HACKWAGON
• ACADEMY •

DATA SCIENCE 102: DECISION TREE

AGENDA

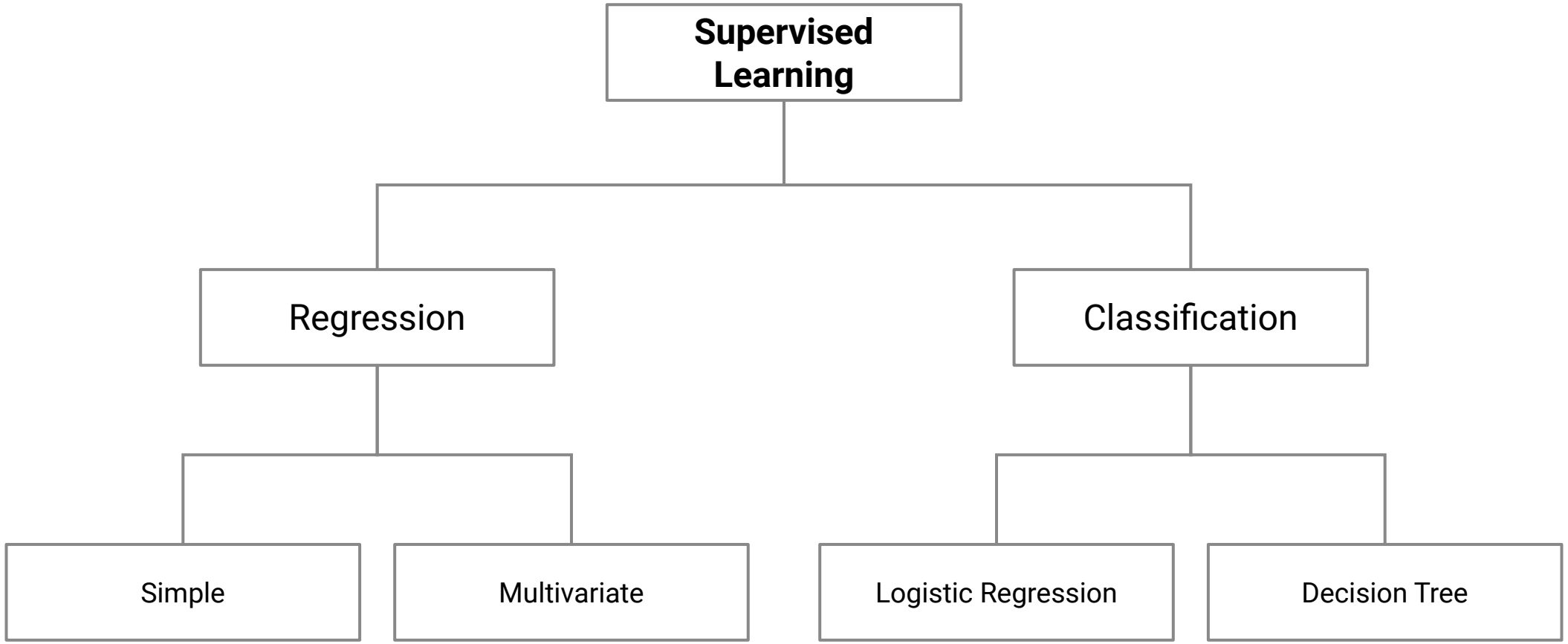
- Supervised Learning
- Decision Tree
  - Intuition
  - Interpretation
  - Improved Prediction Trees
  - Strengths and Weaknesses
- Model Evaluation
  - Confusion Matrix
  - Statistical Measures
  - Overfitting
  - Pruning

SUPERVISED LEARNING

SUPERVISED LEARNING

- Supervised Learning
  - Based on **labelled data,** makes **predictions** on a test set
  - There are two types of problems for supervised learning: **Regression** and **Classification**

SUPERVISED LEARNING



DECISION TREES

- Classification Trees
- Intuition
- Interpretation
- Improved Prediction Trees
- Strengths and Weaknesses

DECISION TREES

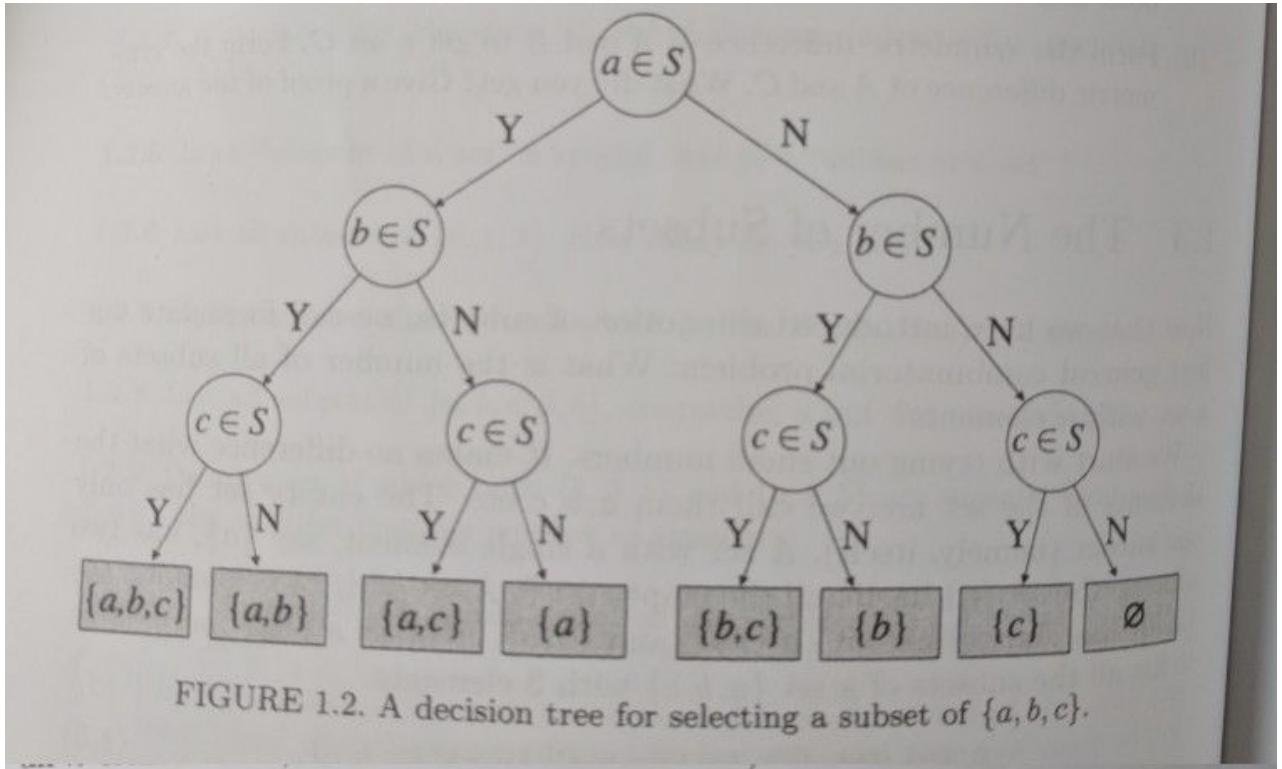


FIGURE 1.2. A decision tree for selecting a subset of $\{a, b, c\}$.

Remark. A picture like this is called a *tree*. (This is not a formal definition; that will follow later.) If you want to know why the tree is growing upside down, ask the computer scientists who introduced this convention. (The conventional wisdom is that they never went out of the room, and so they never saw a real tree.)

DECISION TREES

The Akinator is a good use case of Decision Trees:
https://en.akinator.com

DECISION TREES

- Decision Trees are flexible and easy to interpret
- Based on separating records into **subgroups** by creating **splits** on predictors
- These **splits** create logical rules that are transparent and easily understandable
- The resulting **subgroups** should be more homogeneous in terms of the outcome variable, thereby creating useful prediction or classification rules

INTUITION

- A decision tree is like an if-else statement that helps to regress or classify any object based on **predetermined questions**
- Difference between Decision Tree and If-Else statement
  - A decision tree is constructed **automatically** from your dataset
  - This why we call it *machine learning* because it makes decisions based on the data set
- When constructing a decision tree, it chooses the question that has the **lowest impurity**
  - What questions to ask (what to split on)
  - In what order to ask these questions (what order to split on)

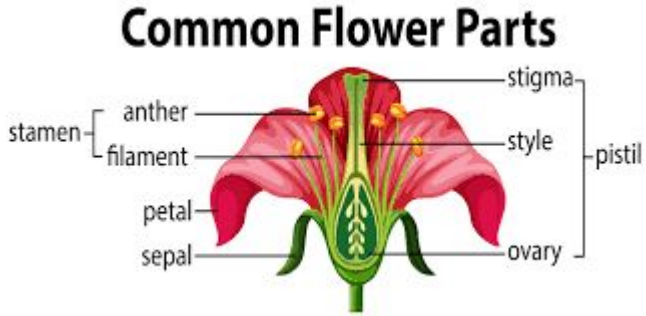
INTUITION - MEASURES OF IMPURITY

- How does the decision tree *split*?
- The decision tree splits by two different popular measures:
  - **Gini Impurity** - Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labelled if it was randomly labelled according to the distribution of labels in the subset
  - **Entropy** - Information Gain; higher the information gain, the better the feature is at homogenous data after the split
- For further explanation, refer to the additional readings slides deck on measuring impurity

INTUITION

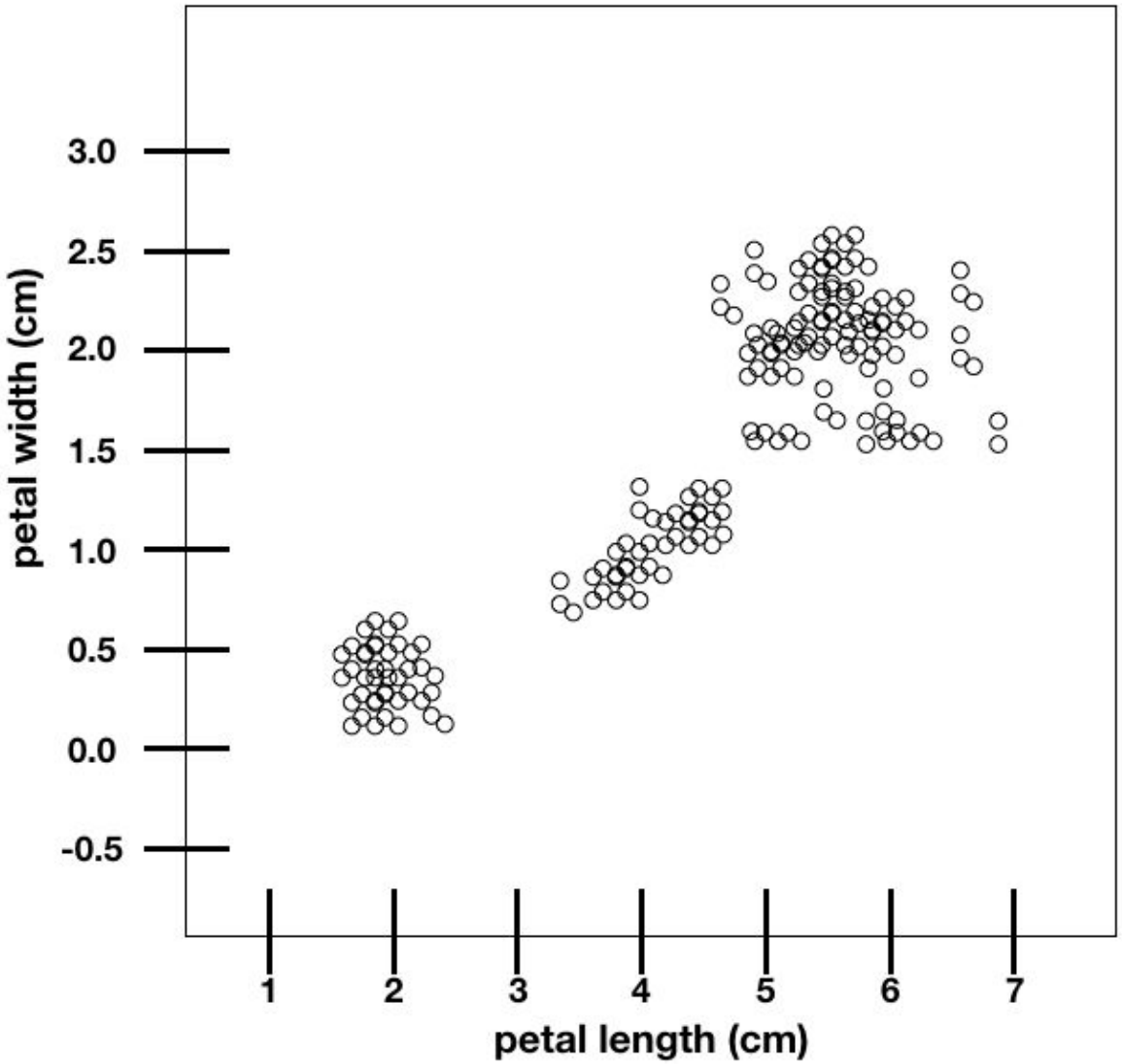
- Given the following dataset, construct a decision tree using just:
  - petal length (cm)
  - petal width (cm)

sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target	target name
5.1	3.5	1.4	0.2	0	setosa
5.9	3.2	4.8	1.8	1	versicolor
4.6	3.6	1.0	0.2	0	setosa
5.5	3.5	1.3	0.2	0	setosa
6.3	2.5	4.9	1.5	1	versicolor

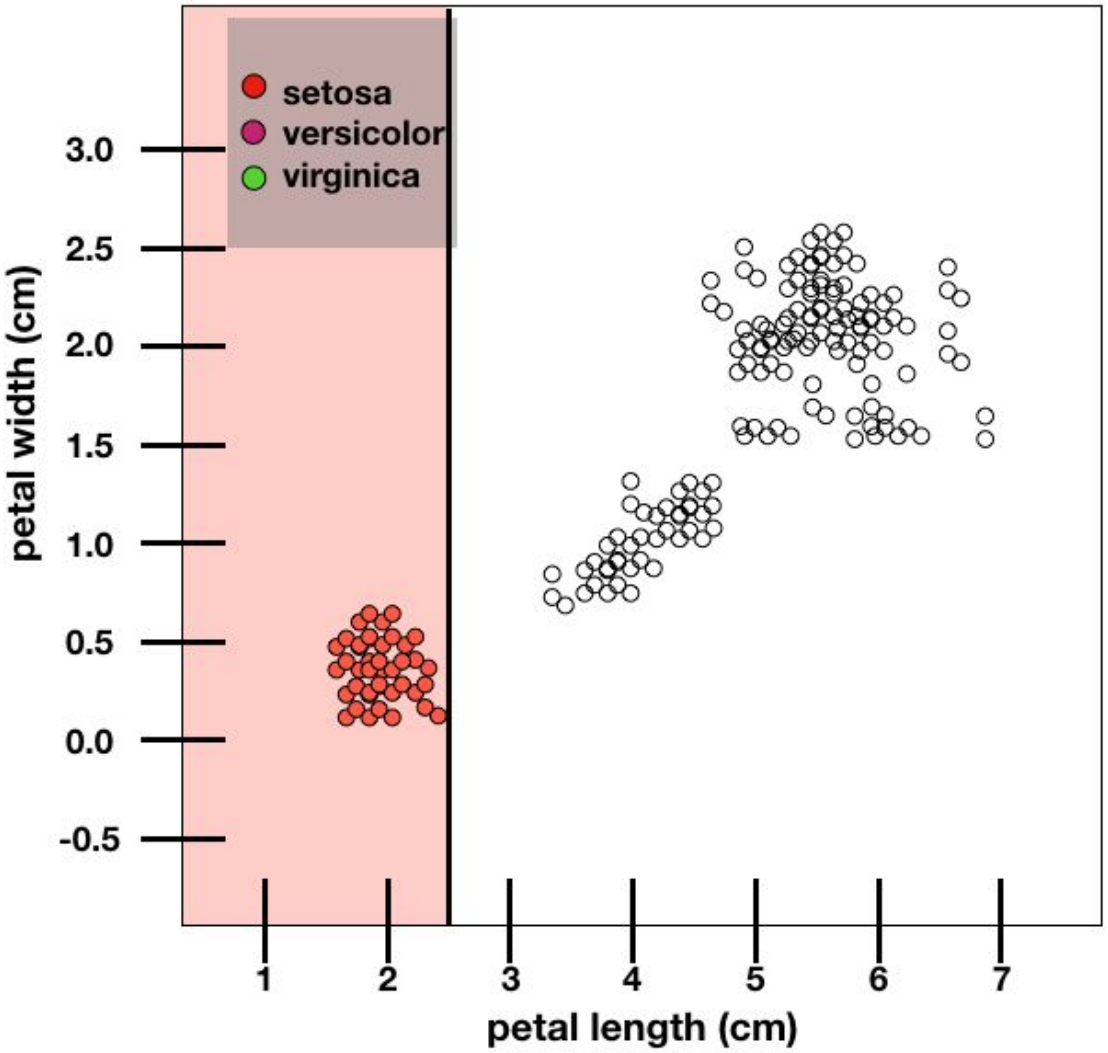


There are over 260 to 300 species of Iris

INTUITION

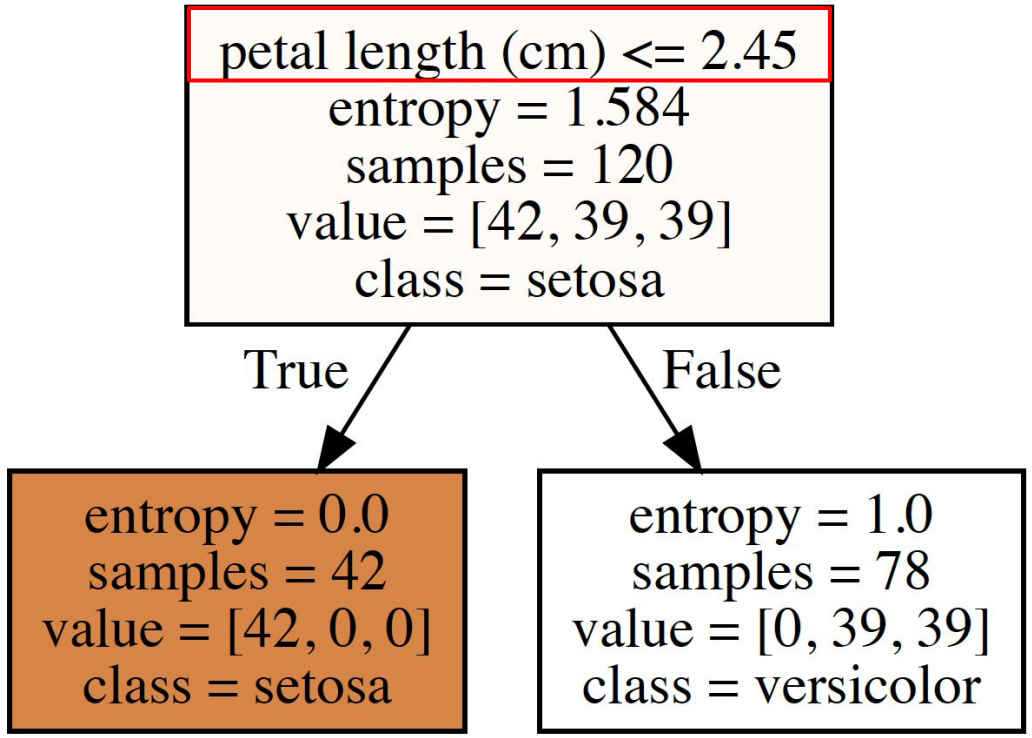


INTUITION

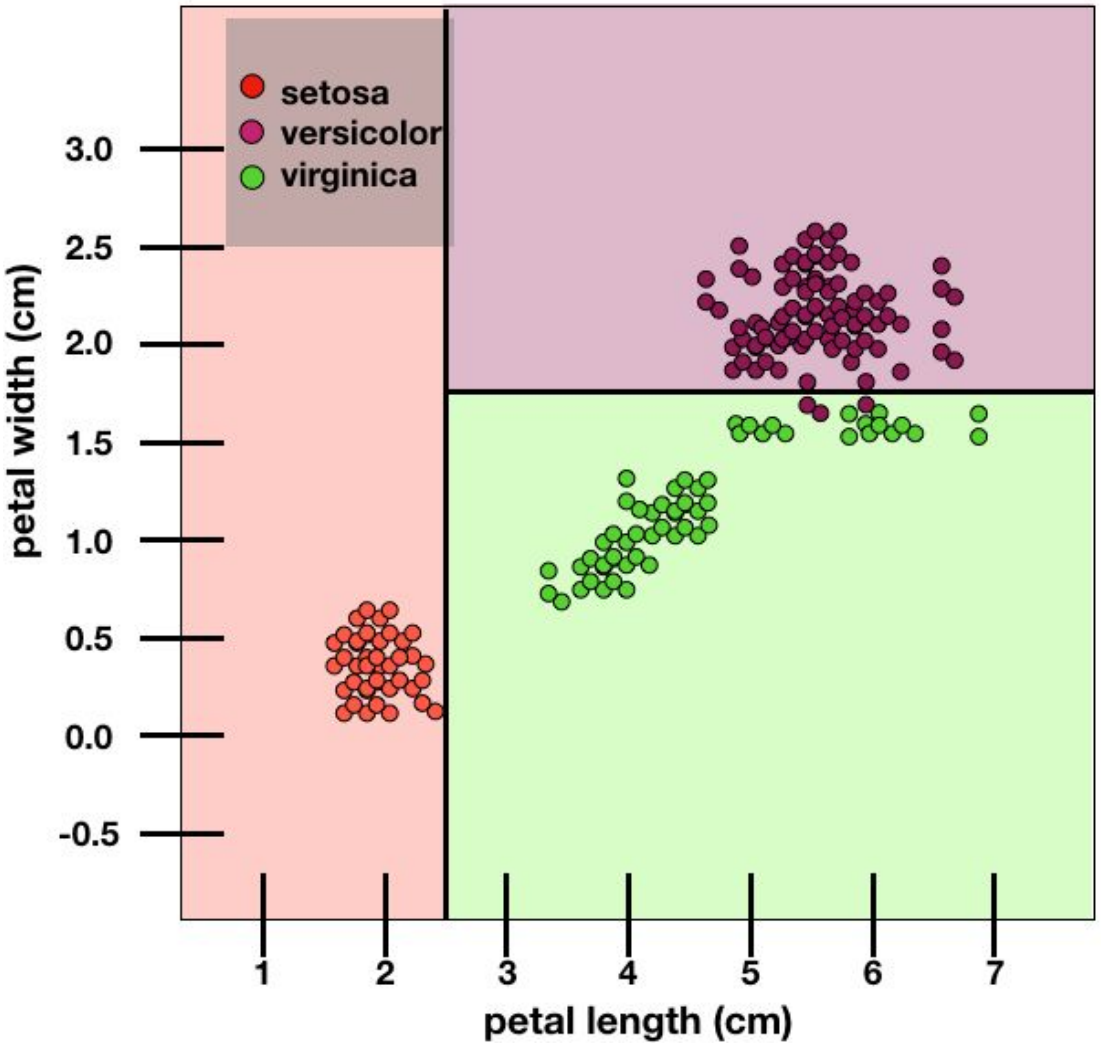


TREE DEPTH = 1

SPLIT 2 : Petal Length <= 2.45

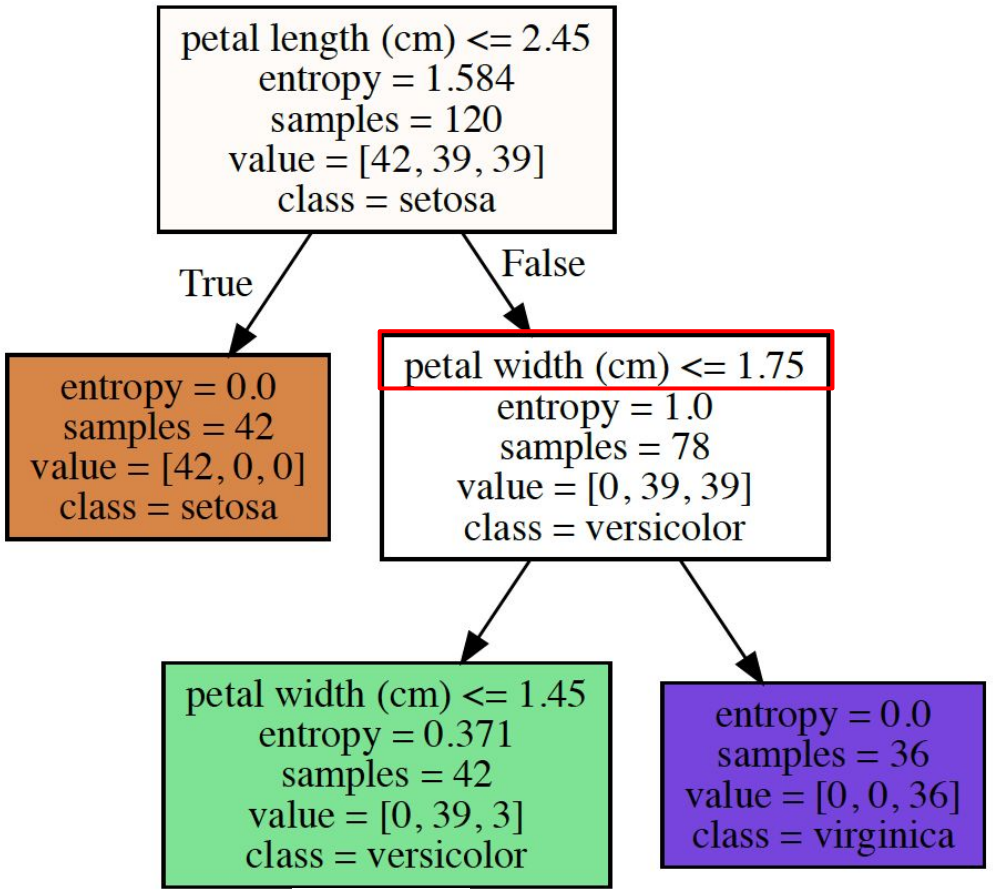


INTUITION

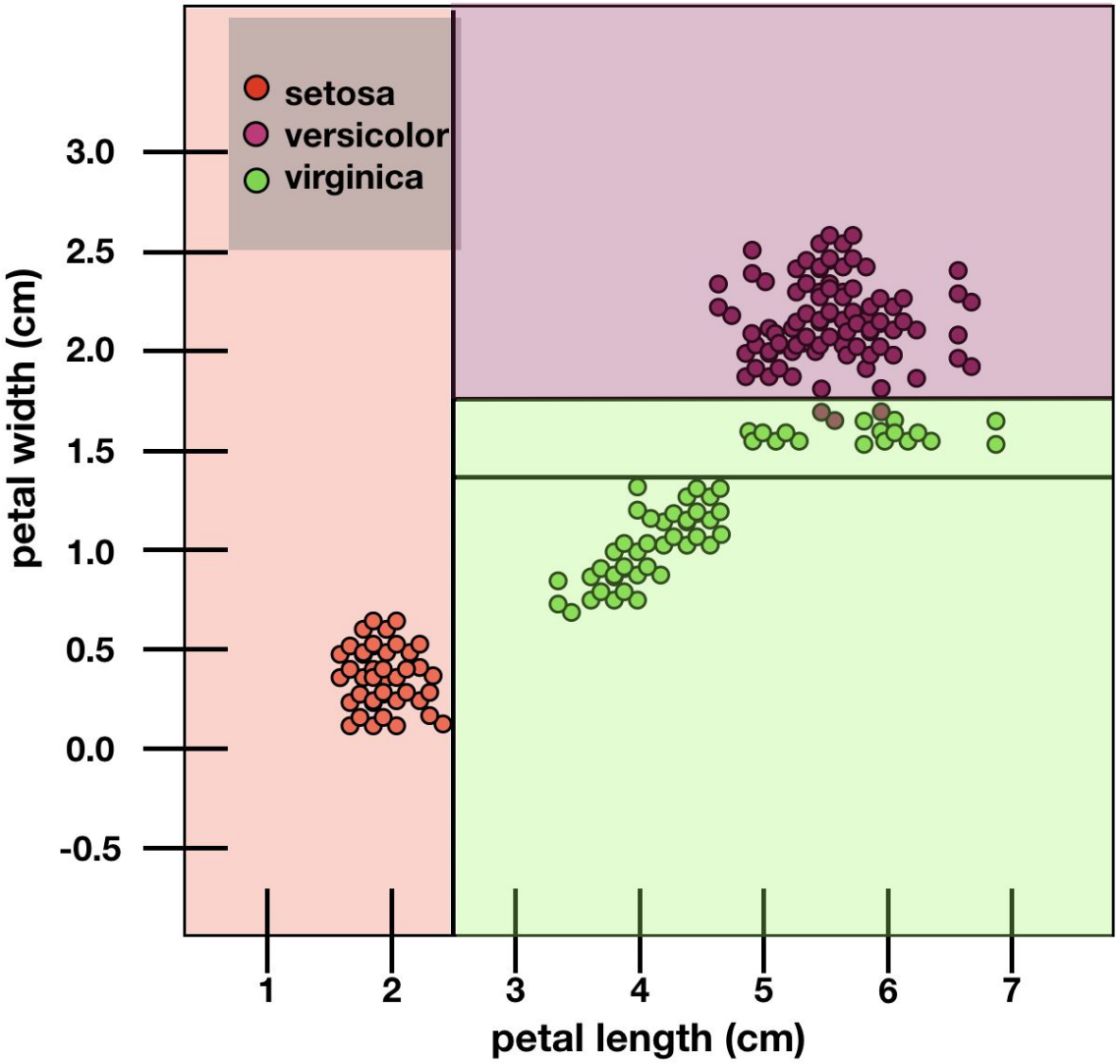


TREE DEPTH = 2

SPLIT 2 : Petal Width <= 1.75

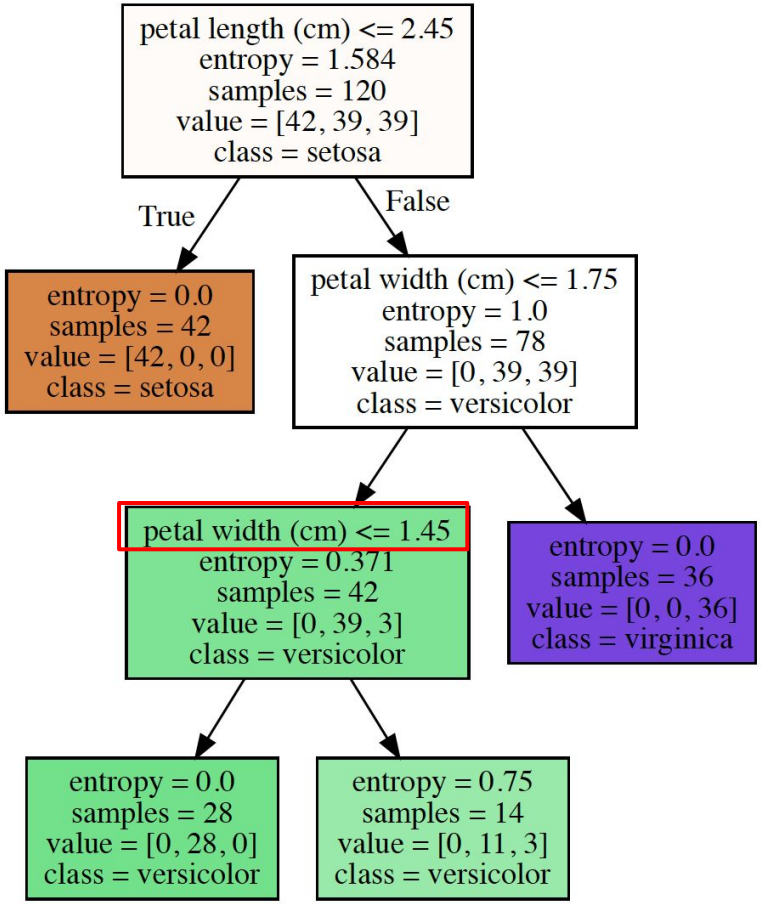


INTUITION

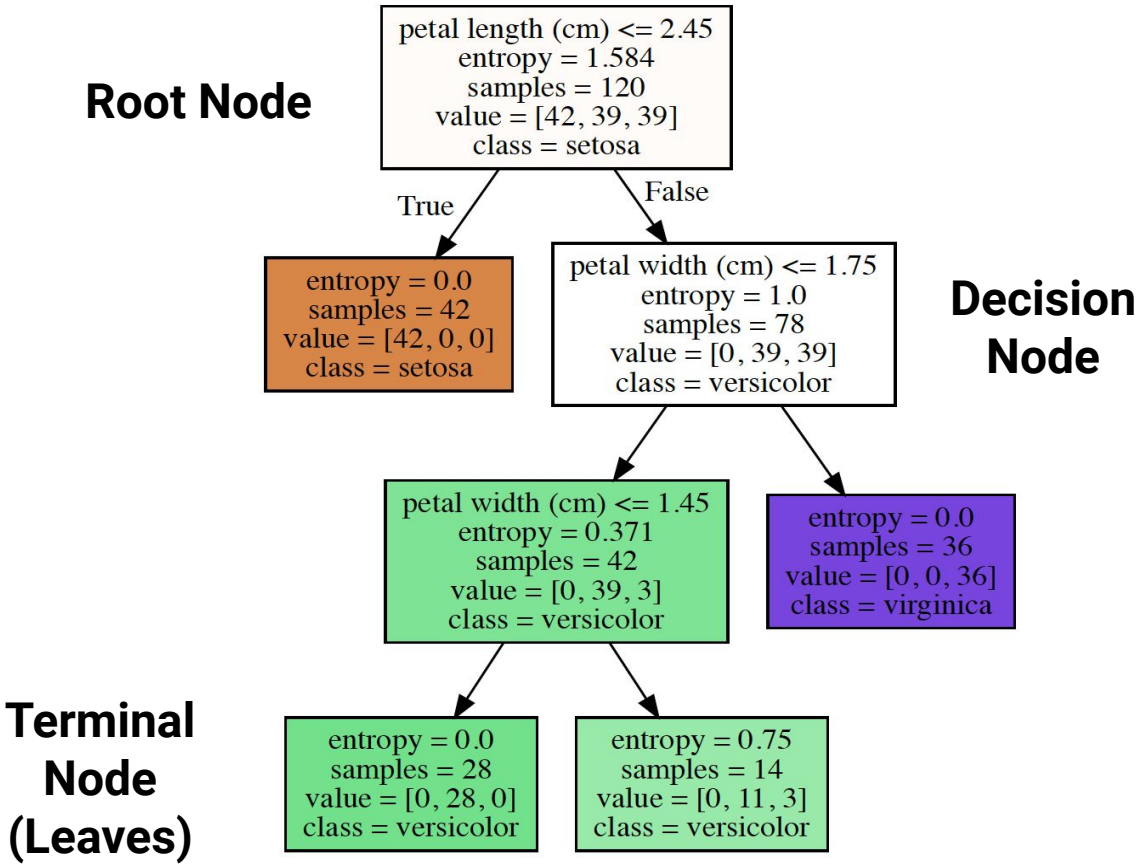


TREE DEPTH = 3

SPLIT 3 : Petal Width <= 1.45



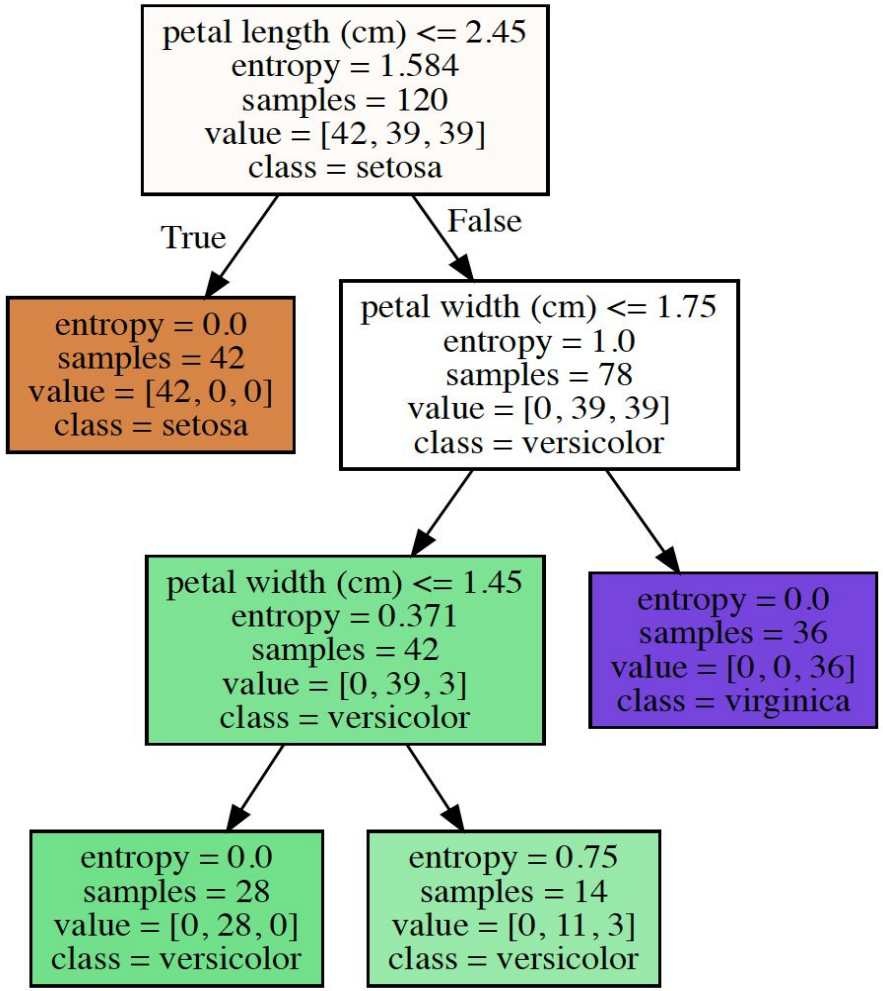
INTERPRETATION - STRUCTURE



- There are two types of nodes in a tree: decision nodes and terminal nodes (leaves)
- The first decision node of the tree is known as the **root node**

INTERPRETATION - RULES

- With a decision tree, we can have transparent and understandable rules
- Rules derived from this decision tree such as:
  - A Setosa Iris has a petal length below 2.45 cm
  - A Versicolor Iris has a petal length greater than 2.45 cm and petal width below 1.45 cm



INTERPRETATION - FEATURE IMPORTANCE & PREDICT

- By using `.feature_importances_`, you can identify the features which are important for classifying the tree
- The feature importances is calculated by the impurity and importance value at each node*
- By using `.predict()`, you can identify the class of the new data predicted
- This is done by “dropping” the new record down the tree. When it reaches a terminal node, that new data is classified / assigned its class

*For more information on how feature importance is calculated here:
https://medium.com/@srnghn/the-mathematics-of-decision-trees-random-forest-and-feature-importance-in-scikit-learn-and-spark-f2861df67e3

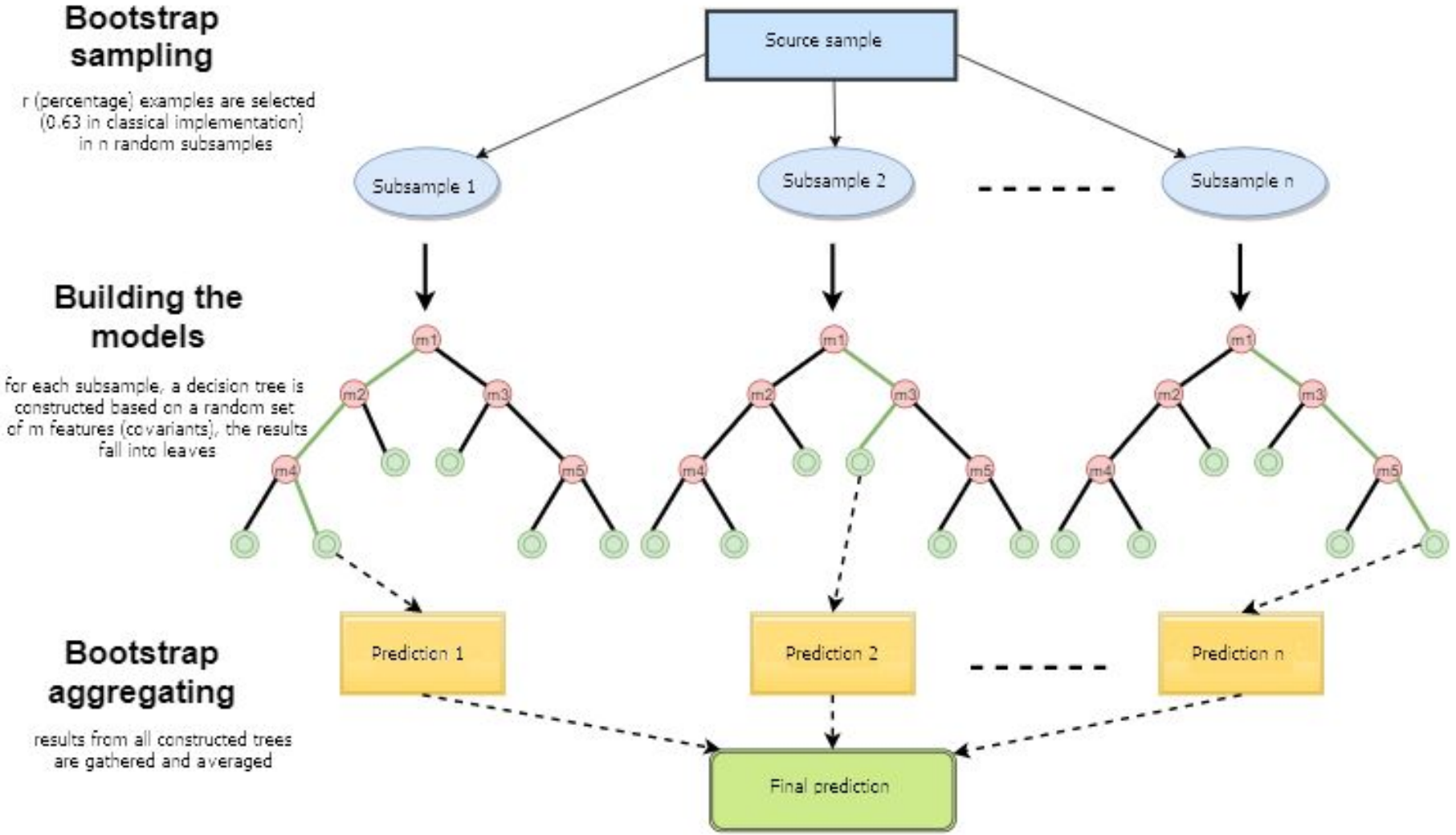
IRIS DATASET - IN-CLASS EXAMPLE*

- Walk through the code behind decision tree on the Iris dataset

IMPROVED PREDICTION TREES - RANDOM FOREST

- Introduced by Breiman and Cutler, *RandomForest* is an **ensemble** (many different trees) approach, using **bootstrap aggregating** to improve predictive power by combining multiple classifiers
- The basic idea in *RandomForest* is to:
  - Draw multiple random sample, with replacement, from the data (this method of sampling is called bootstrap)
  - Using a random subset of predictors at each stage, fit a classification (or regression) tree to each sample
  - Combine the predictions / classifications from the individual trees to obtain improved predictions, then use voting for classification and averaging the prediction

IMPROVED PREDICTION TREES - RANDOM FOREST



STRENGTHS & WEAKNESSES

Strengths	Weaknesses
Good off-the-shelf classifiers and predictors	Sensitive to changes in the data, where a slight change can cause very different splits
Useful for variable selection, identifying the most important variables which are usually at the top of the tree	Miss relationships between predictor variables, like those in linear or logistic regression
Requires little effort from users	

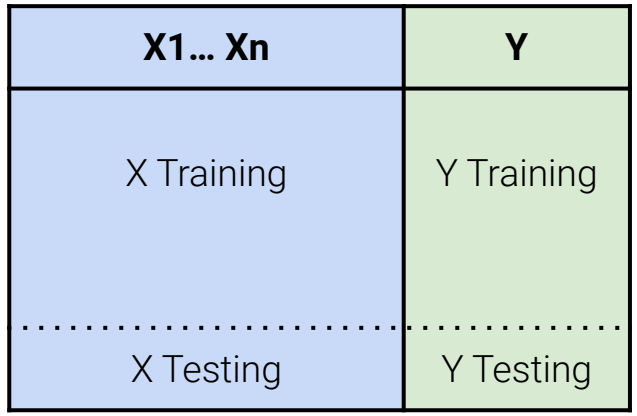
MODEL EVALUATION

- Confusion Matrix
- Statistical Measures
- Overfitting
- Pruning

MODEL EVALUATION

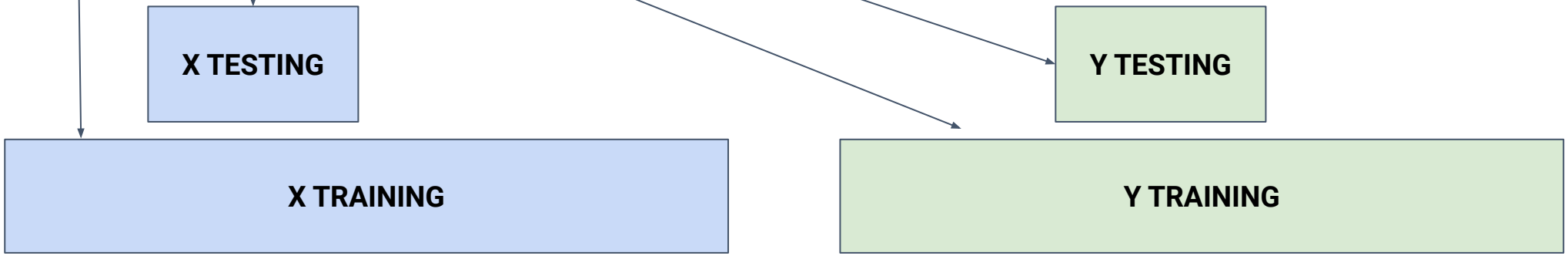
- Fitting a model to training is only the beginning of the Machine Learning process
- The next step is to use out-of-sample data **(using Train Test Split)** to assess and tune the model
  - Think of this like memorising for an exam answer by answer but failing when a completely new question set is used
- This is particularly important because:
  - Decision trees can be quite unstable depending on the sample chosen
  - A fully-fitted tree will lead to **overfitting**
- We can use a **Confusion Matrix** to assess the performance of the model by comparing the predicted and actual results

TRAIN TEST SPLIT - RECAP




```
housing_x = x_all[['sqft_living', 'floors']]

X_train, X_test, y_train, y_test = train_test_split(housing_x, y_all, random_state=42)
```



CONFUSION MATRIX

		ACTUAL		
		0	1	
PREDICTED	0	True Negative (TN)	False Negative (FN)	**Total Negatives (N)**
	1	False Positive (FP)	True Positive (TP)	**Total Positives (P)**
		Actual Negatives	**Actual Positives**	

- True Positive (TP) : **Actual** result of 1 is the same as **Predicted** result of 1
- False Positive (FP) : **Actual** result of 0 was **Predicted** as 1
- False Negative (FN) : **Actual** result of 1 was **Predicted** as 0
- True Negative (TN) : **Actual** result of 0 is the same as **Predicted** result of 0

Confusion Matrix can be found in `sklearn.metrics.confusion_matrix`

STATISTICAL MEASURE OF PERFORMANCE

- Accuracy = $\frac{TP + TN}{P + N}$
- Precision = $\frac{TP}{TP + FP}$ (TP + FP = Predicted Positives)
- Recall / Sensitivity (True Positive Rate) = $\frac{TP}{\text{Actual P}}$
- Specificity (True Negative Rate or 1 - False Positive Rate) = $\frac{TN}{\text{Actual N}}$ or 1 - $\frac{FP}{\text{Actual N}}$
- These measures can be found on `sklearn.metrics`

ACCURACY PARADOX

- Accuracy is usually the first measure to determine how good a predictive model is
- However, a predictive model with high accuracy may actually be of no help in prescriptive analytics because of it's **generalisability** is of no use to the problem
- Certain cases, **precision** and **recall** is sometimes favoured over accuracy as a "predictive power measure"

OVERFITTING

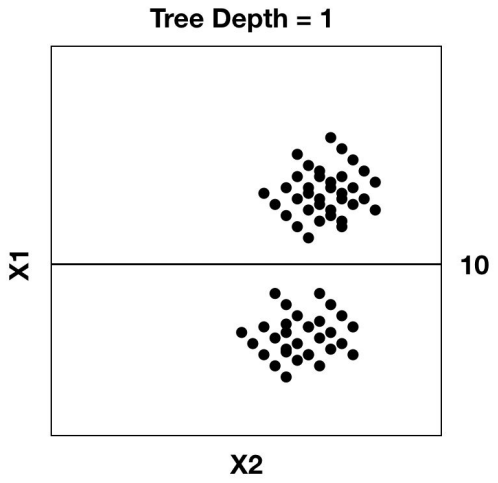
- Overfitting is one of the biggest problems when predictive models are concerned
- It happens when analysis from the predictive model “fits” too closely to a dataset and hence fails to make future predictions reliably
- Overfitting leads to poor performance on new data
- Particularly for Decision Trees, overfitting happens because the trees’ final splits are often based on very small number of records

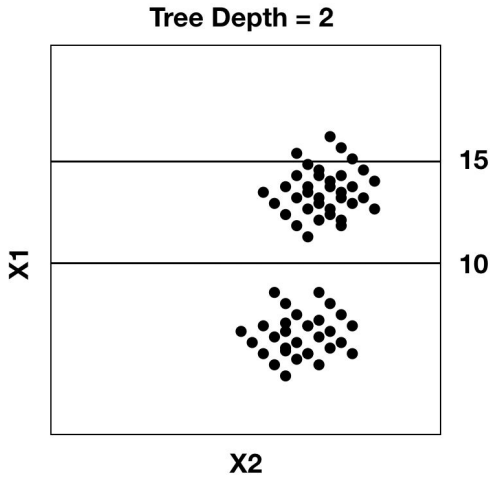
PRUNING

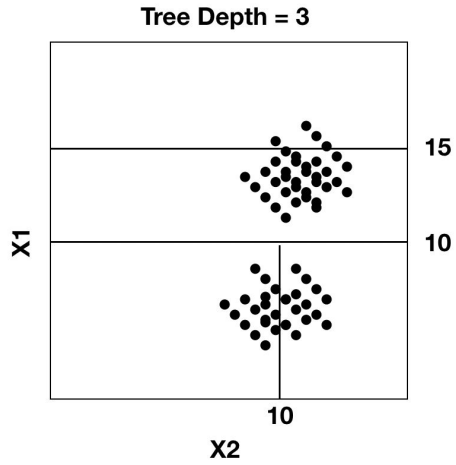
- Pruning is a popular method for stopping a growth of a tree
- A very large tree is likely to overfit the training data, and that the weakest branch should be removed
- Pruning consists of successively selecting a decision node and redesignating it as a terminal node
- Within the `DecisionTreeClassifier`, there is a parameter called `max_depth` which allows you to specify the depth of the tree

PRUNING - VISUALIZED

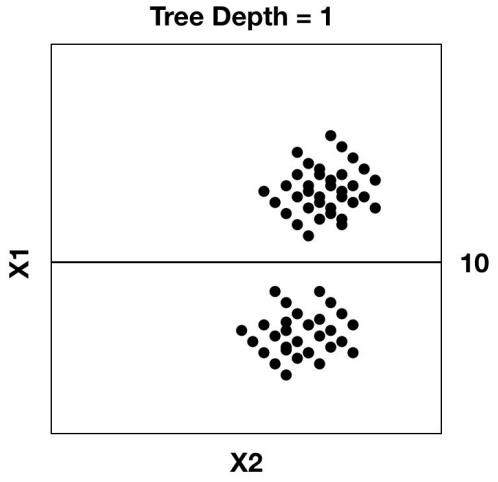
Full Tree







Pruned Tree



DIABETES DATASET - IN CLASS EXAMPLE*

- Follow the in-class example on the Diabetes dataset to assess the performance of the decision tree