: The model learns the training data well and generalizes effectively to new inputs. This is the ideal scenario for a machine learning model.



Bias and Variance Tradeoff

- Bias: Errors due to bias occur when a model oversimplifies the problem, leading to poor predictions on both training and new data. High bias prevents generalization.
- Variance: Variance measures how much a model's predictions deviate from the expected value. High variance means the model is too complex and overfits the training data, failing to generalize to unseen data.
- Balancing Bias and Variance: For good generalization, aim for a model with low bias (understands the data well) and low variance (consistent predictions on new data). This balance ensures the model performs well on both training and unseen datasets.