

# Exploiting SAR to Monitor Agriculture

Heather McNairn, Xianfeng Jiao, Sarah Banks and Amir Behnamian

4 September 2019

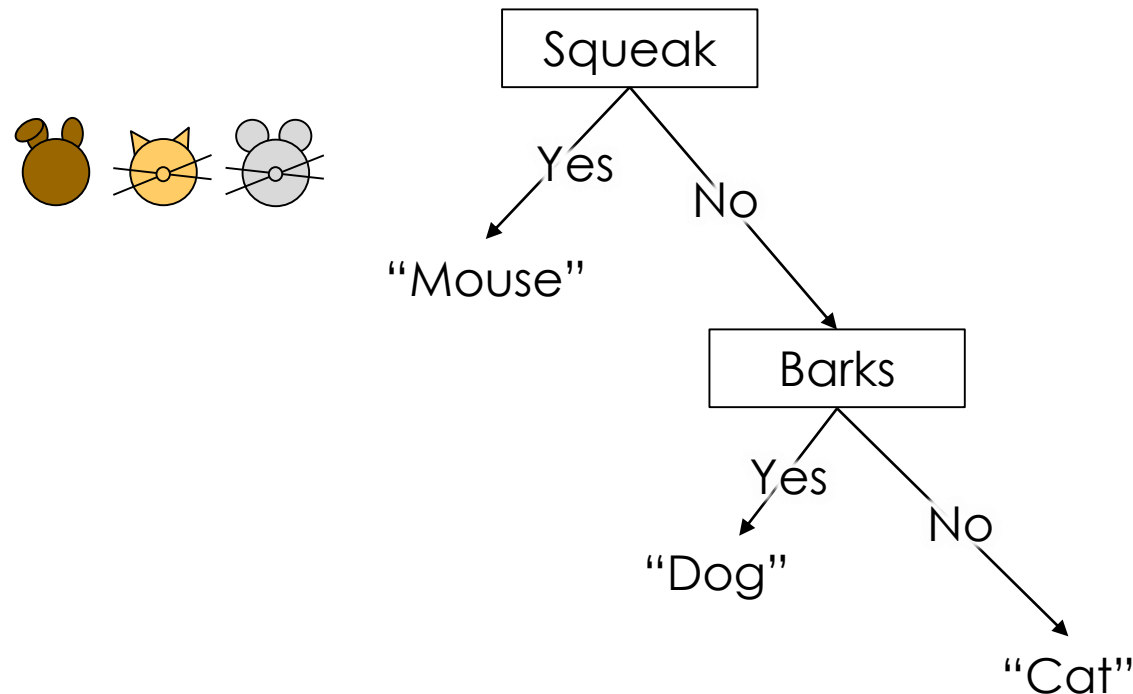
# Learning Objectives

By the end of this presentation, you will be able to understand:

- Classification and Regression Trees (CART)
- Random Forests Advantages
- Random Forests:
  - Basics
  - Inputs and parameters
- How to implement Random Forests in R

# Classification and Regression Trees (CART): Basics

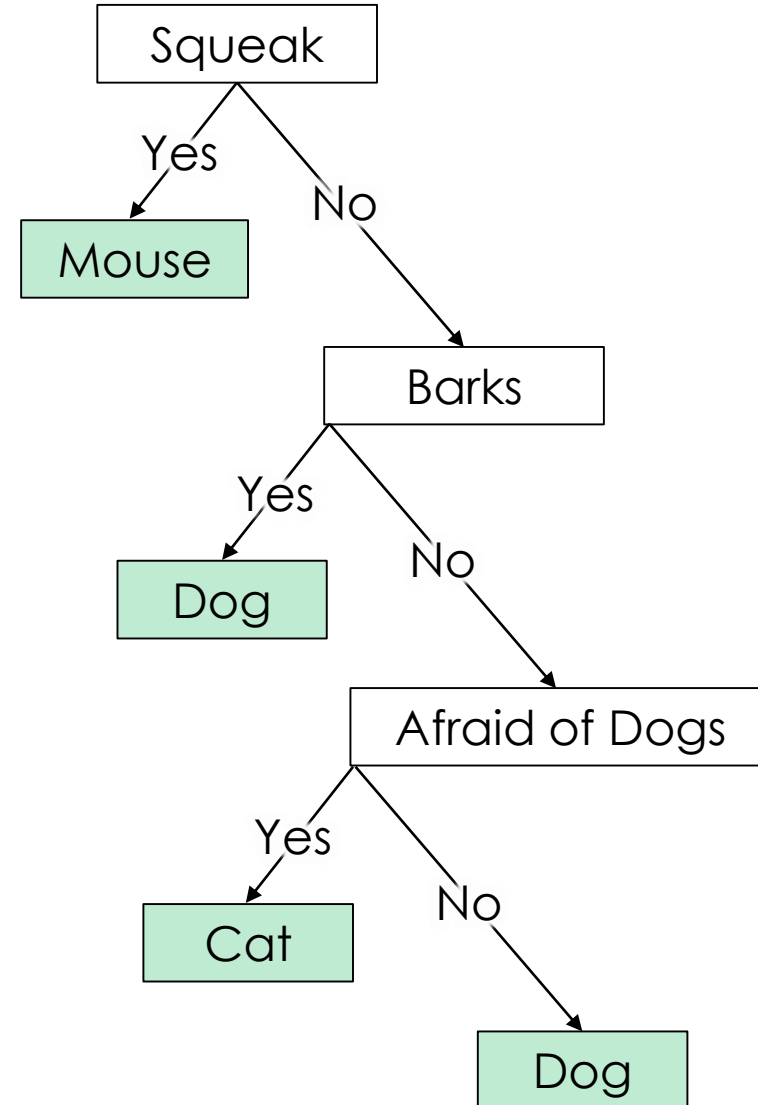
- Partitions or “splits” the data based on a set of binary tests
- Produces an easily-interpretable “tree” of tests and outcomes
- Tries to make groups that are as homogeneous as possible



# Classification and Regression Trees (CART): Basics

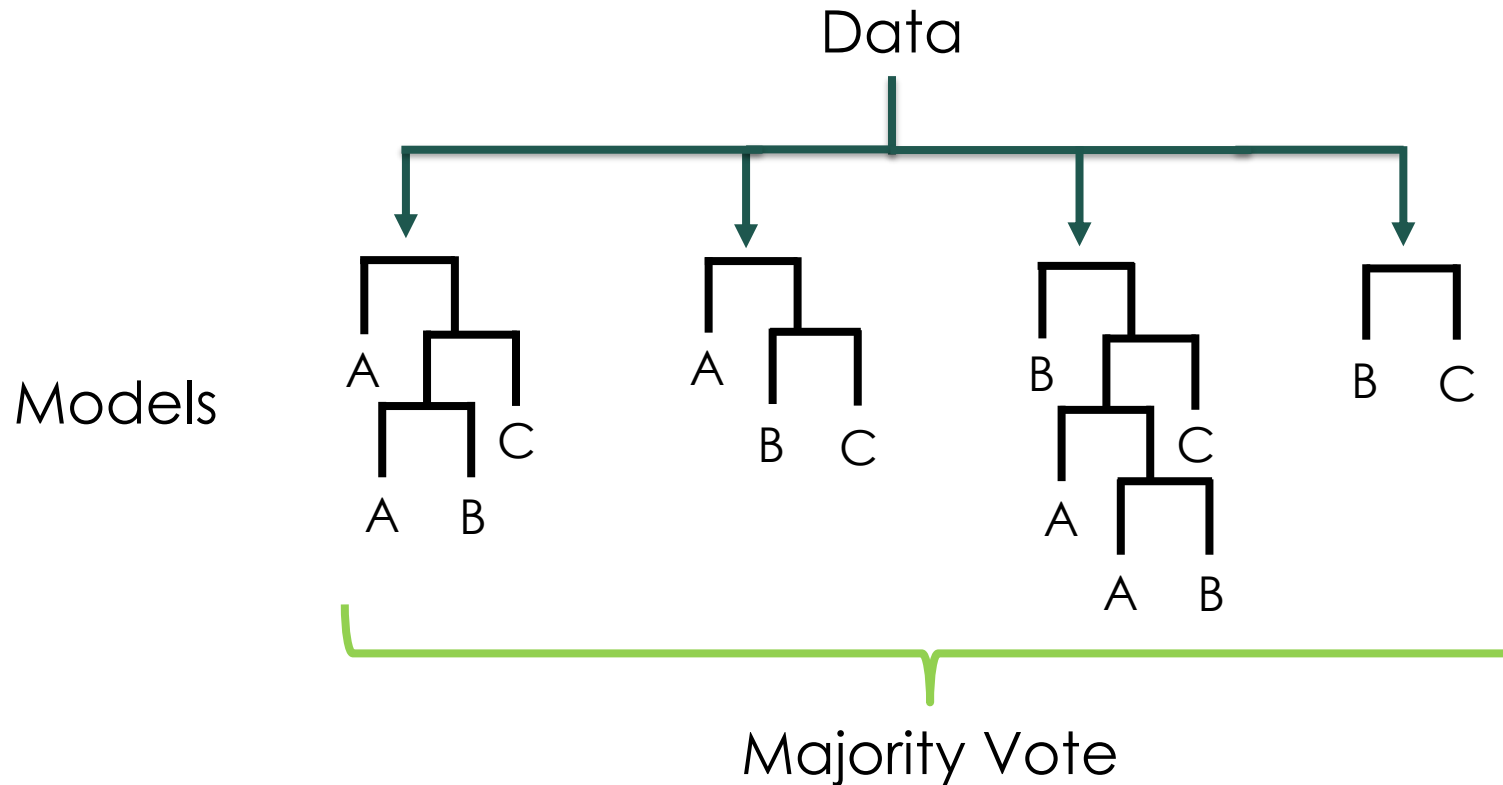
- Important Terms:

	Species	Barks	Pet	Squeeks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
1	Mouse	No	Yes	Yes	No	No	Yes	0.14	0.07	Yes
2	Mouse	No	No	Yes	No	No	Yes	0.18	0.09	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
5	Mouse	No	No	Yes	No	No	Yes	0.13	0.07	Yes
6	Mouse	No	No	Yes	No	No	Yes	0.11	0.05	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
12	Cat	No	Yes	No	No	Yes	Yes	0.40	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
15	Cat	No	Yes	No	Yes	Yes	No	0.30	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
23	Dog	No	No	No	No	No	No	0.52	0.26	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes



# Random Forests: Basics

- A “forest” of binary decision trees (Breiman, 2001)
- Works well with high-dimensional datasets (continuous or categorical)
- Ensemble learning technique



# Random Forests: Basics

- Random Forests vs. CART

	CART	Random Forests
Number of Trees	1	$n \gg 1$
Pruning	Applied	All fully grown
Variables tested for splitting	All	$m \ll M$ (all variables)
Training Set	All	$\approx 2/3$
Accuracy	Independent required	Internally estimated (OOBE)

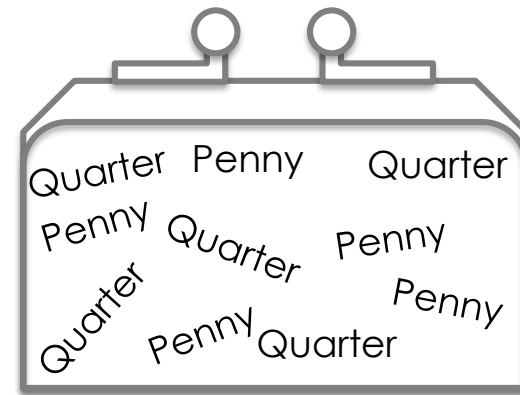
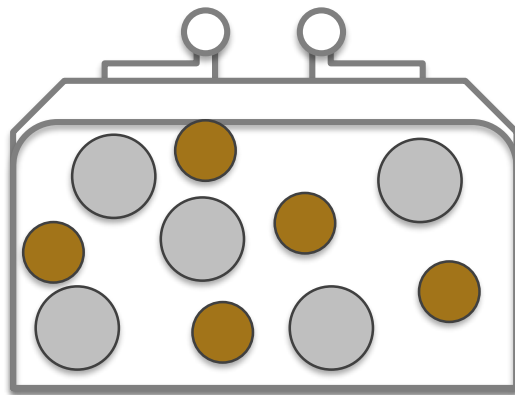
# Random Forests: Parameters and Important Terms

- `ntree` – number of trees per forest.
- Probability – number of trees that voted with the majority divided by the total number of trees.
- `mtry` – number of variables tested to determine optimal split at each node.
- Out of Bag Accuracy – internal validation; based on  $\approx 1/3$  of the dataset not used during the construction of a given tree.
- Mean Decrease in Accuracy (MDA) – quantifies variable “importance” by measuring the change in accuracy after the values of the variable are randomly permuted.

# Random Forests: Basics

- Gini Impurity: probability of classifying a data point incorrectly
- Randomly pick data point & randomly classify it according to class distribution
- $25\% + 25\% = 50\%$ ; Gini Impurity = 0.5

Event	Probability
Choose quarter (50%), classify penny (50%)	$50\% \times 50\% = 25\%$
Choose penny (50%), classify quarter (50%)	$50\% \times 50\% = 25\%$

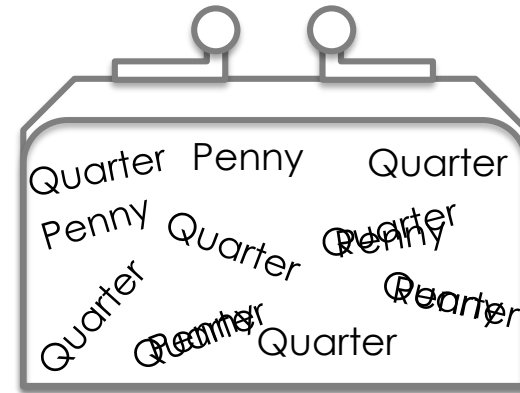
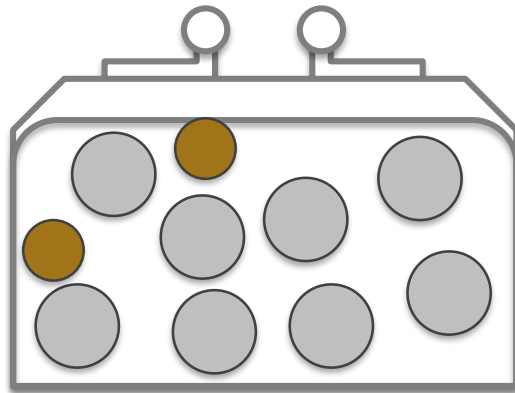




# Random Forests: Basics

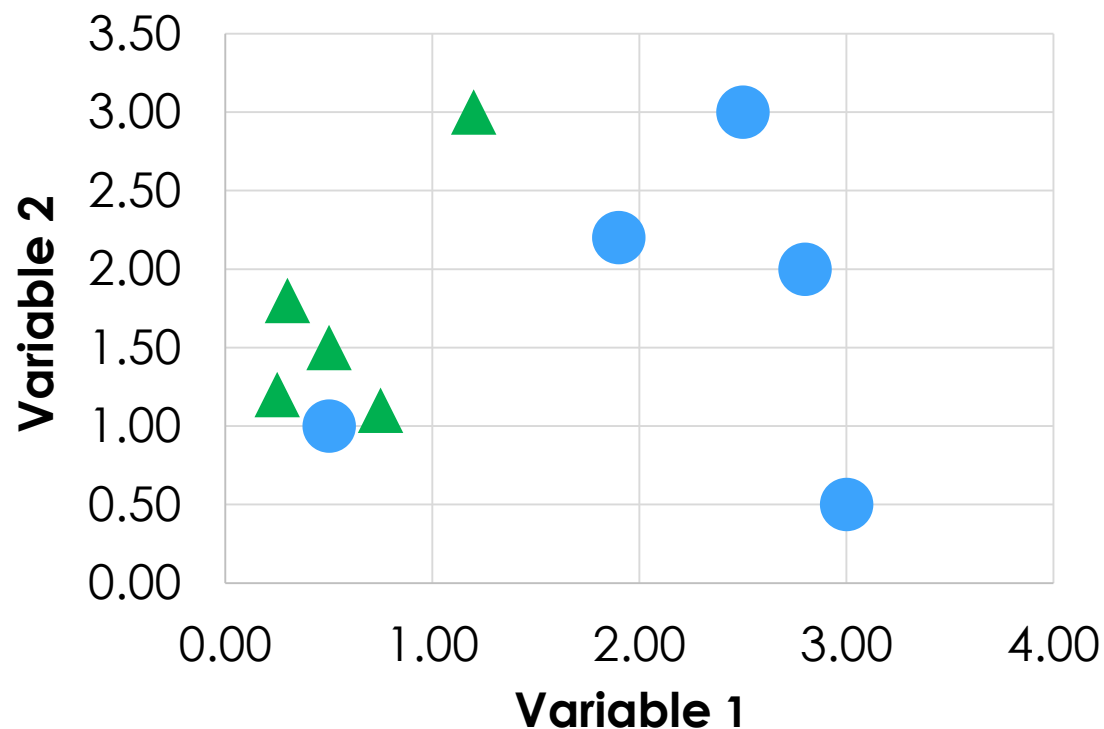
- Gini Impurity: probability of classifying a data point incorrectly
- Randomly pick data point & randomly classify it according to class distribution
- $16\% + 16\% = 32\%$ ; Gini Impurity = 0.32

Event	Probability
Choose quarter (80%), classify penny (20%)	$80\% \times 20\% = 16\%$
Choose penny (20%), classify quarter (80%)	$20\% \times 80\% = 16\%$



# Random Forests: Basics

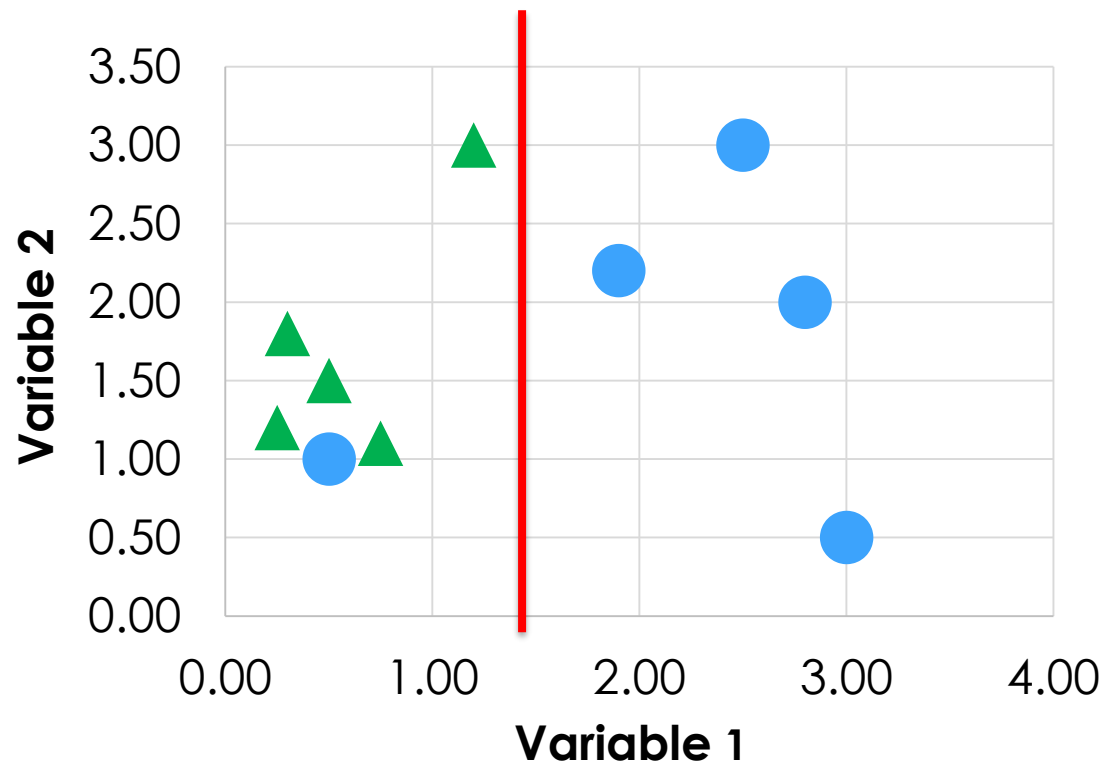
- How Gini Impurity is used for splitting:
  - Don't know where the perfect split is, but can test all possible splits
  - Determine split quality by weighting impurity of subsequent nodes by number of data points it has



# Random Forests: Basics

- How Gini Impurity is used for splitting:
  - Don't know where the perfect split is, but can test all possible splits
  - Determine split quality by weighting impurity of subsequent nodes by number of data points it has

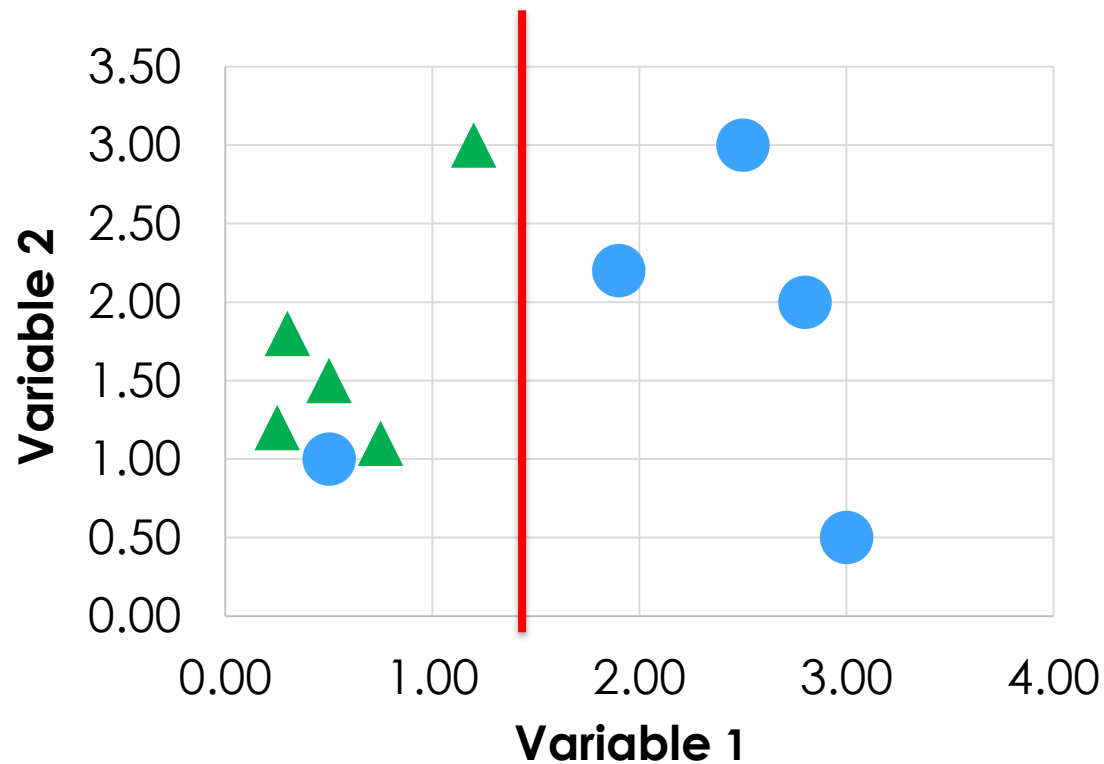
Gini Impurities  
Before the Split = 0.50  
Right Node = 0.00  
Left Node = 0.28



# Random Forests: Basics

- How Gini Impurity is used for splitting:
  - Don't know where the perfect split is, but can test all possible splits
  - Determine split quality by weighting impurity of subsequent nodes by number of data points it has

Gini Impurities  
Before the Split = 0.50  
Right Node = 0.00  
Left Node = 0.28

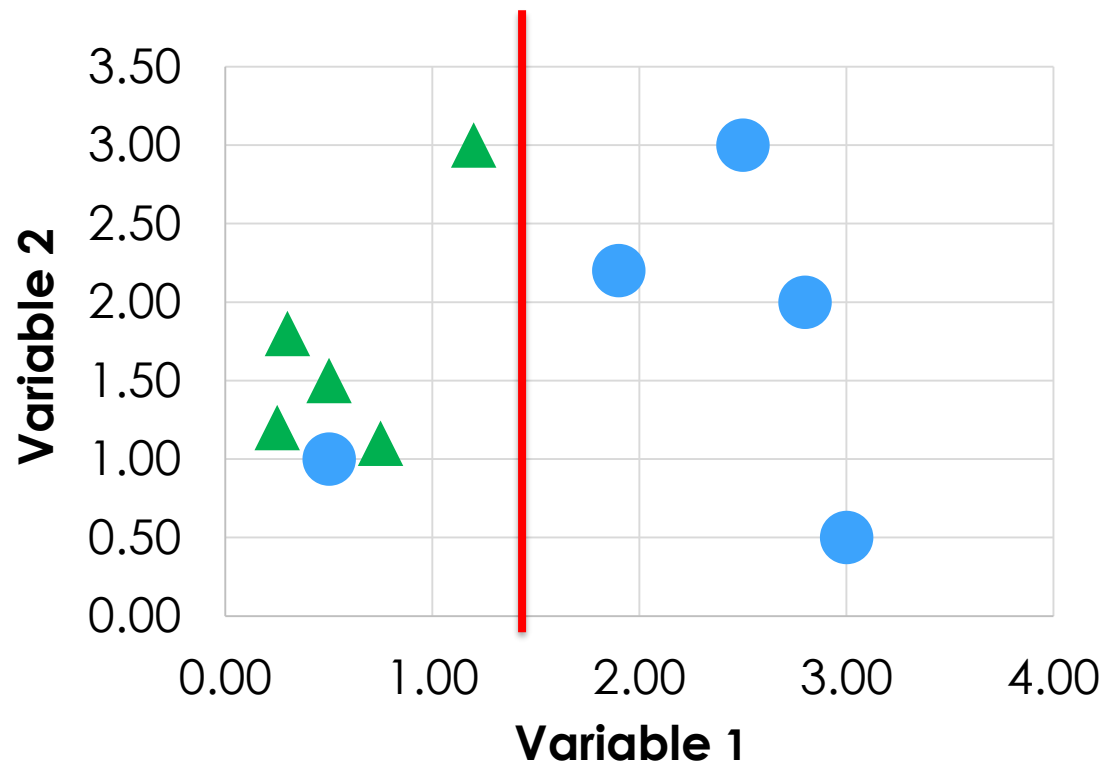


Right node has 40% data points, left node has 60%  
 $(0.40 * 0.00) + (0.60 * 0.28) = 0.17$   
With this split, the amount of impurity removed is:  
 $0.5 - 0.17 = 0.33$

# Random Forests: Basics

- How Gini Impurity is used for splitting:
  - Don't know where the perfect split is, but can test all possible splits
  - Determine split quality by weighting impurity of subsequent nodes by number of data points it has

Gini Impurities  
Before the Split = 0.50  
Right Node = 0.00  
Left Node = 0.28



Right node has 40% data points, left node has 60%

$$(0.40 \times 0.00) + (0.60 \times 0.28) = 0.17$$

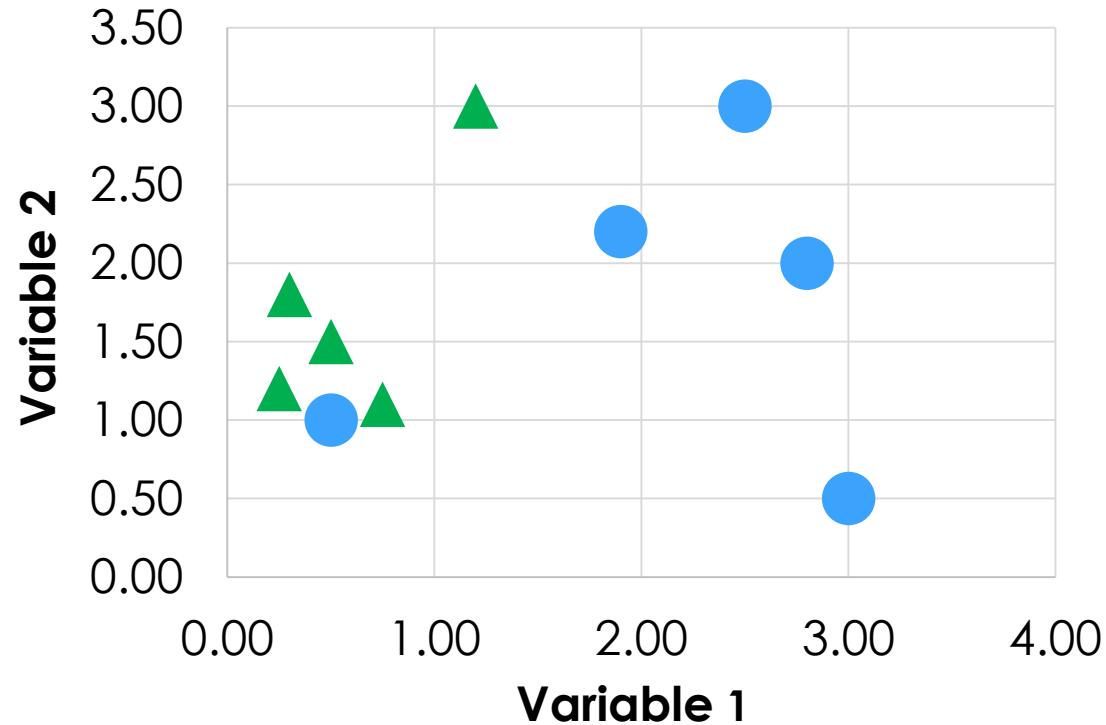
With this split, the amount of impurity removed is:

$$0.5 - 0.17 = 0.33$$

**GINI GAIN**

# Random Forests: Basics

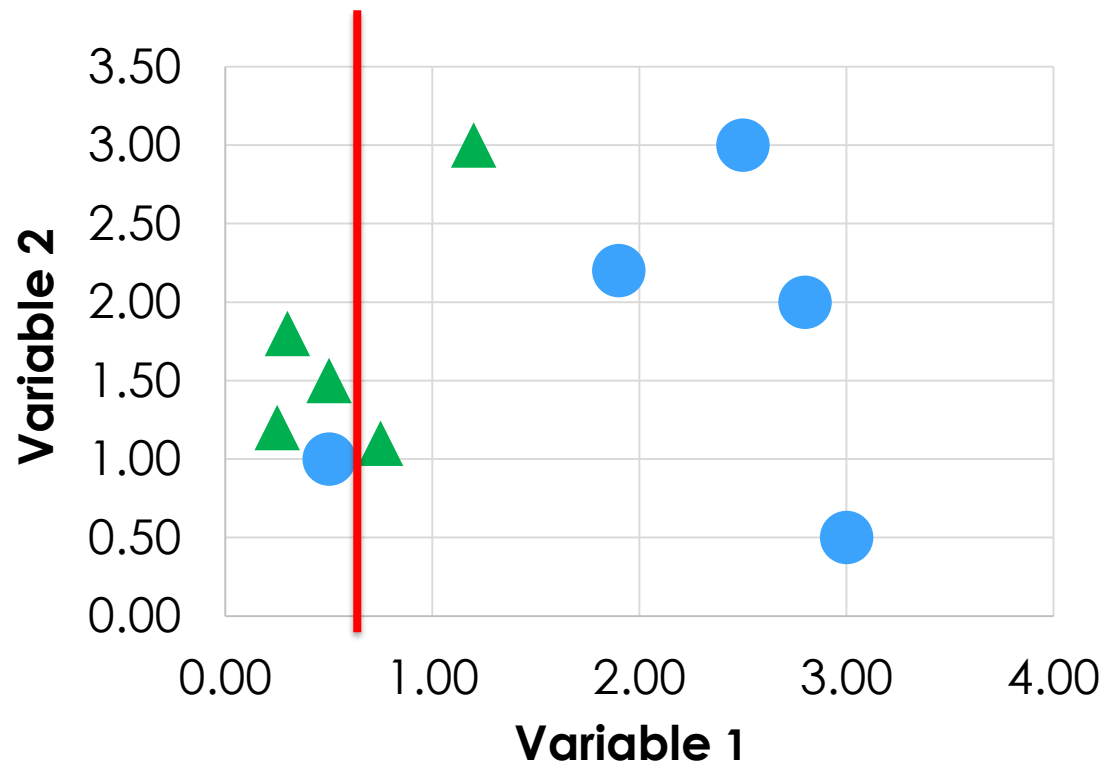
- How Gini Impurity is used for splitting:
  - Don't know where the perfect split is, but can test all possible splits
  - Determine split quality by weighting impurity of subsequent nodes by number of data points it has



# Random Forests: Basics

- How Gini Impurity is used for splitting:
  - Don't know where the perfect split is, but can test all possible splits
  - Determine split quality by weighting impurity of subsequent nodes by number of data points it has

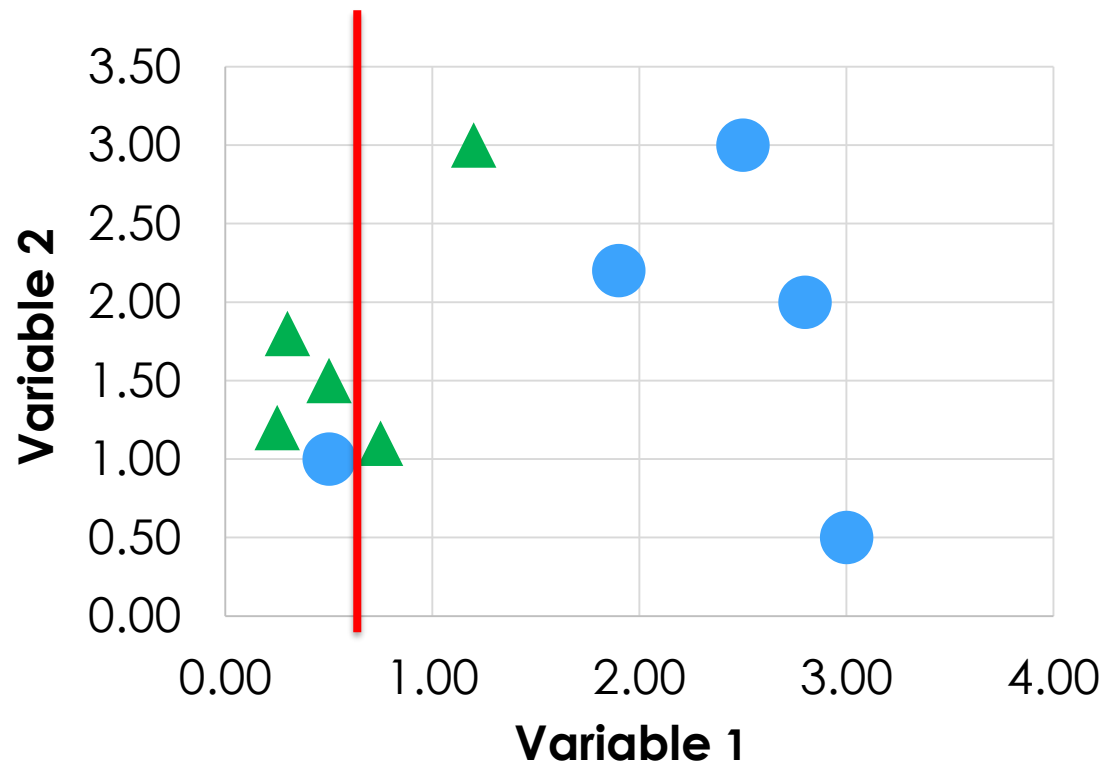
Gini Impurities  
Before the Split = 0.50  
Right Node = 0.44  
Left Node = 0.38



# Random Forests: Basics

- How Gini Impurity is used for splitting:
  - Don't know where the perfect split is, but can test all possible splits
  - Determine split quality by weighting impurity of subsequent nodes by number of data points it has

Gini Impurities  
Before the Split = 0.50  
Right Node = 0.44  
Left Node = 0.38



Right node has 60% data points, left node has 40%

$$(0.60 * 0.38) + (0.40 * 0.44) = 0.42$$

With this split the amount of impurity removed is:

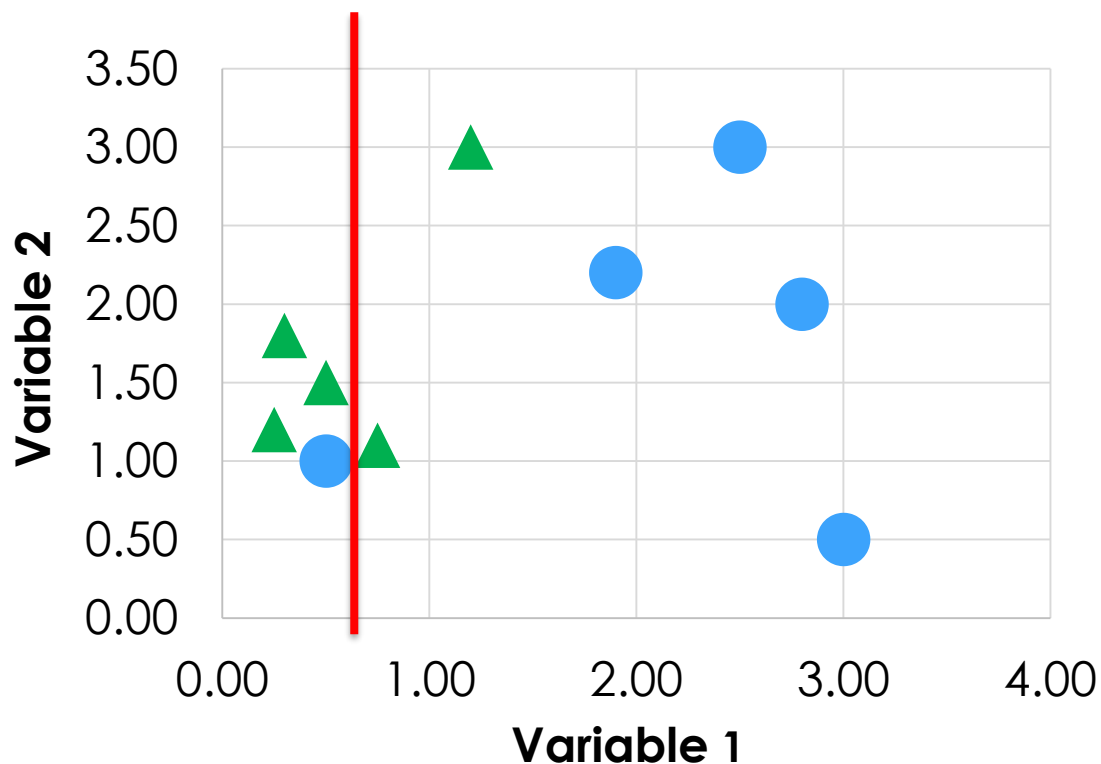
$$0.5 - 0.42 = 0.08$$



# Random Forests: Basics

- How Gini Impurity is used for splitting:
  - Don't know where the perfect split is, but can test all possible splits
  - Determine split quality by weighting impurity of subsequent nodes by number of data points it has

Gini Impurities  
Before the Split = 0.50  
Right Node = 0.44  
Left Node = 0.38



Right node has 60% data points, left node has 40%  
 $(0.60 * 0.38) + (0.40 * 0.44) = 0.42$

With this split the amount of impurity removed is:

$$0.5 - 0.42 = 0.08$$

**GINI GAIN**

# Random Forests: Basics

- Random Forests Inputs:
- Labelled classes with associated predictor variables
  - Categorical or continuous

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
1	Mouse	No	Yes	Yes	No	No	Yes	0.14	0.07	Yes
2	Mouse	No	No	Yes	No	No	Yes	0.18	0.09	Yes
3	Mouse	No	No	Yes	No	No	Yes	0.13	0.06	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
5	Mouse	No	No	Yes	No	No	Yes	0.13	0.07	Yes
6	Mouse	No	No	Yes	No	No	Yes	0.11	0.05	Yes
7	Mouse	No	No	Yes	No	No	Yes	0.13	0.06	Yes
8	Mouse	No	No	Yes	No	No	Yes	0.16	0.08	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
10	Cat	No	Yes	No	Yes	Yes	Yes	0.31	0.19	Yes
11	Cat	No	Yes	No	Yes	No	Yes	0.38	0.20	Yes
12	Cat	No	Yes	No	Yes	Yes	Yes	0.40	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
15	Cat	No	No	No	Yes	No	No	0.32	0.22	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
17	Cat	No	Yes	No	Yes	Yes	Yes	0.35	0.24	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
19	Dog	Yes	No	No	No	No	No	0.58	0.33	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
23	Dog	No	No	No	No	No	No	0.52	0.26	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
25	Dog	Yes	Yes	No	No	Yes	No	0.37	0.16	Yes
26	Dog	Yes	Yes	No	No	Yes	No	0.53	0.29	Yes
27	Dog	Yes	Yes	No	No	Yes	No	0.58	0.29	Yes



# Random Forests: Example

1. Create the training data to grow the tree
  - For N number of data points, randomly sample N cases (with replacement)

# Random Forests: Example

Original

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
1	Mouse	No	Yes	Yes	No	No	Yes	0.14	0.07	Yes
2	Mouse	No	No	Yes	No	No	Yes	0.18	0.09	Yes
3	Mouse	No	No	Yes	No	No	Yes	0.13	0.06	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
5	Mouse	No	No	Yes	No	No	Yes	0.13	0.07	Yes
6	Mouse	No	No	Yes	No	No	Yes	0.11	0.05	Yes
7	Mouse	No	No	Yes	No	No	Yes	0.13	0.06	Yes
8	Mouse	No	No	Yes	No	No	Yes	0.16	0.08	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
10	Cat	No	Yes	No	Yes	Yes	Yes	0.31	0.19	Yes
11	Cat	No	Yes	No	Yes	No	Yes	0.38	0.20	Yes
12	Cat	No	Yes	No	Yes	Yes	Yes	0.40	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
15	Cat	No	No	No	Yes	No	No	0.32	0.22	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
17	Cat	No	Yes	No	Yes	Yes	Yes	0.35	0.24	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
19	Dog	Yes	No	No	No	No	No	0.58	0.33	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
23	Dog	No	No	No	No	No	No	0.52	0.26	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
25	Dog	Yes	Yes	No	No	Yes	No	0.37	0.16	Yes
26	Dog	Yes	Yes	No	No	Yes	No	0.53	0.29	Yes
27	Dog	Yes	Yes	No	No	Yes	No	0.58	0.29	Yes

Training Set

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
--	---------	-------	-----	---------	-------	--------	----------------	--------	--------	--------------

# Random Forests: Example

Original

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
1	Mouse	No	Yes	Yes	No	No	Yes	0.14	0.07	Yes
2	Mouse	No	No	Yes	No	No	Yes	0.18	0.09	Yes
3	Mouse	No	No	Yes	No	No	Yes	0.13	0.06	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
5	Mouse	No	No	Yes	No	No	Yes	0.13	0.07	Yes
6	Mouse	No	No	Yes	No	No	Yes	0.11	0.05	Yes
7	Mouse	No	No	Yes	No	No	Yes	0.13	0.06	Yes
8	Mouse	No	No	Yes	No	No	Yes	0.16	0.08	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
10	Cat	No	Yes	No	Yes	Yes	Yes	0.31	0.19	Yes
11	Cat	No	Yes	No	Yes	No	Yes	0.38	0.20	Yes
12	Cat	No	Yes	No	Yes	Yes	Yes	0.40	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
15	Cat	No	No	No	Yes	No	No	0.32	0.22	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
17	Cat	No	Yes	No	Yes	Yes	Yes	0.35	0.24	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
19	Dog	Yes	No	No	No	No	No	0.58	0.33	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
23	Dog	No	No	No	No	No	No	0.52	0.26	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
25	Dog	Yes	Yes	No	No	Yes	No	0.37	0.16	Yes
26	Dog	Yes	Yes	No	No	Yes	No	0.53	0.29	Yes
27	Dog	Yes	Yes	No	No	Yes	No	0.58	0.29	Yes

Training Set

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes



# Random Forests: Example

Original

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
1	Mouse	No	Yes	Yes	No	No	Yes	0.14	0.07	Yes
2	Mouse	No	No	Yes	No	No	Yes	0.18	0.09	Yes
3	Mouse	No	No	Yes	No	No	Yes	0.13	0.06	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
5	Mouse	No	No	Yes	No	No	Yes	0.13	0.07	Yes
6	Mouse	No	No	Yes	No	No	Yes	0.11	0.05	Yes
7	Mouse	No	No	Yes	No	No	Yes	0.13	0.06	Yes
8	Mouse	No	No	Yes	No	No	Yes	0.16	0.08	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
10	Cat	No	Yes	No	Yes	Yes	Yes	0.31	0.19	Yes
11	Cat	No	Yes	No	Yes	No	Yes	0.38	0.20	Yes
12	Cat	No	Yes	No	Yes	Yes	Yes	0.40	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
15	Cat	No	No	No	Yes	No	No	0.32	0.22	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
17	Cat	No	Yes	No	Yes	Yes	Yes	0.35	0.24	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
19	Dog	Yes	No	No	No	No	No	0.58	0.33	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
23	Dog	No	No	No	No	No	No	0.52	0.26	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
25	Dog	Yes	Yes	No	No	Yes	No	0.37	0.16	Yes
26	Dog	Yes	Yes	No	No	Yes	No	0.53	0.29	Yes
27	Dog	Yes	Yes	No	No	Yes	No	0.58	0.29	Yes

Training Set

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes

# Random Forests: Example

Original

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
1	Mouse	No	Yes	Yes	No	No	Yes	0.14	0.07	Yes
2	Mouse	No	No	Yes	No	No	Yes	0.18	0.09	Yes
3	Mouse	No	No	Yes	No	No	Yes	0.13	0.06	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
5	Mouse	No	No	Yes	No	No	Yes	0.13	0.07	Yes
6	Mouse	No	No	Yes	No	No	Yes	0.11	0.05	Yes
7	Mouse	No	No	Yes	No	No	Yes	0.13	0.06	Yes
8	Mouse	No	No	Yes	No	No	Yes	0.16	0.08	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
10	Cat	No	Yes	No	Yes	Yes	Yes	0.31	0.19	Yes
11	Cat	No	Yes	No	Yes	No	Yes	0.38	0.20	Yes
12	Cat	No	Yes	No	Yes	Yes	Yes	0.40	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
15	Cat	No	No	No	Yes	No	No	0.32	0.22	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
17	Cat	No	Yes	No	Yes	Yes	Yes	0.35	0.24	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
19	Dog	Yes	No	No	No	No	No	0.58	0.33	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
23	Dog	No	No	No	No	No	No	0.52	0.26	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
25	Dog	Yes	Yes	No	No	Yes	No	0.37	0.16	Yes
26	Dog	Yes	Yes	No	No	Yes	No	0.53	0.29	Yes
27	Dog	Yes	Yes	No	No	Yes	No	0.58	0.29	Yes

Training Set

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
27	Dog	Yes	Yes	No	No	Yes	No	0.58	0.29	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
23	Dog	Yes	No	No	No	No	No	0.52	0.26	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
2	Mouse	No	No	Yes	No	No	Yes	0.18	0.09	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.21	Yes
12	Cat	No	Yes	No	Yes	Yes	Yes	0.40	0.15	No
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
5	Mouse	No	No	Yes	No	No	Yes	0.13	0.07	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
27	Dog	Yes	Yes	No	No	Yes	No	0.58	0.29	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
1	Mouse	No	Yes	Yes	No	No	Yes	0.14	0.07	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
6	Mouse	No	No	Yes	No	No	Yes	0.11	0.05	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes



# Random Forests: Example

Original

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
1	Mouse	No	Yes	Yes	No	No	Yes	0.14	0.07	Yes
2	Mouse	No	No	Yes	No	No	Yes	0.18	0.09	Yes
3	Mouse	No	No	Yes	No	No	Yes	0.13	0.06	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
5	Mouse	No	No	Yes	No	No	Yes	0.13	0.07	Yes
6	Mouse	No	No	Yes	No	No	Yes	0.11	0.05	Yes
7	Mouse	No	No	Yes	No	No	Yes	0.13	0.06	Yes
8	Mouse	No	No	Yes	No	No	Yes	0.16	0.08	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
10	Cat	No	Yes	No	Yes	Yes	Yes	0.31	0.19	Yes
11	Cat	No	Yes	No	Yes	No	Yes	0.38	0.20	Yes
12	Cat	No	Yes	No	Yes	Yes	Yes	0.40	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
15	Cat	No	No	No	Yes	No	No	0.32	0.22	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
17	Cat	No	Yes	No	Yes	Yes	Yes	0.35	0.24	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
19	Dog	Yes	No	No	No	No	No	0.58	0.33	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
23	Dog	No	No	No	No	No	No	0.52	0.26	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
25	Dog	Yes	Yes	No	No	Yes	No	0.37	0.16	Yes
26	Dog	Yes	Yes	No	No	Yes	No	0.53	0.29	Yes
27	Dog	Yes	Yes	No	No	Yes	No	0.58	0.29	Yes

Training Set

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
1	Mouse	No	Yes	Yes	No	No	Yes	0.14	0.07	Yes
2	Mouse	No	No	Yes	No	No	Yes	0.18	0.09	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
5	Mouse	No	No	Yes	No	No	Yes	0.13	0.07	Yes
6	Mouse	No	No	Yes	No	No	Yes	0.11	0.05	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
12	Cat	No	Yes	No	Yes	Yes	Yes	0.4	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.21	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
23	Dog	Yes	No	No	No	No	No	0.52	0.26	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
27	Dog	Yes	Yes	No	No	Yes	No	0.58	0.29	Yes
27	Dog	Yes	Yes	No	No	Yes	No	0.58	0.29	Yes

Remaining for internal accuracy assessment  $\approx 1/3$





# Random Forests: Example

- For M number of variables, randomly select a subset ( $m \ll M$ ) to determine how each node is split (**mtry**)

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
1	Mouse	No	Yes	Yes	No	No	Yes	0.14	0.07	Yes
2	Mouse	No	No	Yes	No	No	Yes	0.18	0.09	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
5	Mouse	No	No	Yes	No	No	Yes	0.13	0.07	Yes
6	Mouse	No	No	Yes	No	No	Yes	0.11	0.05	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
12	Cat	No	Yes	No	No	Yes	Yes	0.4	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
23	Dog	No	No	No	No	No	No	0.52	0.26	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes

# Random Forests: Example

## Training Set

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
1	Mouse	No	Yes	Yes	No	No	Yes	0.14	0.07	Yes
2	Mouse	No	No	Yes	No	No	Yes	0.18	0.09	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
5	Mouse	No	No	Yes	No	No	Yes	0.13	0.07	Yes
6	Mouse	No	No	Yes	No	No	Yes	0.11	0.05	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
12	Cat	No	Yes	No	No	Yes	Yes	0.4	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
23	Dog	No	No	No	No	No	No	0.52	0.26	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes

## Classification Results Based on Split Points

	Mouse	Cat	Dog
Yes	8	0	0
No	0	8	11

## Gini Gain to Identify Optimal Split

Variable	Evaluated Split	
	Points	Gini
Squeaks	Yes or No	0.32

# Random Forests: Example

## Training Set

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
1	Mouse	No	Yes	Yes	No	No	Yes	0.14	0.07	Yes
2	Mouse	No	No	Yes	No	No	Yes	0.18	0.09	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
5	Mouse	No	No	Yes	No	No	Yes	0.13	0.07	Yes
6	Mouse	No	No	Yes	No	No	Yes	0.11	0.05	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
12	Cat	No	Yes	No	No	Yes	Yes	0.4	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
23	Dog	No	No	No	No	No	No	0.52	0.26	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes

## Classification Results Based on Split Points

	Mouse	Cat	Dog
Yes	0	7	0
No	8	1	11

## Gini Gain to Identify Optimal Split

Variable	Evaluated Split	
	Points	Gini
Squeaks	Yes or No	0.32
Meows	Yes or No	0.28

# Random Forests: Example

## Training Set

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
1	Mouse	No	Yes	Yes	No	No	Yes	0.14	0.07	Yes
2	Mouse	No	No	Yes	No	No	Yes	0.18	0.09	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
5	Mouse	No	No	Yes	No	No	Yes	0.13	0.07	Yes
6	Mouse	No	No	Yes	No	No	Yes	0.11	0.05	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
12	Cat	No	Yes	No	No	Yes	Yes	0.4	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
23	Dog	No	No	No	No	No	No	0.52	0.26	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes

## Classification Results Based on Split Points

	Mouse	Cat	Dog
$\leq 0.05$	1	0	0
$> 0.05$	7	8	11

## Gini Gain to Identify Optimal Split

Variable	Evaluated Split	
	Points	Gini
Squeaks	Yes or No	0.32
Meows	Yes or No	0.28
Height	$\leq 0.05$	0.03

# Random Forests: Example

## Training Set

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
1	Mouse	No	Yes	Yes	No	No	Yes	0.14	0.07	Yes
2	Mouse	No	No	Yes	No	No	Yes	0.18	0.09	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
5	Mouse	No	No	Yes	No	No	Yes	0.13	0.07	Yes
6	Mouse	No	No	Yes	No	No	Yes	0.11	0.05	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
12	Cat	No	Yes	No	No	Yes	Yes	0.4	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
23	Dog	No	No	No	No	No	No	0.52	0.26	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes

## Classification Results Based on Split Points

	Mouse	Cat	Dog
$\leq 0.06$	3	0	0
$> 0.06$	5	8	11

## Gini Gain to Identify Optimal Split

Variable	Evaluated Split	
	Points	Gini
Squeaks	Yes or No	0.32
Meows	Yes or No	0.28
Height	$\leq 0.05$	0.03
Height	$\leq 0.06$	0.09

# Random Forests: Example

## Training Set

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
1	Mouse	No	Yes	Yes	No	No	Yes	0.14	0.07	Yes
2	Mouse	No	No	Yes	No	No	Yes	0.18	0.09	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
5	Mouse	No	No	Yes	No	No	Yes	0.13	0.07	Yes
6	Mouse	No	No	Yes	No	No	Yes	0.11	0.05	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
12	Cat	No	Yes	No	No	Yes	Yes	0.4	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
23	Dog	No	No	No	No	No	No	0.52	0.26	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes

## Classification Results Based on Split Points

	Mouse	Cat	Dog
$\leq 0.07$	5	0	0
$> 0.07$	3	8	11

## Gini Gain to Identify Optimal Split

Variable	Evaluated Split	
	Points	Gini
Squeaks	Yes or No	0.32
Meows	Yes or No	0.28
Height	$\leq 0.05$	0.03
Height	$\leq 0.06$	0.09
Height	$\leq 0.07$	0.17

...

# Random Forests: Example

## Training Set

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
1	Mouse	No	Yes	Yes	No	No	Yes	0.14	0.07	Yes
2	Mouse	No	No	Yes	No	No	Yes	0.18	0.09	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
5	Mouse	No	No	Yes	No	No	Yes	0.13	0.07	Yes
6	Mouse	No	No	Yes	No	No	Yes	0.11	0.05	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
12	Cat	No	Yes	No	No	Yes	Yes	0.4	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
23	Dog	No	No	No	No	No	No	0.52	0.26	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes

## Gini Gain to Identify Optimal Split

Variable	Evaluated Split Points	Gini
Squeaks	Yes or No	0.32
Meows	Yes or No	0.28
Height	$\leq 0.05$	0.03
Height	$\leq 0.06$	0.09
Height	$\leq 0.07$	0.17
Height	$\leq 0.08$	0.26
Height	$\leq 0.09$	0.28
Height	$\leq 0.15$	0.25
Height	$\leq 0.16$	0.26
Height	$\leq 0.17$	0.28
Height	$\leq 0.22$	0.27
Height	$\leq 0.26$	0.31
Height	$\leq 0.27$	0.22
Height	$\leq 0.29$	0.18
Height	$\leq 0.32$	0.09
Height	$\leq 0.33$	0.04
Height	$\leq 0.35$	0.00

# Random Forests: Example

## Training Set

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
1	Mouse	No	Yes	Yes	No	No	Yes	0.14	0.07	Yes
2	Mouse	No	No	Yes	No	No	Yes	0.18	0.09	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
5	Mouse	No	No	Yes	No	No	Yes	0.13	0.07	Yes
6	Mouse	No	No	Yes	No	No	Yes	0.11	0.05	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
12	Cat	No	Yes	No	No	Yes	Yes	0.4	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
23	Dog	No	No	No	No	No	No	0.52	0.26	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes

## Gini Gain to Identify Optimal Split

Variable	Evaluated Split Points	Gini
Squeaks	Yes or No	0.32
Meows	Yes or No	0.28
Height	$\leq 0.05$	0.03
Height	$\leq 0.06$	0.09
Height	$\leq 0.07$	0.17
Height	$\leq 0.08$	0.26
Height	$\leq 0.09$	0.28
Height	$\leq 0.15$	0.25
Height	$\leq 0.16$	0.26
Height	$\leq 0.17$	0.28
Height	$\leq 0.22$	0.27
Height	$\leq 0.26$	0.31
Height	$\leq 0.27$	0.22
Height	$\leq 0.29$	0.18
Height	$\leq 0.32$	0.09
Height	$\leq 0.33$	0.04
Height	$\leq 0.35$	0.00



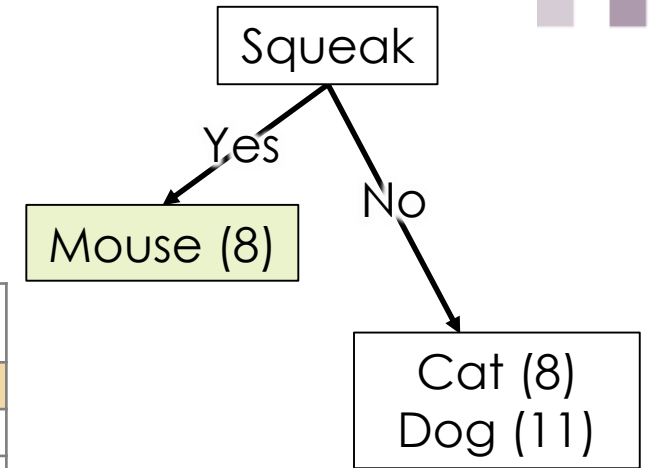
# Random Forests: Example

Training Set

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
1	Mouse	No	Yes	Yes	No	No	Yes	0.14	0.07	Yes
2	Mouse	No	No	Yes	No	No	Yes	0.18	0.09	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
5	Mouse	No	No	Yes	No	No	Yes	0.13	0.07	Yes
6	Mouse	No	No	Yes	No	No	Yes	0.11	0.05	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
12	Cat	No	Yes	No	No	Yes	Yes	0.4	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
23	Dog	No	No	No	No	No	No	0.52	0.26	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes

Gini Gain to Identify Optimal Split

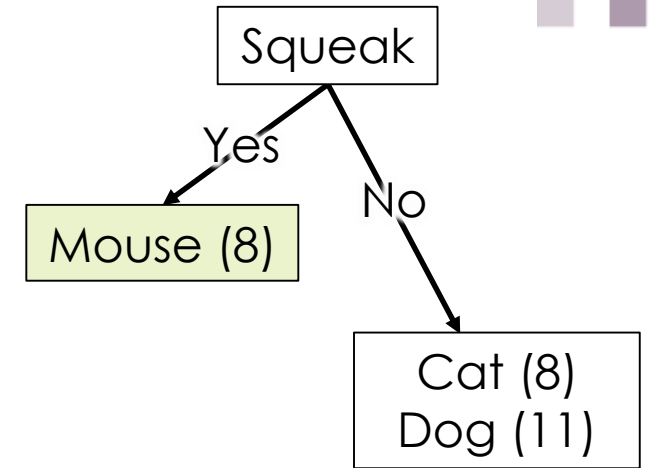
Variable	Evaluated Split Points	Gini
Squeaks	Yes or No	0.32
Meows	Yes or No	0.28
Height	$\leq 0.05$	0.03
Height	$\leq 0.06$	0.09
Height	$\leq 0.07$	0.17
Height	$\leq 0.08$	0.26
Height	$\leq 0.09$	0.28
Height	$\leq 0.15$	0.25
Height	$\leq 0.16$	0.26
Height	$\leq 0.17$	0.28
Height	$\leq 0.22$	0.27
Height	$\leq 0.26$	0.31
Height	$\leq 0.27$	0.22
Height	$\leq 0.29$	0.18
Height	$\leq 0.32$	0.09
Height	$\leq 0.33$	0.04
Height	$\leq 0.35$	0.00



# Random Forests: Example

Training Set

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
12	Cat	No	Yes	No	No	Yes	Yes	0.4	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
23	Dog	No	No	No	No	No	No	0.52	0.26	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes



# Random Forests: Example

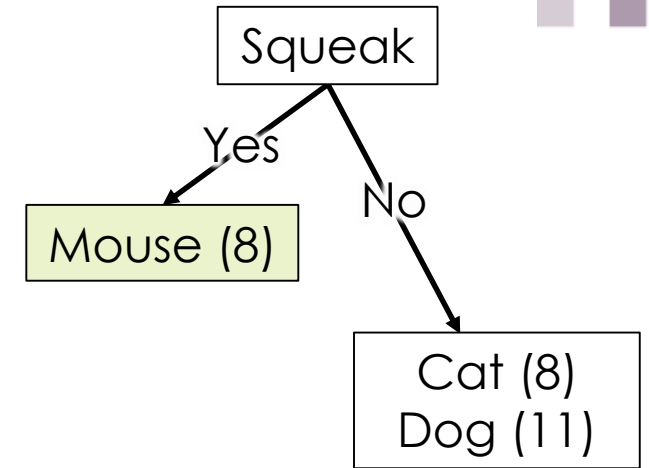
Training Set

Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
---------	-------	-----	---------	-------	--------	----------------	--------	--------	--------------

12	Cat	No	Yes	No	No	Yes	Yes	0.4	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
23	Dog	No	No	No	No	No	No	0.52	0.26	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes

Classification Results Based on Split Points

	Cat	Dog
Yes	1	0
No	7	11



Gini Gain to Identify Optimal Split

Variable	Evaluated Split Points	Gini
Cheese	Yes or No	0.04

# Random Forests: Example

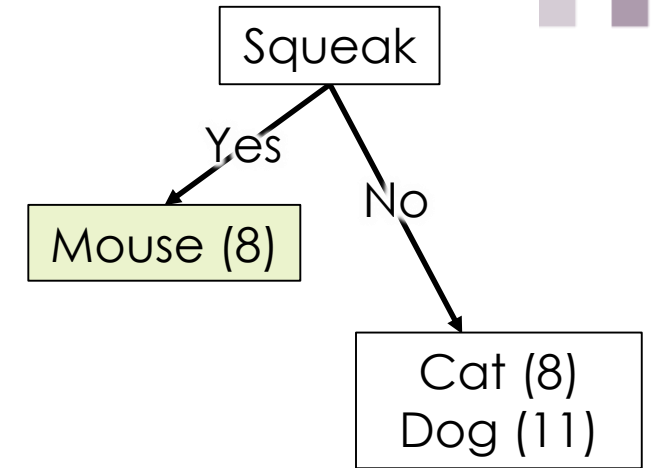
Training Set

Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
---------	-------	-----	---------	-------	--------	----------------	--------	--------	--------------

12	Cat	No	Yes	No	No	Yes	Yes	0.4	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
23	Dog	No	No	No	No	No	No	0.52	0.26	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes

Classification Results Based on Split Points

	Cat	Dog
Yes	0	8
No	8	3



Gini Gain to Identify Optimal Split

Variable	Evaluated Split	
	Points	Gini
Cheese	Yes or No	0.04
Barks	Yes or No	0.26

# Random Forests: Example

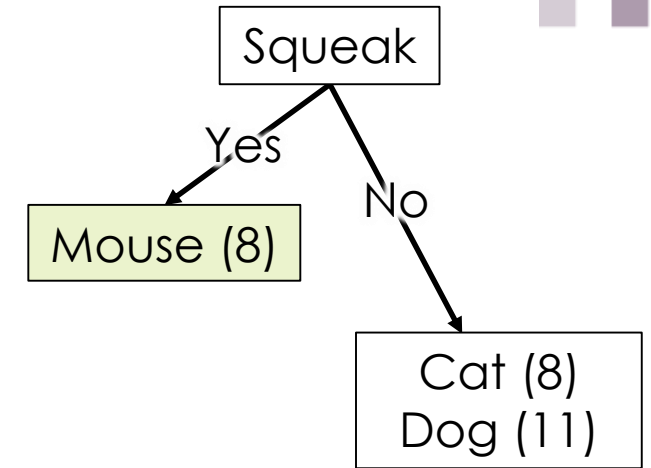
Training Set

Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
---------	-------	-----	---------	-------	--------	----------------	--------	--------	--------------

12	Cat	No	Yes	No	No	Yes	Yes	0.4	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
23	Dog	No	No	No	No	No	No	0.52	0.26	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes

Classification Results Based on Split Points

	Cat	Dog
Yes	8	10
No	0	1



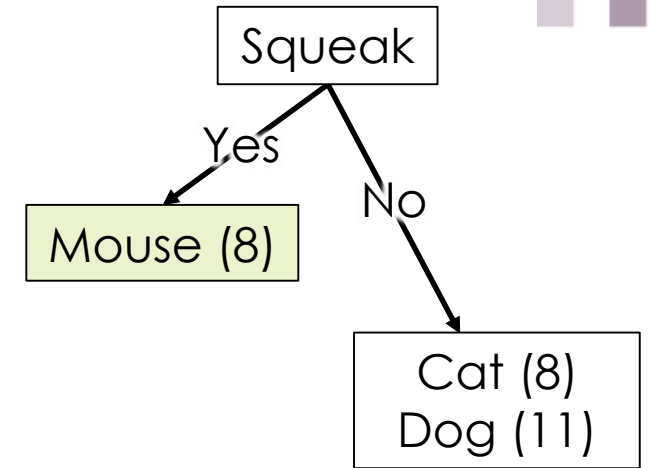
Gini Gain to Identify Optimal Split

Variable	Evaluated Split	
	Points	Gini
Cheese	Yes or No	0.04
Barks	Yes or No	0.26
Collar	Yes or No	0.02

# Random Forests: Example

## Training Set

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
12	Cat	No	Yes	No	No	Yes	Yes	0.4	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
23	Dog	No	No	No	No	No	No	0.52	0.26	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes



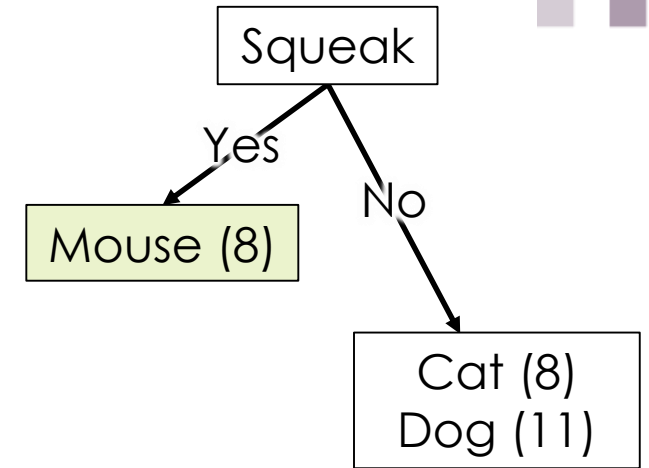
## Gini Gain to Identify Optimal Split

Variable	Evaluated Split	
	Points	Gini
Cheese	Yes or No	0.04
Barks	Yes or No	0.26
Collar	Yes or No	0.02

# Random Forests: Example

## Training Set

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
12	Cat	No	Yes	No	No	Yes	Yes	0.4	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
23	Dog	No	No	No	No	No	No	0.52	0.26	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes



## Gini Gain to Identify Optimal Split

Variable	Evaluated Split	
	Points	Gini
Cheese	Yes or No	0.04
Barks	Yes or No	0.26
Collar	Yes or No	0.02

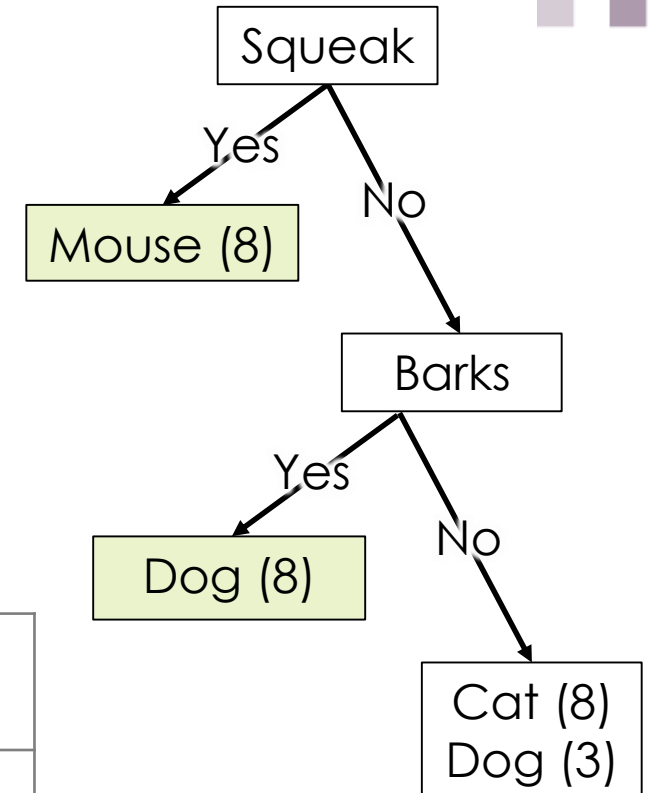
# Random Forests: Example

## Training Set

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
12	Cat	No	Yes	No	No	Yes	Yes	0.4	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
23	Dog	No	No	No	No	No	No	0.52	0.26	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes

## Gini Gain to Identify Optimal Split

Evaluated Split		
Variable	Points	Gini
Cheese	Yes or No	0.04
Barks	Yes or No	0.26
Collar	Yes or No	0.02

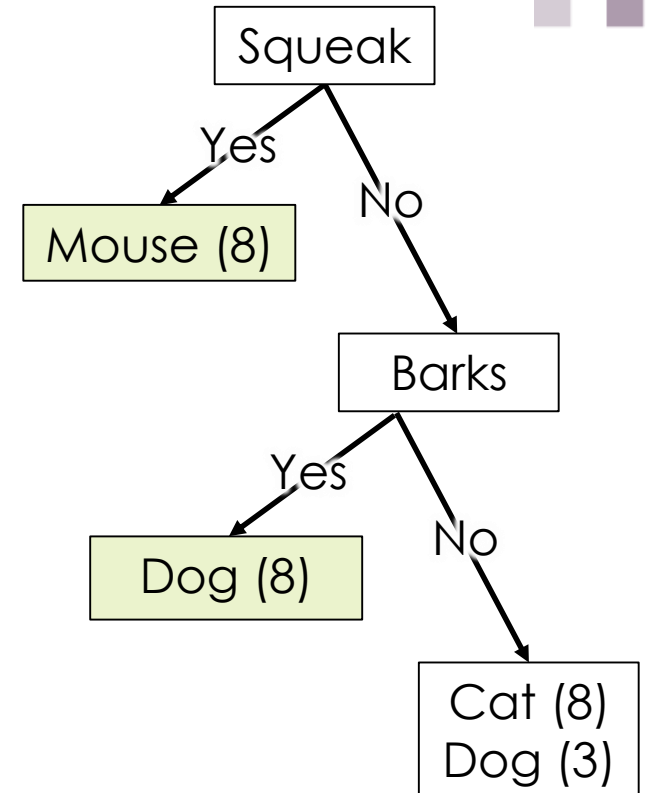




# Random Forests: Example

Training Set

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
12	Cat	No	Yes	No	No	Yes	Yes	0.4	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
23	Dog	No	No	No	No	No	No	0.52	0.26	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes



# Random Forests: Example

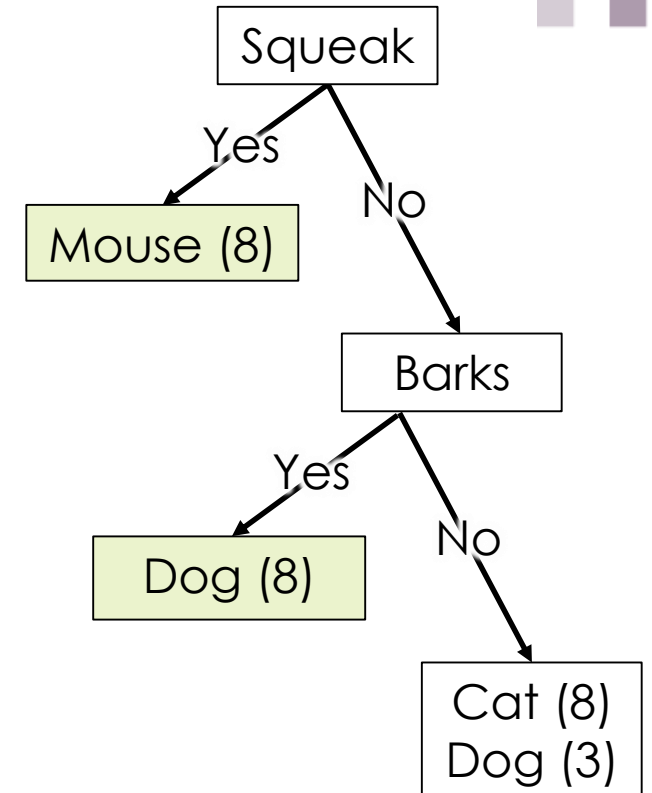
Training Set

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
12	Cat	No	Yes	No	No	Yes	Yes	0.4	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes

23 Dog No No No No No No 0.52 0.26 Yes

27 Dog No Yes No No Yes No 0.58 0.29 Yes

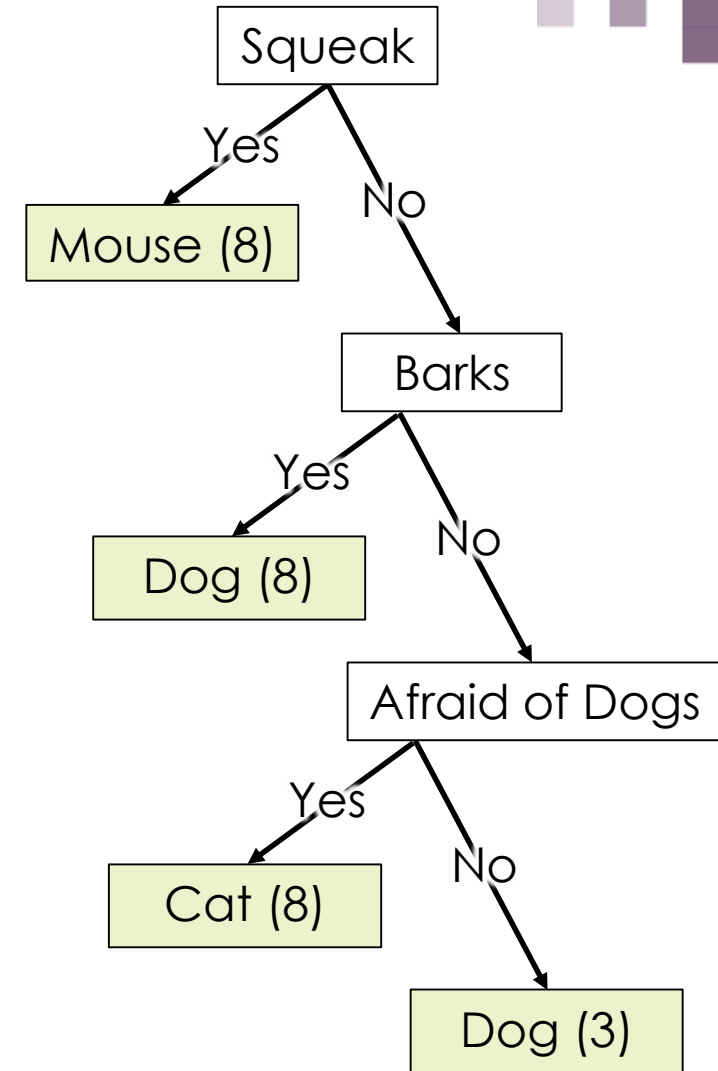
27 Dog No Yes No No Yes No 0.58 0.29 Yes



# Random Forests: Example

Training Set

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
12	Cat	No	Yes	No	No	Yes	Yes	0.4	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.3	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
23	Dog	No	No	No	No	No	No	0.52	0.26	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes
27	Dog	No	Yes	No	No	Yes	No	0.58	0.29	Yes



# Random Forests: Example

3. Take the mode prediction of all trees (**ntree**) to determine the final classification

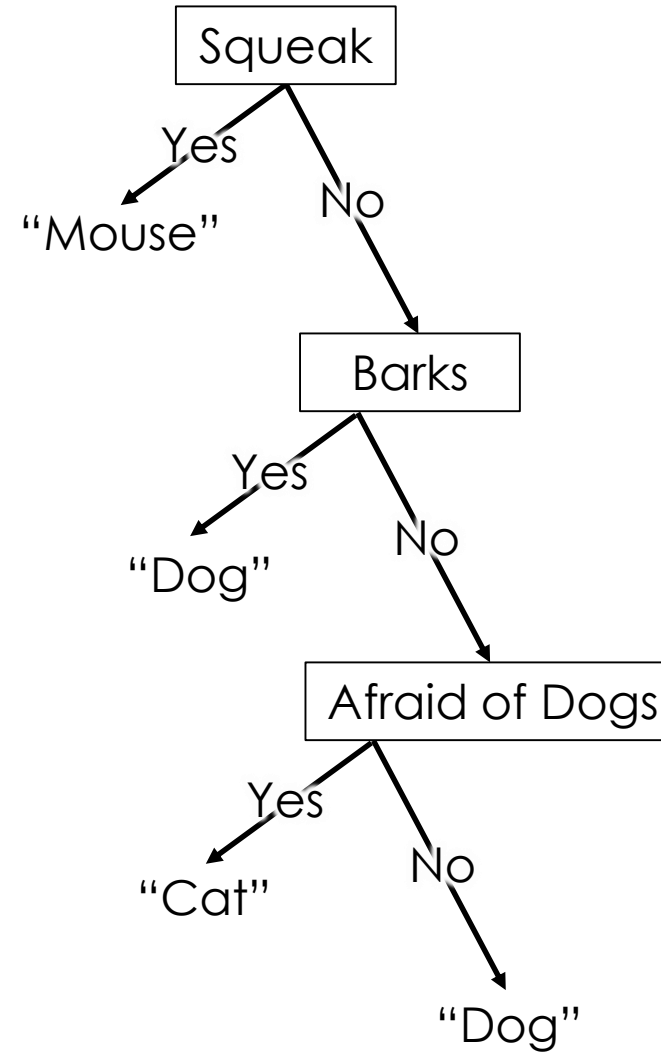
# Random Forests: Example

Original

Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
---------	-------	-----	---------	-------	--------	----------------	--------	--------	--------------



Cat	No	Yes	No	Yes	Yes	No	0.35	0.24	Yes
-----	----	-----	----	-----	-----	----	------	------	-----



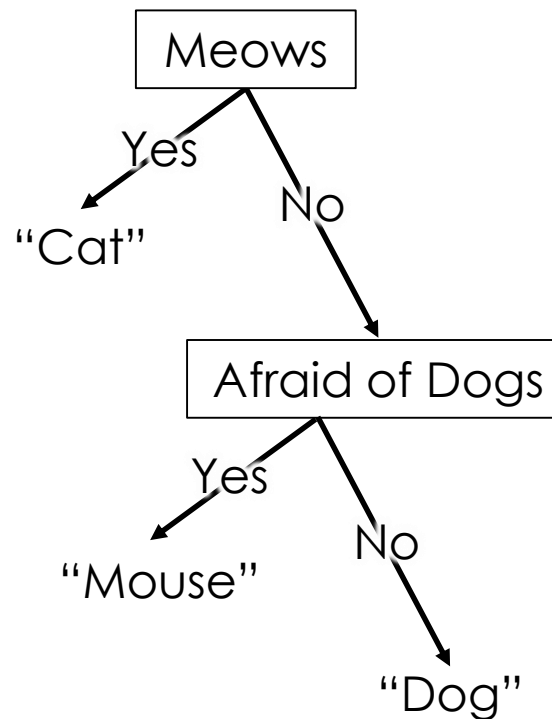
Tree	Vote
1	Dog

# Random Forests: Example

Original

Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
---------	-------	-----	---------	-------	--------	----------------	--------	--------	--------------

	Cat	No	Yes	No	Yes	Yes	No	0.35	0.24	Yes
--	-----	----	-----	----	-----	-----	----	------	------	-----



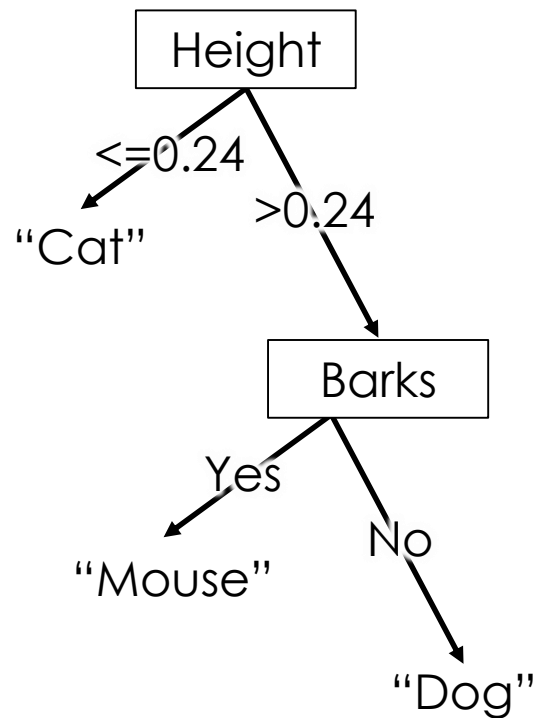
Tree	Vote
1	Dog
2	Cat

# Random Forests: Example

Original

Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
---------	-------	-----	---------	-------	--------	----------------	--------	--------	--------------

	Cat	No	Yes	No	Yes	Yes	No	0.35	0.24	Yes
--	-----	----	-----	----	-----	-----	----	------	------	-----



Tree	Vote
1	Dog
2	Cat
3	Cat

# Random Forests: Example

Original

Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
---------	-------	-----	---------	-------	--------	----------------	--------	--------	--------------

17	Cat	No	Yes	No	Yes	Yes	No	0.35	0.24	Yes
----	-----	----	-----	----	-----	-----	----	------	------	-----

Tree	Vote
1	Dog
2	Cat
3	Cat
4	Mouse
5	Cat
6	Cat
7	Cat
8	Cat
9	Dog

Majority = Cat

Final Classification = Cat



# Random Forests: Example

Original

Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
---------	-------	-----	---------	-------	--------	----------------	--------	--------	--------------

17	Cat	No	Yes	No	Yes	Yes	No	0.35	0.24	Yes
----	-----	----	-----	----	-----	-----	----	------	------	-----

Tree	Vote
1	Dog
2	Cat
3	Cat
4	Mouse
5	Cat
6	Cat
7	Cat
8	Cat
9	Cat

Probability:  
=2/3  
=67%

# Random Forests: Example

Original

Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
---------	-------	-----	---------	-------	--------	----------------	--------	--------	--------------

17	Cat	No	Yes	No	Yes	Yes	No	0.35	0.24	Yes
----	-----	----	-----	----	-----	-----	----	------	------	-----

n<sub>tree</sub> = 9

Tree	Vote
1	Dog
2	Cat
3	Cat
4	Mouse
5	Cat
6	Cat
7	Cat
8	Cat
9	Cat

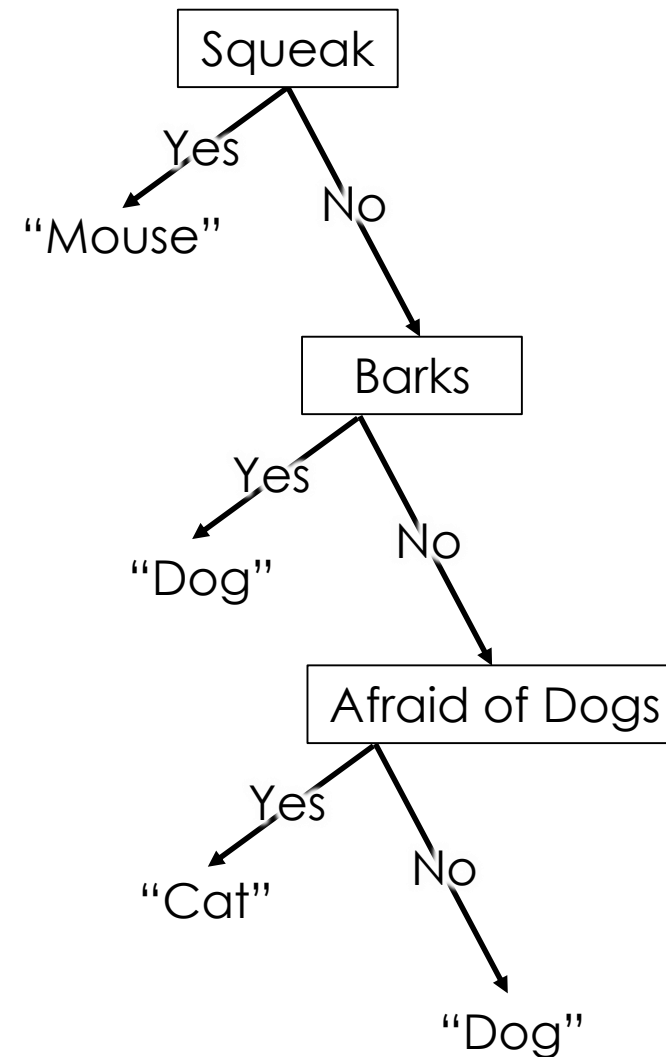
# Random Forests: Example

- Run the Out of Bag Samples down the tree and calculate the Out of Bag Error (for a given tree)

# Random Forests: Example

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
1	Mouse	No	Yes	Yes	No	No	Yes	0.14	0.07	Yes
2	Mouse	No	No	Yes	No	No	Yes	0.18	0.09	Yes
3	Mouse	No	No	Yes	No	No	Yes	0.13	0.06	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
5	Mouse	No	No	Yes	No	No	Yes	0.13	0.07	Yes
6	Mouse	No	No	Yes	No	No	Yes	0.11	0.05	Yes
7	Mouse	No	No	Yes	No	No	Yes	0.13	0.06	Yes
8	Mouse	No	No	Yes	No	No	Yes	0.16	0.08	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
10	Cat	No	Yes	No	Yes	Yes	Yes	0.31	0.19	Yes
11	Cat	No	Yes	No	Yes	No	Yes	0.38	0.20	Yes
12	Cat	No	Yes	No	Yes	Yes	Yes	0.40	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
15	Cat	No	No	No	Yes	No	No	0.32	0.22	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
17	Cat	No	Yes	No	Yes	Yes	Yes	0.35	0.24	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
19	Dog	Yes	No	No	No	No	No	0.58	0.33	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
23	Dog	No	No	No	No	No	No	0.52	0.26	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
25	Dog	Yes	Yes	No	No	Yes	No	0.37	0.16	Yes
26	Dog	Yes	Yes	No	No	Yes	No	0.53	0.29	Yes
27	Dog	Yes	Yes	No	No	Yes	No	0.58	0.29	Yes

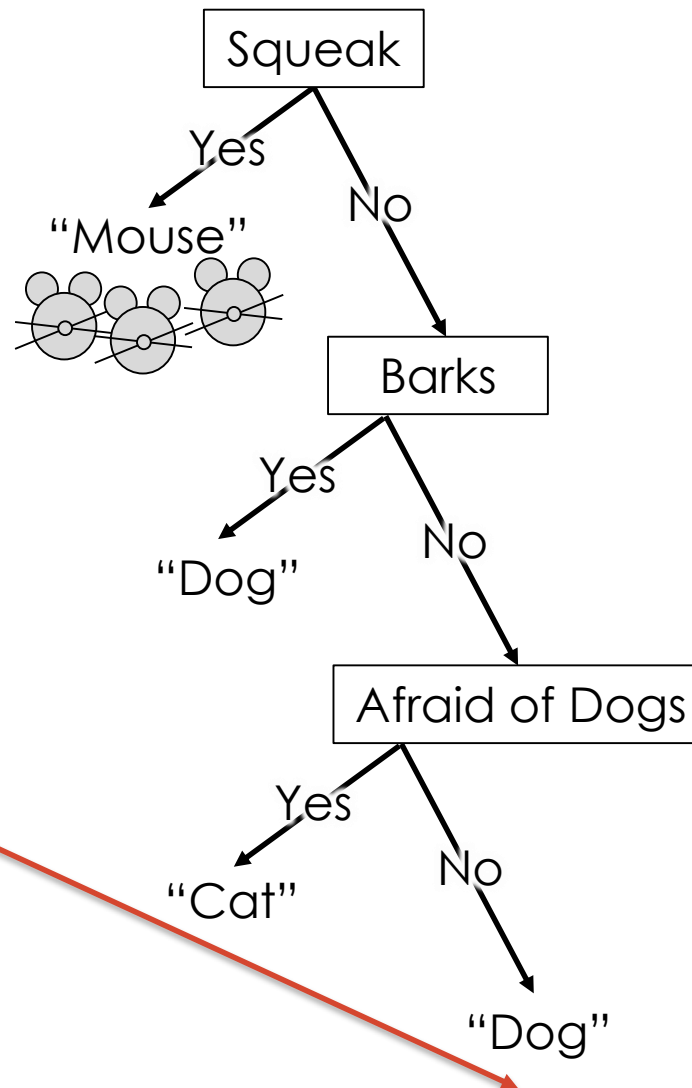
Remaining for internal accuracy assessment  $\approx 1/3$



# Random Forests: Example

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
1	Mouse	No	Yes	Yes	No	No	Yes	0.14	0.07	Yes
2	Mouse	No	No	Yes	No	No	Yes	0.18	0.09	Yes
3	Mouse	No	No	Yes	No	No	Yes	0.13	0.06	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
5	Mouse	No	No	Yes	No	No	Yes	0.13	0.07	Yes
6	Mouse	No	No	Yes	No	No	Yes	0.11	0.05	Yes
7	Mouse	No	No	Yes	No	No	Yes	0.13	0.06	Yes
8	Mouse	No	No	Yes	No	No	Yes	0.16	0.08	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
10	Cat	No	Yes	No	Yes	Yes	Yes	0.31	0.19	Yes
11	Cat	No	Yes	No	Yes	No	Yes	0.38	0.20	Yes
12	Cat	No	Yes	No	Yes	Yes	Yes	0.40	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
15	Cat	No	No	No	Yes	No	No	0.32	0.22	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
17	Cat	No	Yes	No	Yes	Yes	Yes	0.35	0.24	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
19	Dog	Yes	No	No	No	No	No	0.58	0.33	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
23	Dog	No	No	No	No	No	No	0.52	0.26	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
25	Dog	Yes	Yes	No	No	Yes	No	0.37	0.16	Yes
26	Dog	Yes	Yes	No	No	Yes	No	0.53	0.29	Yes
27	Dog	Yes	Yes	No	No	Yes	No	0.58	0.29	Yes

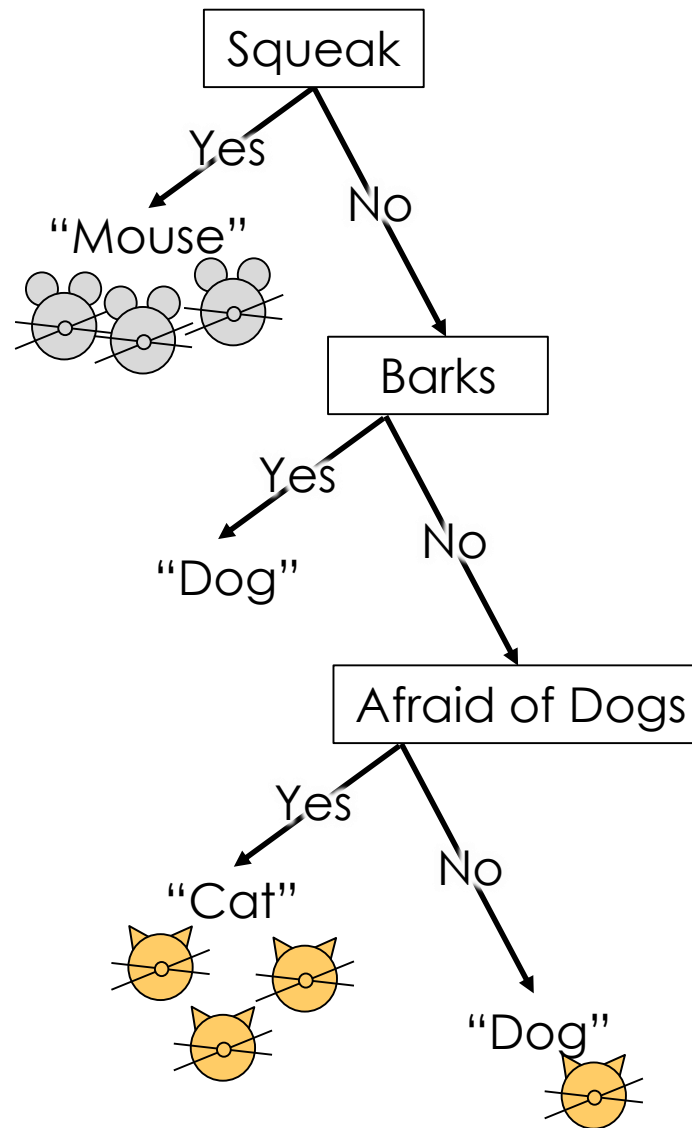
Remaining for internal accuracy assessment ≈ 1/3



# Random Forests: Example

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
1	Mouse	No	Yes	Yes	No	No	Yes	0.14	0.07	Yes
2	Mouse	No	No	Yes	No	No	Yes	0.18	0.09	Yes
3	Mouse	No	No	Yes	No	No	Yes	0.13	0.06	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
5	Mouse	No	No	Yes	No	No	Yes	0.13	0.07	Yes
6	Mouse	No	No	Yes	No	No	Yes	0.11	0.05	Yes
7	Mouse	No	No	Yes	No	No	Yes	0.13	0.06	Yes
8	Mouse	No	No	Yes	No	No	Yes	0.16	0.08	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
10	Cat	No	Yes	No	Yes	Yes	Yes	0.31	0.19	Yes
11	Cat	No	Yes	No	Yes	No	Yes	0.38	0.20	Yes
12	Cat	No	Yes	No	Yes	Yes	Yes	0.40	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
15	Cat	No	No	No	Yes	No	No	0.32	0.22	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
17	Cat	No	Yes	No	Yes	Yes	Yes	0.35	0.24	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
19	Dog	Yes	No	No	No	No	No	0.58	0.33	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
23	Dog	No	No	No	No	No	No	0.52	0.26	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
25	Dog	Yes	Yes	No	No	Yes	No	0.37	0.16	Yes
26	Dog	Yes	Yes	No	No	Yes	No	0.53	0.29	Yes
27	Dog	Yes	Yes	No	No	Yes	No	0.58	0.29	Yes

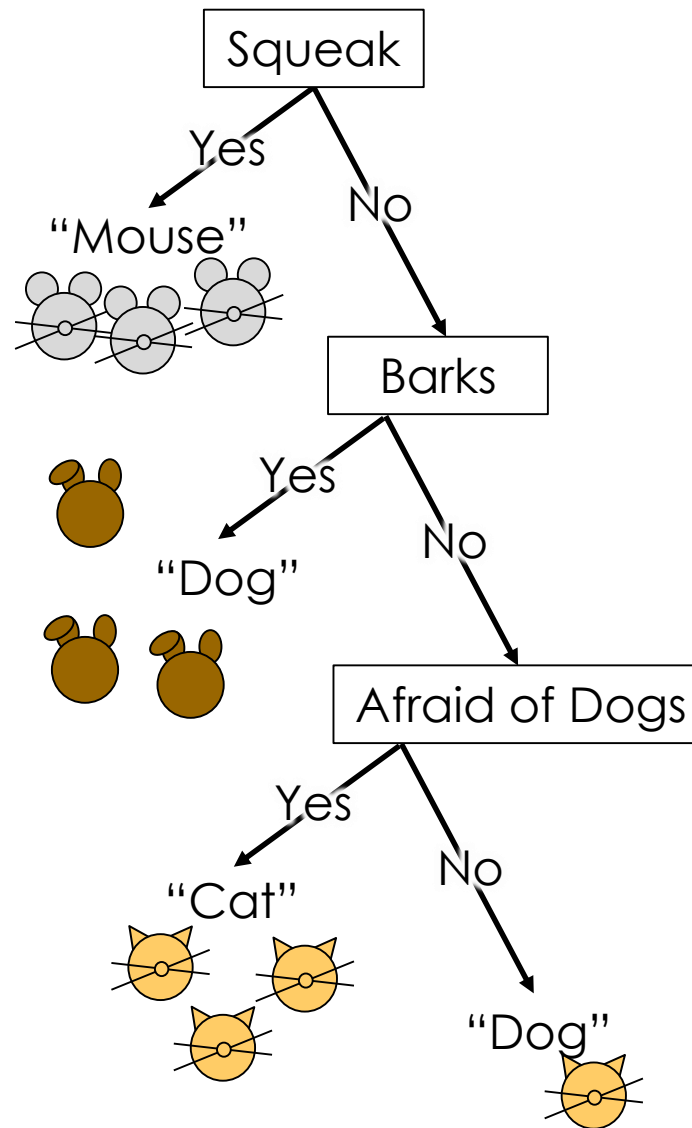
Remaining for internal accuracy assessment ≈ 1/3



# Random Forests: Example

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
1	Mouse	No	Yes	Yes	No	No	Yes	0.14	0.07	Yes
2	Mouse	No	No	Yes	No	No	Yes	0.18	0.09	Yes
3	Mouse	No	No	Yes	No	No	Yes	0.13	0.06	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
5	Mouse	No	No	Yes	No	No	Yes	0.13	0.07	Yes
6	Mouse	No	No	Yes	No	No	Yes	0.11	0.05	Yes
7	Mouse	No	No	Yes	No	No	Yes	0.13	0.06	Yes
8	Mouse	No	No	Yes	No	No	Yes	0.16	0.08	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
10	Cat	No	Yes	No	Yes	Yes	Yes	0.31	0.19	Yes
11	Cat	No	Yes	No	Yes	No	Yes	0.38	0.20	Yes
12	Cat	No	Yes	No	Yes	Yes	Yes	0.40	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
15	Cat	No	No	No	Yes	No	No	0.32	0.22	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
17	Cat	No	Yes	No	Yes	Yes	Yes	0.35	0.24	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
19	Dog	Yes	No	No	No	No	No	0.58	0.33	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
23	Dog	No	No	No	No	No	No	0.52	0.26	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
25	Dog	Yes	Yes	No	No	Yes	No	0.37	0.16	Yes
26	Dog	Yes	Yes	No	No	Yes	No	0.53	0.29	Yes
27	Dog	Yes	Yes	No	No	Yes	No	0.58	0.29	Yes

Remaining for internal accuracy assessment ≈ 1/3

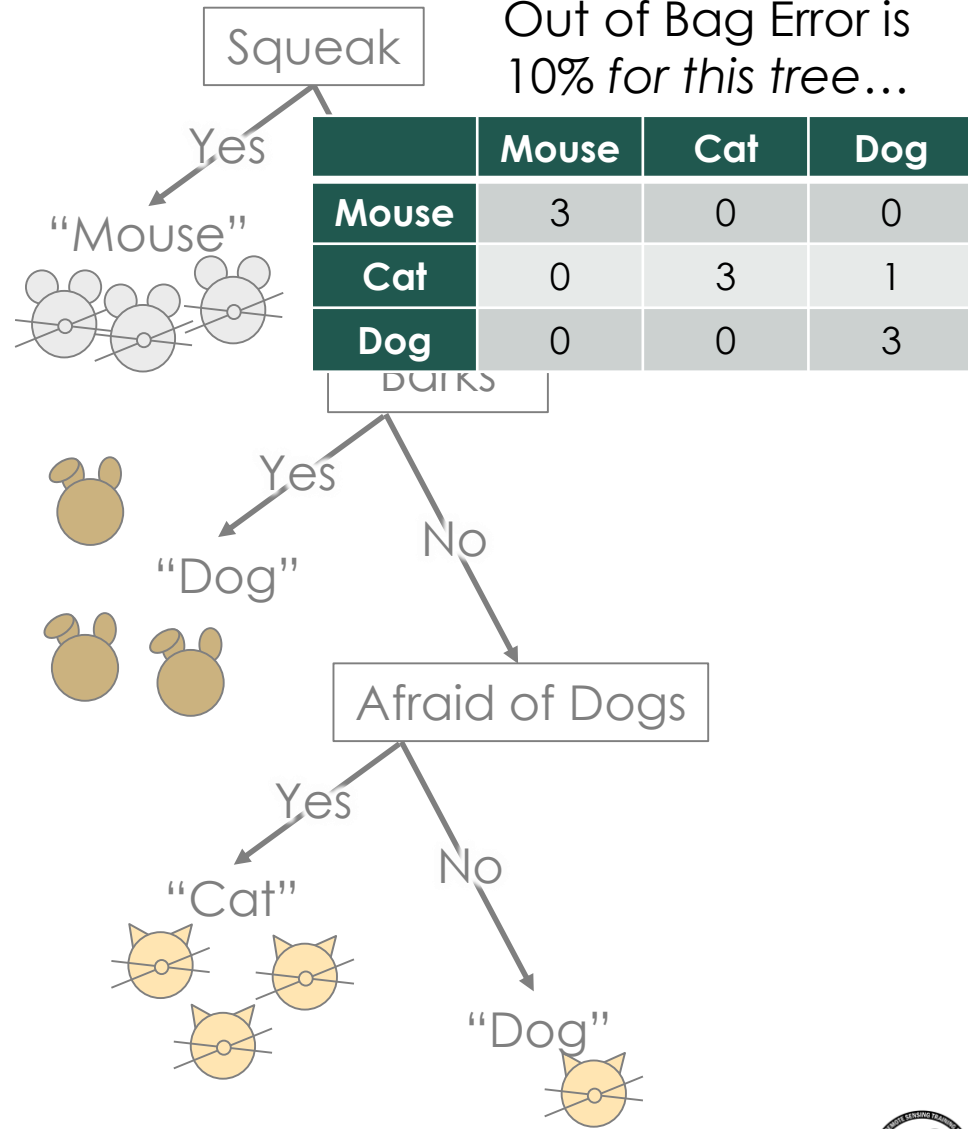


# Random Forests: Example

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
1	Mouse	No	Yes	Yes	No	No	Yes	0.14	0.07	Yes
2	Mouse	No	No	Yes	No	No	Yes	0.18	0.09	Yes
3	Mouse	No	No	Yes	No	No	Yes	0.13	0.06	Yes
4	Mouse	No	Yes	Yes	No	No	Yes	0.13	0.06	Yes
5	Mouse	No	No	Yes	No	No	Yes	0.13	0.07	Yes
6	Mouse	No	No	Yes	No	No	Yes	0.11	0.05	Yes
7	Mouse	No	No	Yes	No	No	Yes	0.13	0.06	Yes
8	Mouse	No	No	Yes	No	No	Yes	0.16	0.08	Yes
9	Mouse	No	No	Yes	No	No	Yes	0.15	0.08	Yes
10	Cat	No	Yes	No	Yes	Yes	Yes	0.31	0.19	Yes
11	Cat	No	Yes	No	Yes	No	Yes	0.38	0.20	Yes
12	Cat	No	Yes	No	Yes	Yes	Yes	0.40	0.15	No
13	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.09	Yes
14	Cat	No	Yes	No	Yes	Yes	Yes	0.36	0.17	Yes
15	Cat	No	No	No	Yes	No	No	0.32	0.22	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.30	0.16	Yes
17	Cat	No	Yes	No	Yes	Yes	Yes	0.35	0.24	Yes
18	Cat	No	Yes	No	Yes	Yes	Yes	0.33	0.22	Yes
19	Dog	Yes	No	No	No	No	No	0.58	0.33	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.53	0.35	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.51	0.33	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.16	0.32	Yes
23	Dog	No	No	No	No	No	No	0.52	0.26	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.53	0.27	Yes
25	Dog	Yes	Yes	No	No	Yes	No	0.37	0.16	Yes
26	Dog	Yes	Yes	No	No	Yes	No	0.53	0.29	Yes
27	Dog	Yes	Yes	No	No	Yes	No	0.58	0.29	Yes

Remaining for internal accuracy assessment  $\approx 1/3$

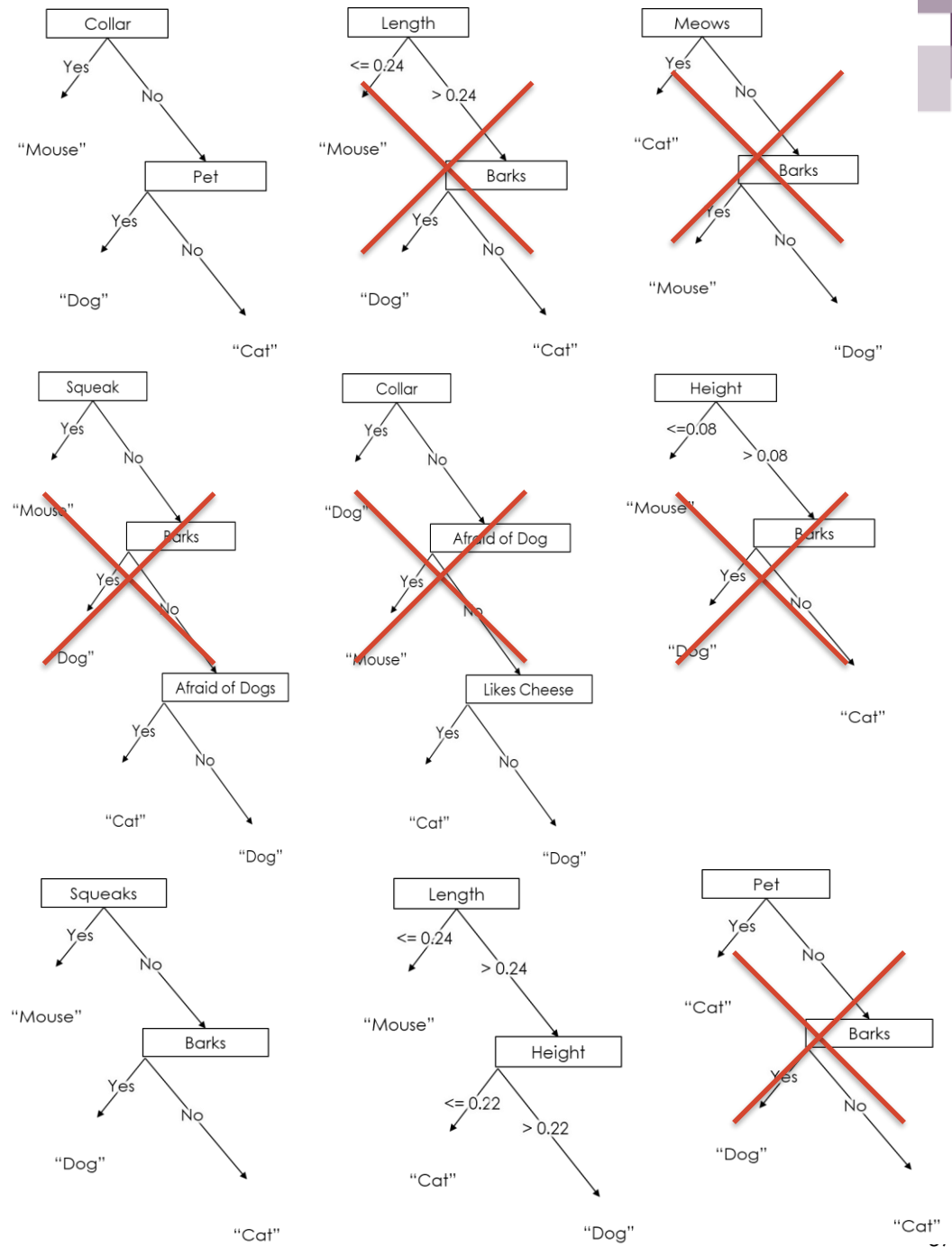
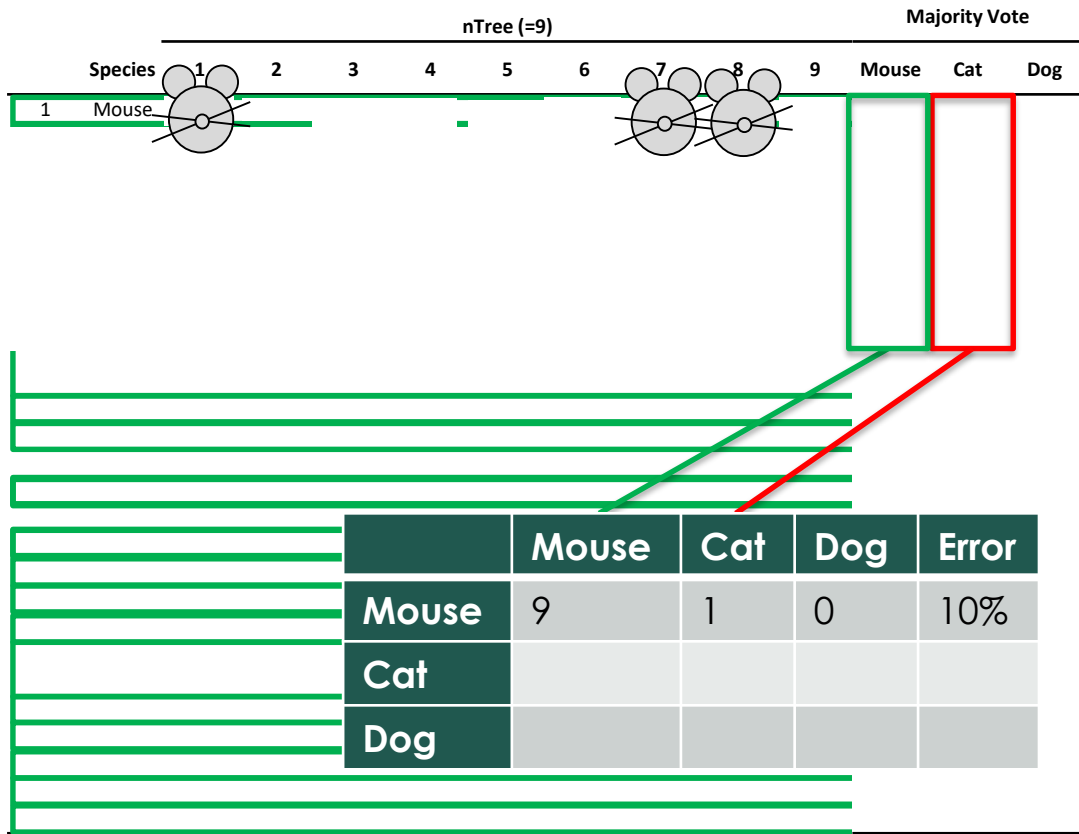
Out of Bag Error is 10% for this tree...





# Random Forests: Example

- Calculate the overall Out of Bag Error



# Random Forests: Example

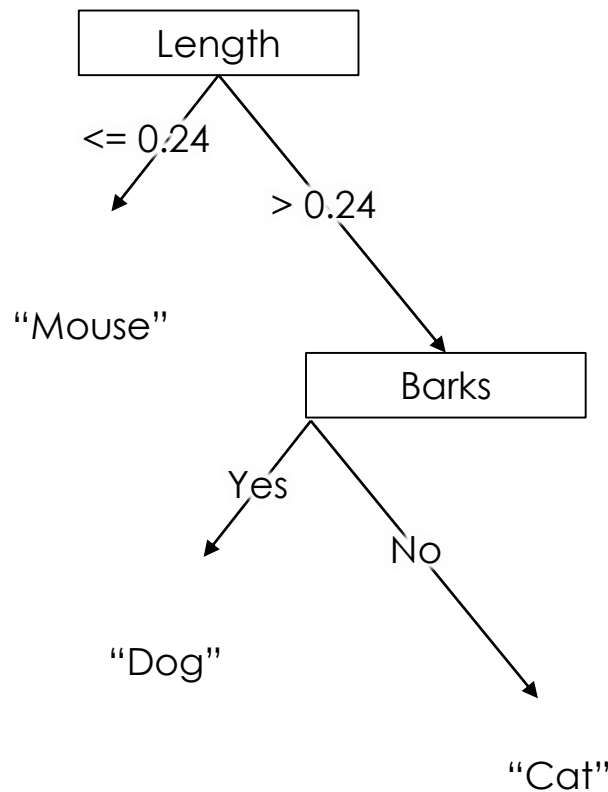
## 5. Calculating the Mean Decrease in Accuracy (MDA)

- Values permuted across all trees
- Average decrease in accuracy; normalize by standard deviation

$$\begin{aligned} \text{Decrease in Accuracy} &= 90\% - 40\% \\ &= 50\% \end{aligned}$$

Original

	Species	Barks	Pet	Squeaks	Meows	Collar	Afraid of Dogs	Length	Height	Likes Cheese
1	Mouse	No	Yes	Yes	No	No	Yes	0.07	0.09	Yes
2	Mouse	No	No	Yes	No	No	Yes	0.31	0.06	Yes
3	Mouse	No	Yes	Yes	No	No	Yes	0.06	0.06	Yes
4	Mouse	No	No	Yes	No	No	Yes	0.07	0.07	Yes
5	Mouse	No	No	Yes	No	No	Yes	0.05	0.05	Yes
6	Mouse	No	No	Yes	No	No	Yes	0.32	0.06	Yes
7	Mouse	No	No	Yes	No	No	Yes	0.38	0.08	Yes
8	Mouse	No	No	Yes	No	No	Yes	0.08	0.08	Yes
9	Cat	No	Yes	No	Yes	Yes	Yes	0.37	0.19	Yes
10	Cat	No	Yes	No	Yes	No	Yes	0.53	0.20	Yes
11	Cat	No	Yes	No	Yes	Yes	Yes	0.15	0.15	No
12	Cat	No	Yes	No	Yes	Yes	Yes	0.09	0.09	Yes
13	Cat	No	Yes	No	Yes	Yes	Yes	0.17	0.17	Yes
14	Cat	No	No	No	Yes	No	No	0.35	0.22	Yes
15	Cat	No	Yes	No	Yes	Yes	Yes	0.16	0.16	Yes
16	Cat	No	Yes	No	Yes	Yes	Yes	0.13	0.24	Yes
17	Cat	No	Yes	No	Yes	Yes	Yes	0.22	0.22	Yes
18	Dog	Yes	No	No	No	No	No	0.58	0.33	Yes
19	Dog	Yes	Yes	No	No	Yes	No	0.35	0.35	Yes
20	Dog	Yes	Yes	No	No	Yes	No	0.33	0.33	Yes
21	Dog	Yes	Yes	No	No	Yes	No	0.33	0.33	Yes
22	Dog	Yes	Yes	No	No	Yes	No	0.32	0.32	Yes
23	Dog	No	No	No	No	No	No	0.26	0.26	Yes
24	Dog	Yes	Yes	No	No	Yes	No	0.27	0.27	Yes
25	Dog	Yes	Yes	No	No	Yes	No	0.13	0.16	Yes
26	Dog	Yes	Yes	No	No	Yes	No	0.16	0.29	Yes
27	Dog	Yes	Yes	No	No	Yes	No	0.29	0.29	Yes



Original Out of Bag Accuracy 90%

	Mouse	Cat	Dog
Mouse	3	0	0
Cat	0	3	1
Dog	0	0	3

Out of Bag Accuracy After Permutation 40%

	Mouse	Cat	Dog
Mouse	0	3	0
Cat	1	3	0
Dog	2	0	1

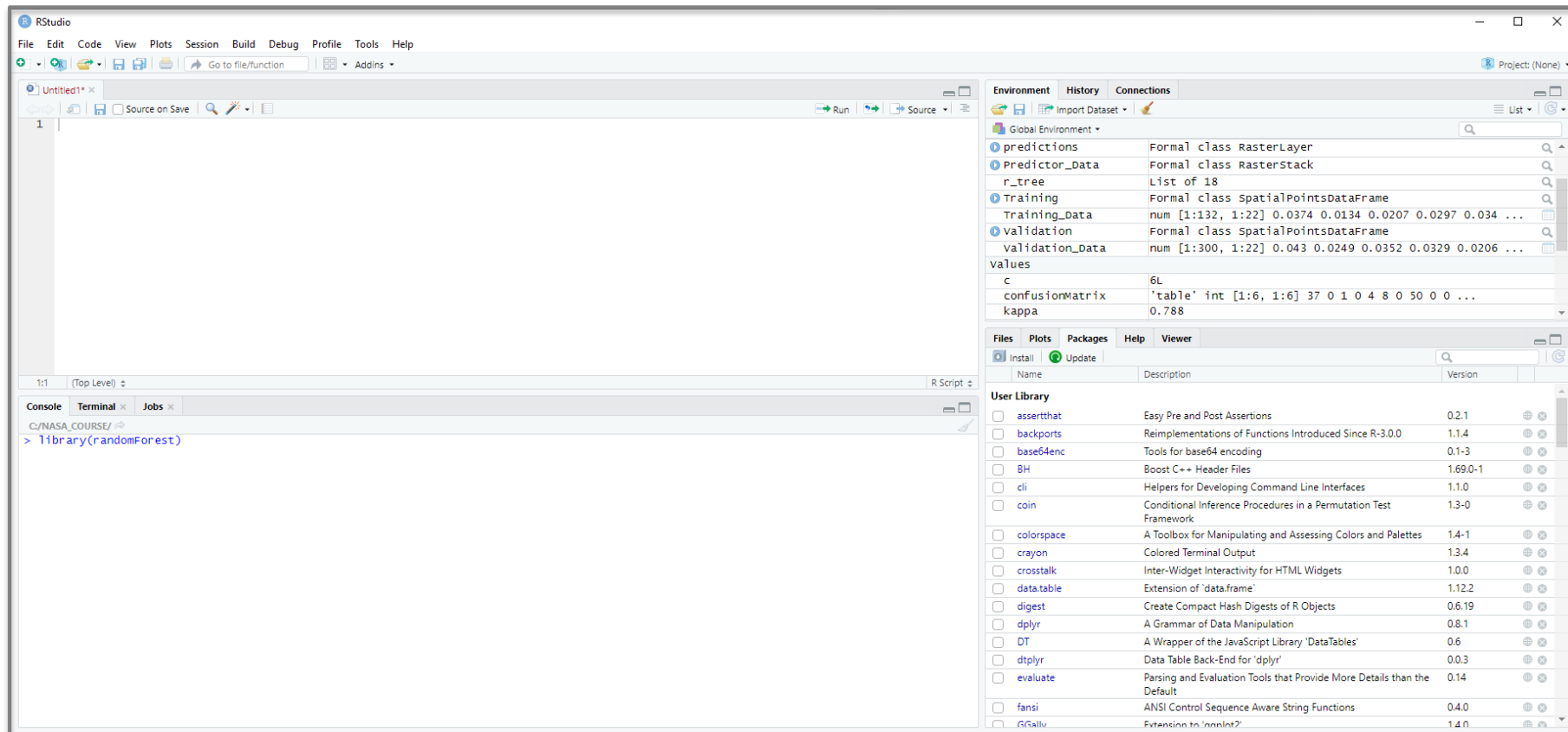
Original Length Data  
Perth Data  
Length Data

# Random Forests: Hands on Demonstration of Crop Classification

1. Install the latest version of R and RStudio
  - <https://cran.r-project.org/bin/windows/base/>
  - <https://www.rstudio.com/products/rstudio/download/>
2. Open Rstudio

# Random Forests: Hands on Demonstration

3. Install (raster, randomForest, sp, rgdal)
4. Load (raster, randomForest, sp, rgdal)



The screenshot shows the RStudio interface. The main editor window is empty. The console window at the bottom shows the command `library(randomForest)` being executed. The Environment pane on the right shows the following objects:

Object	Class
predictions	Formal class RasterLayer
Predictor_Data	Formal class RasterStack
r_tree	List of 18
Training	Formal class SpatialPointsDataFrame
Training_Data	num [1:132, 1:22] 0.0374 0.0134 0.0207 0.0297 0.034 ...
Validation	Formal class SpatialPointsDataFrame
Validation_Data	num [1:300, 1:22] 0.043 0.0249 0.0352 0.0329 0.0206 ...
Values	
c	6L
confusionMatrix	'table' int [1:6, 1:6] 37 0 1 0 4 8 0 50 0 0 ...
kappa	0.788

The Packages pane on the right shows the User Library with the following packages installed:

Package	Description	Version
assertthat	Easy Pre and Post Assertions	0.2.1
backports	Reimplementations of Functions Introduced Since R-3.0.0	1.1.4
base64enc	Tools for base64 encoding	0.1-3
BH	Boost C++ Header Files	1.69.0-1
cli	Helpers for Developing Command Line Interfaces	1.1.0
coin	Conditional Inference Procedures in a Permutation Test Framework	1.3-0
colorspace	A Toolbox for Manipulating and Assessing Colors and Palettes	1.4-1
crayon	Colored Terminal Output	1.3.4
crosstalk	Inter-Widget Interactivity for HTML Widgets	1.0.0
data.table	Extension of 'data.frame'	1.12.2
digest	Create Compact Hash Digests of R Objects	0.6.19
dplyr	A Grammar of Data Manipulation	0.8.1
DT	A Wrapper of the JavaScript Library 'DataTables'	0.6
dtplyr	Data Table Back-End for 'dplyr'	0.0.3
evaluate	Parsing and Evaluation Tools that Provide More Details than the Default	0.14
fansi	ANSI Control Sequence Aware String Functions	0.4.0
GGally	Extension to 'ggplot2'	1.4.0

# Random Forests: Hands on Demonstration

## 5. Set the Working Directory

The screenshot displays the RStudio interface with the following R code in the editor:

```
1 #1) Set the working directory
2 setwd('C:/NASA_COURSE')
3
4
5 #2) Create a raster object
6 inraster <- raster::stack('0_RASTER/RS2_TSX_Sentinel1_1_30m_UTM.tif')
7
8 #3) Set the path to the training data; a csv file containing class labels (class)
9 #and Easting (POINT_X) and Northing (POINT_Y) information
10 Training <- read.csv('1_TRAIN_VAL_FINAL_FINAL/TRAINING.csv', header=TRUE, sep = ",")
11 Validation <- read.csv('1_TRAIN_VAL_FINAL_FINAL/VALIDATION.csv', header=TRUE, sep = ",")
12
13
14 #4) Identify which columns contain coordinate information
15 coordinates(Training) <- ~Point_X+Point_Y
16 coordinates(Validation) <- ~Point_X+Point_Y
17
18
19 #5) Set the projection of the point data
20 proj4string(Training) <- CRS("+proj=utm +zone=14 +datum=WGS84 +units=m +no_defs +ellps=WGS84 +towgs84=0,0,0")
21 proj4string(Validation) <- CRS("+proj=utm +zone=14 +datum=WGS84 +units=m +no_defs +ellps=WGS84 +towgs84=0,0,0")
22
23
24 #6) Extract training data from the raster
25 Training_Data <- raster::extract(inraster, Training)
```

The console output shows the execution of the first two lines of code:

```
C:/NASA_COURSE/ #>
> #1) Set the working directory
> setwd('C:/NASA_COURSE')
> |
```

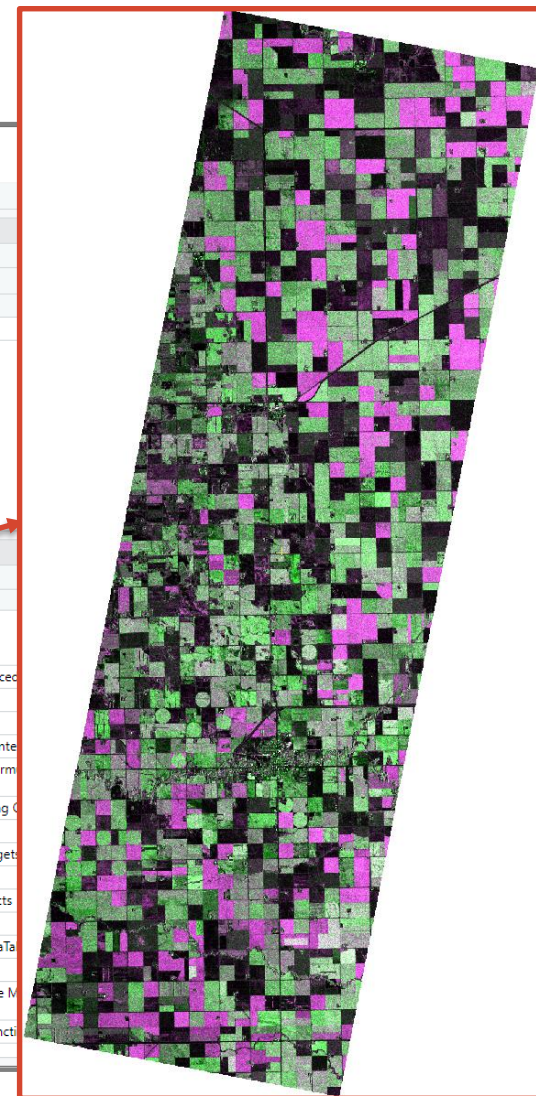
The Environment pane on the right shows the 'Data' section with 'inraster' listed as a 'Formal class RasterStack'. The Packages pane shows a list of installed user libraries.

# Random Forests: Hands on Demonstration

## 6. Create a raster object

```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
+ Run + Source
1 #1) Set the working directory
2 setwd('C:/NASA_COURSE')
3
4
5 #2) Create a raster object
6 inraster <- raster::stack('0_RASTER/RS2_TSX_sentinel1_1_30m_UTM.tif')
7
8 #3) Set the path to the training data; a csv file containing class labels (Class)
9 #and Easting (POINT_X) and Northing (POINT_Y) information
10 Training <- read.csv('1_TRAIN_VAL_FINAL_FINAL/TRAINING.csv', header=TRUE, sep = ",")
11 Validation <- read.csv('1_TRAIN_VAL_FINAL_FINAL/VALIDATION.csv', header=TRUE, sep = ",")
12
13
14 #4) Identify which columns contain coordinate information
15 coordinates(Training) <- ~Point_X+Point_Y
16 coordinates(Validation) <- ~Point_X+Point_Y
17
18
19 #5) Set the projection of the point data
20 proj4string(Training) <- CRS("+proj=utm +zone=14 +datum=WGS84 +units=m +no_defs +ellps=WGS84 +towgs84=0,0,0")
21 proj4string(Validation) <- CRS("+proj=utm +zone=14 +datum=WGS84 +units=m +no_defs +ellps=WGS84 +towgs84=0,0,0")
22
23
24 #6) Extract training data from the raster
25 Training_Data <- raster::extract(inraster, Training)
--
&1 (Top Level)
R Script

Console Terminal Jobs
C:/NASA_COURSE/
> #1) Set the working directory
> setwd('C:/NASA_COURSE')
> #2) Create a raster object
> inraster <- raster::stack('0_RASTER/RS2_TSX_sentinel1_1_30m_UTM.tif')
> |
```



# Random Forests: Hands on Demonstration

## 7. Read training and validation data (csv files)

The screenshot shows the RStudio interface with the following R code in the editor:

```
1 #1) Set the working directory
2 setwd('C:/NASA_COURSE')
3
4
5 #2) Create a raster object
6 inraster <- raster::stack('0_RASTER/RS2_TSX_Sentinel1_1_30m_UTM.tif')
7
8 #3) Set the path to the training data; a csv file containing class labels (class)
9 #and Easting (POINT_X) and Northing (POINT_Y) information
10 Training <- read.csv('1_TRAIN_VAL_FINAL_FINAL/TRAINING.csv', header=TRUE, sep = ",")
11 Validation <- read.csv('1_TRAIN_VAL_FINAL_FINAL/VALIDATION.csv', header=TRUE, sep = ",") |
12
13
14 #4) Identify which columns contain coordinate information
15 coordinates(Training) <- ~Point_X+Point_Y
16 coordinates(Validation) <- ~Point_X+Point_Y
17
18
19 #5) Set the projection of the point data
20 proj4string(Training) <- CRS("+proj=utm +zone=14 +datum=WGS84 +units=m +no_defs +ellps=WGS84 +towgs84=0,0,0")
21 proj4string(Validation) <- CRS("+proj=utm +zone=14 +datum=WGS84 +units=m +no_defs +ellps=WGS84 +towgs84=0,0,0")
22
23
24 #6) Extract training data from the raster
25 Training_Data <- raster::extract(inraster, Training)
```

The Environment pane on the right shows the following data objects:

Object	Class	Details
inraster	Formal class RasterStack	
Training		132 obs. of 3 variables
Validation		300 obs. of 3 variables

The Notepad window titled "TRAINING - Notepad" displays the following data:

```
Class,Point_X,Point_Y
Barley,571228.626047000000,5516307.572170000000
Barley,573654.537612000000,5519554.465660000000
Barley,573534.361710000000,5519458.885590000000
Barley,573512.377323000000,5519589.390450000000
Barley,579210.637366000000,5521301.384130000000
Barley,579303.195679000000,5521339.726470000000
Barley,579272.999664000000,5521249.234720000000
Barley,580100.042556000000,5521221.617050000000
Barley,580094.146626000000,5521159.343240000000
Barley,580024.444619000000,5521284.841660000000
Barley,581088.244124000000,5515604.722860000000
Barley,574437.943870000000,5519484.989090000000
Barley,574439.603131000000,5519629.232600000000
Barley,579207.927005000000,5522104.981850000000
Barley,580921.478439000000,5521307.702860000000
Barley,580873.839083000000,5519735.121680000000
Barley,578554.091948000000,5521235.228940000000
Barley,578474.805706000000,5520349.535140000000
```

# Random Forests: Hands on Demonstration

## 8. Identify which columns from the csv files contain coordinate information

The screenshot displays the RStudio interface. The main editor window contains R code for setting up a raster and reading training data. The console shows the execution of these commands. A Notepad window titled 'TRAINING - Notepad' is open, showing the output of the `coordinates()` function applied to the training data. The output lists the class label 'Barley' followed by the X and Y coordinates for each training point.

```
#1) Set the working directory
2 setwd('c:/NASA_COURSE')
3
4
5 #2) Create a raster object
6 inraster <- raster::stack('0_RASTER/RS2_TSX_Sentinel1_1_30m_UTM.tif')
7
8 #3) Set the path to the training data; a csv file containing class labels (class)
9 #and Easting (POINT_X) and Northing (POINT_Y) information
10 Training <- read.csv('1_TRAIN_VAL_FINAL_FINAL/TRAINING.csv', header=TRUE, sep = ",")
11 Validation <- read.csv('1_TRAIN_VAL_FINAL_FINAL/VALIDATION.csv', header=TRUE, sep = ",")
12
13
14 #4) Identify which columns contain coordinate information
15 coordinates(Training)<- ~Point_X+Point_Y
16 coordinates(Validation)<- ~Point_X+Point_Y
17
18 |
19 #5) Set the projection of the point data
20 proj4string(Training)<- CRS("+proj=utm +zone=14 +datum=WGS84 +units=m +no_defs +ellps=WGS84 +towgs84=0,0,0,0,0,0,0")
21 proj4string(Validation)<- CRS("+proj=utm +zone=14 +datum=WGS84 +units=m +no_defs +ellps=WGS84 +towgs84=0,0,0,0,0,0,0")
22
23
24 #6) Extract training data from the raster
25 Training_Data <- raster::extract(inraster, Training)
```

```
TRAINING - Notepad
File Edit Format View Help
class,Point_X,Point_Y
Barley,571228.626047000000,5516307.572170000000
Barley,573654.537612000000,5519554.465660000000
Barley,573534.361710000000,5519458.885590000000
Barley,573512.377323000000,5519589.390450000000
Barley,579210.637366000000,5521301.384130000000
Barley,579303.195679000000,5521339.726470000000
Barley,579272.999664000000,5521249.234720000000
Barley,580100.042556000000,5521221.617050000000
Barley,580094.146626000000,5521159.343240000000
Barley,580024.444619000000,5521284.841660000000
Barley,581088.244124000000,5515604.722860000000
Barley,574437.943870000000,5519484.989090000000
Barley,574439.603131000000,5519629.232600000000
Barley,579207.927005000000,5522104.981850000000
Barley,580921.478439000000,5521307.702860000000
Barley,580873.839083000000,5519735.121680000000
Barley,578554.091948000000,5521235.228940000000
Barley,578474.805706000000,5520349.535140000000
```



# Random Forests: Hands on Demonstration

## 9. Define the projection of the point data in the csv files

The screenshot displays the RStudio interface with the following R code in the editor:

```
11 validation <- read.csv('1_TRAIN_VAL_FINAL_FINAL/VALIDATION.csv', header=TRUE, sep = ",")
12
13
14 #4) Identify which columns contain coordinate information
15 coordinates(Training)<- ~Point_X+Point_Y
16 coordinates(validation)<- ~Point_X+Point_Y
17
18
19 #5) Set the projection of the point data
20 proj4string(Training)<- CRS("+proj=utm +zone=14 +datum=WGS84 +units=m +no_defs +ellps=WGS84 +towgs84=0,0,0")
21 proj4string(validation)<- CRS("+proj=utm +zone=14 +datum=WGS84 +units=m +no_defs +ellps=WGS84 +towgs84=0,0,0")
22
23
24 #6) Extract training data from the raster
25 Training_Data <- raster::extract(inraster, Training)
26 Training_Response <- as.factor(Training$class)
27
28
29 #7) Select which variables from the raster stack to use in your model
30 selection <- c(1:22)
31 Predictor_Data <- Training_Data[,selection]
32
33
34 #8) Create and save the forest
35 r_tree <- randomForest(Predictor_Data, y=Training_Response, ntree = 1000, keep.forest=TRUE, importance = TRUE, na.action=na.omit)
```

The Environment pane on the right shows the following data objects:

Object	Class
inraster	Formal class RasterStack
Training	Formal class SpatialPointsDataFrame
validation	Formal class SpatialPointsDataFrame

The Console pane shows the execution of the code, including the following output:

```
> #1) Set the working directory
> setwd('C:/NASA_COURSE')
> #2) Create a raster object
> inraster <- raster::stack('0_RASTER/RS2_TSX_Sentinel1_1_30m_UTM.tif')
> #3) Set the path to the training data; a csv file containing class labels (class)
> #and Easting (POINT_X) and Northing (POINT_Y) information
> training <- read.csv('1_TRAIN_VAL_FINAL_FINAL/TRAINING.csv', header=TRUE, sep = ",")
> validation <- read.csv('1_TRAIN_VAL_FINAL_FINAL/VALIDATION.csv', header=TRUE, sep = ",")
> #4) Identify which columns contain coordinate information
> coordinates(Training)<- ~Point_X+Point_Y
> coordinates(validation)<- ~Point_X+Point_Y
> #5) Set the projection of the point data
> proj4string(Training)<- CRS("+proj=utm +zone=14 +datum=WGS84 +units=m +no_defs +ellps=WGS84 +towgs84=0,0,0")
> proj4string(validation)<- CRS("+proj=utm +zone=14 +datum=WGS84 +units=m +no_defs +ellps=WGS84 +towgs84=0,0,0")
>
```

# Random Forests: Hands on Demonstration

10. Extract the training data (values of each raster band coincident with point data)

The screenshot displays the RStudio interface. The script editor on the left contains R code for training a random forest model. The console at the bottom shows the execution of these commands, including setting the working directory, creating a raster object, reading training and validation data, and printing the variable importance.

```
23
24 #6) Extract training data from the raster
25 Training_Data <- raster::extract(inraster, Training)
26 Training_Response <- as.factor(Training$class)
27
28
29 #7) Select which variables from the raster stack to use in your model
30 Selection <- c(1:22)
31 Predictor_Data <- Training_Data[,Selection]
32
33
34 #8) Create and save the forest
35 r_tree <- randomForest(Predictor_Data, y=Training_Response, ntree = 1000, keep.forest=TRUE, importance = TRUE, na.action=na.omit)
36
37
38 #9) See the Out of Bag Confusion Matrix
39 r_tree
40
41
42 #10) Print the variable importance (Mean Decrease in Accuracy; for Gini Index type = 2)
43 imp <- importance(r_tree, type = 1)
44 imp
45
46
47 #11) Extract values to be used for independent validation
48
2648 (Top Level) R Script
```

Console Output:

```
C:/NASA_COURSE/
> setwd('C:/NASA_COURSE')
> #2) Create a raster object
> inraster <- raster::stack('0_RASTER/RS2_TSX_Sentinel1_1_30m_UTM.tif')
> #3) Set the path to the training data; a csv file containing class labels (class)
> #and Easting (POINT_X) and Northing (POINT_Y) information
> Training <- read.csv('1_TRAIN_VAL_FINAL_FINAL/TRAINING.csv', header=TRUE, sep = ",")
> Validation <- read.csv('1_TRAIN_VAL_FINAL_FINAL/VALIDATION.csv', header=TRUE, sep = ",")
> #4) Identify which columns contain coordinate information
> coordinates(Training) <- ~Point_X+Point_Y
> coordinates(Validation) <- ~Point_X+Point_Y
> #5) Set the projection of the point data
> proj4string(Training) <- CRS("+proj=utm +zone=14 +datum=WGS84 +units=m +no_defs +ellps=WGS84 +towgs84=0,0,0")
> proj4string(Validation) <- CRS("+proj=utm +zone=14 +datum=WGS84 +units=m +no_defs +ellps=WGS84 +towgs84=0,0,0")
> #6) Extract training data from the raster
> Training_Data <- raster::extract(inraster, Training)
> Training_Response <- as.factor(Training$class)
>
```

# Random Forests: Hands on Demonstration

## 11. Select which variables to use in the model

The image displays a hands-on demonstration of random forest variable selection in R. It features three main components:

- RStudio Code Editor:** Shows R code for extracting training data, selecting variables, and training a random forest model. A red arrow points from the code to the variable list.
- Variable List:** A vertical list of 20 variable names, each representing a different satellite sensor and date.
- RStudio Environment Panel:** Shows the current environment with variables like 'inraster', 'Training', and 'Validation'.
- Satellite Image:** A satellite image of a field with a red bounding box, likely representing the training area.

VH_TSX_20160726_mst_26Jul2016
VV_TSX_20160726_mst_26Jul2016
VH_TSX_20160726_slv1_26Jul2016
VV_TSX_20160726_slv2_26Jul2016
HH_RS2_20160703_slv3_01Jan2000
HV_RS2_20160703_slv4_01Jan2000
VH_RS2_20160703_slv5_01Jan2000
VV_RS2_20160703_slv6_01Jan2000
HH_RS2_20160727_slv7_01Jan2000
HV_RS2_20160727_slv8_01Jan2000
VH_RS2_20160727_slv9_01Jan2000
VV_RS2_20160727_slv10_01Jan2000
HH_RS2_20160820_slv11_01Jan2000
HV_RS2_20160820_slv12_01Jan2000
VH_RS2_20160820_slv13_01Jan2000
VV_RS2_20160820_slv14_01Jan2000
VH_S1A_20160613_slv15_13Jul2016
VV_S1A_20160613_slv16_13Jul2016
VH_S1A_20160707_slv17_07Jul2016
VV_S1A_20160707_slv18_07Jul2016
VH_S1A_20160731_slv19_31Jul2016
VV_S1A_20160731_slv20_31Jul2016

# Random Forests: Hands on Demonstration

## 12. Create your Random Forest Model

The screenshot shows the RStudio interface with the following R code in the editor:

```
## #7) Select which variables from the raster stack to use in your model
Selection <- c(1:22)
Predictor_Data <- Training_Data[,Selection]
## #8) Create and save the forest
r_tree <- randomForest(Predictor_Data, y=Training_Response, ntree = 1000, keep.forest=TRUE, importance = TRUE, na.action=na.omit)
## #9) See the Out of Bag Confusion Matrix
r_tree
## #10) Print the variable importance (Mean Decrease in Accuracy; for Gini Index type = 2)
imp <- importance(r_tree, type = 1)
imp
## #11) Extract values to be used for independent validation
validation_Data <- raster::extract(inraster, validation)[,Selection]
validation_Response <- as.factor(validation$Class)
## #12) Classify the independent validation data
validation_Predictions <- predict(r_tree, validation_Data)
```

The console shows the execution of the code, with the following output for the random forest model:

```
> #7) Select which variables from the raster stack to use in your model
> Selection <- c(1:22)
> Predictor_Data <- Training_Data[,Selection]
> #8) Create and save the forest
> r_tree <- randomForest(Predictor_Data, y=Training_Response, ntree = 1000, keep.forest=TRUE, importance = TRUE, na.action=na.omit)
> |
```

The table below shows the results of the random forest model, with the 'Class' column indicating the predicted class for each validation point. The table is highlighted with a red border in the original image.

Class	VH_TSX_20160726_mst_26Jul2016	VV_TSX_20160726_mst_26Jul2016	VH_TSX_20160726_slv1_26Jul2016
Barley	0.037364677	0.133464038	0.037364677
Barley	0.013369569	0.067057364	0.013369569
Barley	0.020673014	0.074069843	0.020673014
Barley	0.029688779	0.08695662	0.029688779
Barley	0.033956379	0.109707654	0.033956379
Barley	0.0146865	0.052232202	0.0146865
Barley	0.018100204	0.06887947	0.018100204
Barley	0.018511023	0.051729776	0.018511023
Barley	0.01569712	0.045542698	0.01569712
Barley	0.015248202	0.05094168	0.015248202
Canola	0.039782844	0.145142376	0.039782844
Canola	0.040325992	0.161023989	0.040325992
Canola	0.055418897	0.135640427	0.055418897
Canola	0.066946477	0.154822439	0.066946477
Canola	0.094055369	0.140497714	0.094055369
Canola	0.084282458	0.166150674	0.084282458
Canola	0.09123531	0.211286813	0.09123531

# Random Forests: Hands on Demonstration

## 13. Print the Out of Bag confusion matrix

The screenshot shows the RStudio interface with the following R code in the script editor:

```
33
34 #8) Create and save the forest]
35 r_tree <- randomForest(Predictor_Data, y=Training_Response, ntree = 1000, keep.forest=TRUE, importance = TRUE, na.action=na.omit)
36
37
38 #9) See the out of Bag Confusion Matrix
39 r_tree
40
41
42 #10) Print the variable importance (Mean Decrease in Accuracy; for Gini Index type = 2)
43 imp <- importance(r_tree, type = 1)
44 imp
45
46
47 #11) Extract values to be used for independent validation
48 Validation_Data <- raster::extract(inraster, validation)[,Selection]
49 Validation_Response <- as.factor(Validation$class)
50
51
52 #12) Classify the independent validation data
53 Validation_Predictions <- predict(r_tree, Validation_Data)
54
55
56 #13) Generate a confusion matrix from the independent validation data
```

The console output shows the following information:

```
> #9) See the Out of Bag Confusion Matrix
> r_tree

Call:
randomForest(x = Predictor_Data, y = Training_Response, ntree = 1000, importance = TRUE, keep.forest = TRUE, na.action = na.omit)
Type of random forest: classification
Number of trees: 1000
No. of variables tried at each split: 4

OOB estimate of error rate: 17.42%
Confusion matrix:
      Barley  canola  corn  oats  soybeans  Spring wheat  class.error
Barley      17      0      0      0          3          2  0.22727273
canola       0     21      0      1          0          0  0.04545455
corn         0      0     20      0          2          0  0.09090909
oats         0      0      0     19          0          3  0.13636364
soybeans     1      0      5      0         16          0  0.27272727
Spring wheat 1      0      0      3          2         16  0.27272727
>
```

The Environment pane on the right shows the following objects:

Object	Class	Value
Global Environment		
training_data	num	[1:152, 1:22] 0.0374 0.0154 0.0207 0.0297 0.034 ...
Validation	Formal class SpatialPointsDataFrame	
validation_data	num	[1:300, 1:22] 0.043 0.0249 0.0352 0.0329 0.0206 ...
values		
c	6L	
confusionMatrix	'table' int	[1:6, 1:6] 37 0 1 0 4 8 0 49 0 1 ...
kappa	0.788	
n_classes	6L	
n_obs	300L	
outputRaster	"3_RESULTS/RS2_TSX_Sentinel1_1_30m_UTM_RandomForest.tif"	
overallAccuracy	0.823333333333333	
rowColSumProdSum	15000L	

The Files pane shows the User Library with the following packages:

Name	Description	Version
assertthat	Easy Pre and Post Assertions	0.2.1
backports	Reimplementations of Functions Introduced Since R-3.0.0	1.1.4
base64enc	Tools for base64 encoding	0.1-3
BH	Boost C++ Header Files	1.69.0-1
cli	Helpers for Developing Command Line Interfaces	1.1.0
coin	Conditional Inference Procedures in a Permutation Test Framework	1.3-0
colorspace	A Toolbox for Manipulating and Assessing Colors and Palettes	1.4-1
crayon	Colored Terminal Output	1.3.4
crosstalk	Inter-Widget Interactivity for HTML Widgets	1.0.0
data.table	Extension of 'data.frame'	1.12.2
digest	Create Compact Hash Digests of R Objects	0.6.19
dplyr	A Grammar of Data Manipulation	0.8.1
DT	A Wrapper of the JavaScript Library 'DataTables'	0.6
dtplyr	Data Table Back-End for 'dplyr'	0.0.3
evaluate	Parsing and Evaluation Tools that Provide More Details than the Default	0.14
fansi	ANSI Control Sequence Aware String Functions	0.4.0
GGally	Extension to 'ggplot2'	1.4.0

# Random Forests: Hands on Demonstration

## 14. Print the variable importance values

The screenshot shows the RStudio interface with the following code in the script editor:

```
33
34 #8) Create and save the forest
35 r_tree <- randomForest(Predictor_Data, y=Training_Response, ntree = 1000, keep.forest=TRUE, importance = TRUE, na.action=na.omit)
36
37
38 #9) See the Out of Bag Confusion Matrix
39 r_tree
40
41
42 #10) Print the variable importance (Mean Decrease in Accuracy; for Gini Index type = 2)
43 imp <- importance(r_tree, type = 1)
44 imp
45
46
47 #11) Extract values to be used for independent validation
```

The console output shows the variable importance values:

```
> #10) Print the variable importance (Mean Decrease in Accuracy; for Gini Index type = 2)
> imp <- importance(r_tree, type = 1)
> imp
              MeanDecreaseAccuracy
RS2_TSX_Sentinel_1_30m_UTM.1      18.706255
RS2_TSX_Sentinel_1_30m_UTM.2      22.983380
RS2_TSX_Sentinel_1_30m_UTM.3      19.562098
RS2_TSX_Sentinel_1_30m_UTM.4      23.879731
RS2_TSX_Sentinel_1_30m_UTM.5      10.365050
RS2_TSX_Sentinel_1_30m_UTM.6      16.029312
RS2_TSX_Sentinel_1_30m_UTM.7      18.290303
RS2_TSX_Sentinel_1_30m_UTM.8      15.955196
RS2_TSX_Sentinel_1_30m_UTM.9      17.161803
RS2_TSX_Sentinel_1_30m_UTM.10     23.430637
RS2_TSX_Sentinel_1_30m_UTM.11     23.996702
RS2_TSX_Sentinel_1_30m_UTM.12     18.072785
RS2_TSX_Sentinel_1_30m_UTM.13      4.783384
RS2_TSX_Sentinel_1_30m_UTM.14      4.348962
RS2_TSX_Sentinel_1_30m_UTM.15      5.640269
RS2_TSX_Sentinel_1_30m_UTM.16     17.624666
RS2_TSX_Sentinel_1_30m_UTM.17      8.594696
RS2_TSX_Sentinel_1_30m_UTM.18      8.847914
RS2_TSX_Sentinel_1_30m_UTM.19     11.249187
RS2_TSX_Sentinel_1_30m_UTM.20     11.835973
RS2_TSX_Sentinel_1_30m_UTM.21     20.294629
RS2_TSX_Sentinel_1_30m_UTM.22     19.942409
```

# Random Forests: Hands on Demonstration

## 15. Extract the validation data

The screenshot shows the RStudio interface with the following R code in the editor:

```
45  
46  
47 #11) Extract values to be used for independent validation  
48 validation_data <- raster::extract(inraster, validation)[,selection]  
49 validation_response <- as.factor(validation$class) |  
50  
51  
52 #12) Classify the independent validation data  
53 validation_predictions <- predict(r_tree, validation_data)  
54  
55  
56 #13) Generate a confusion matrix from the independent validation data  
49:52 (Top Level)
```

The console shows the execution of the code:

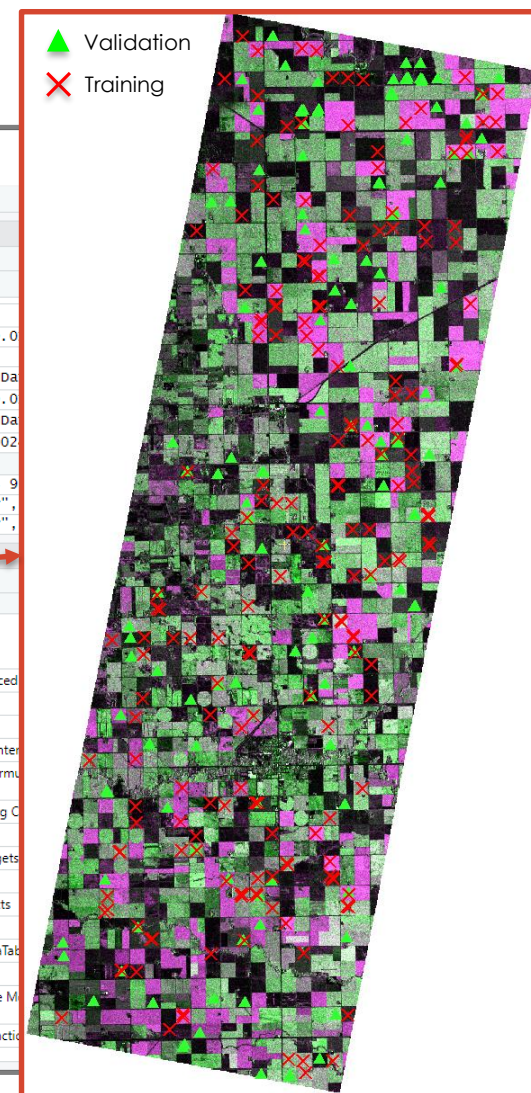
```
> #11) Extract values to be used for independent validation  
> validation_data <- raster::extract(inraster, validation)[,selection]  
> validation_response <- as.factor(validation$class)  
> |
```

The Environment pane on the right shows the following objects:

Object	Class	Attributes
inraster	Formal class RasterStack	
Predictor_Data	num [1:132, 1:22]	0.0374 0.0
r_tree	List of 18	
Training	Formal class SpatialPointsData	
Training_Data	num [1:132, 1:22]	0.0374 0.0
validation	Formal class SpatialPointsData	
validation_data	num [1:300, 1:22]	0.043 0.02

The Values pane shows the following data types:

Object	Class	Attributes
selection	int [1:22]	1 2 3 4 5 6 7 8 9
Training_Response	Factor w/ 6 levels "Barley",	
validation_Response	Factor w/ 6 levels "Barley",	



# Random Forests: Hands on Demonstration

## 16. Classify the independent validation data

The screenshot displays the RStudio interface with the following components:

- Source Editor:** Contains R code for steps 11, 12, and 13. Step 11 extracts validation data, step 12 classifies it, and step 13 generates a confusion matrix.
- Console:** Shows the execution of the R code from step 11, including the extraction of validation data and the classification step.
- Environment:** Lists objects in the global environment, including `inraster`, `Predictor_Data`, `r_tree`, `Training`, `Training_Data`, `validation`, and `validation_data`.
- User Library:** A list of installed R packages with their versions, such as `assertthat` (0.2.1), `backports` (1.1.4), `base64enc` (0.1-3), `BH` (1.69.0-1), `cli` (1.1.0), `coin` (1.3-0), `colorspace` (1.4-1), `crayon` (1.3.4), `crosstalk` (1.0.0), `data.table` (1.12.2), `digest` (0.6.19), `dplyr` (0.8.1), `DT` (0.6), `dtplyr` (0.0.3), `evaluate` (0.14), `fansi` (0.4.0), and `GGally` (1.4.0).



# Random Forests: Hands on Demonstration

## 17. Generate a confusion matrix

The screenshot shows the RStudio interface with the following components:

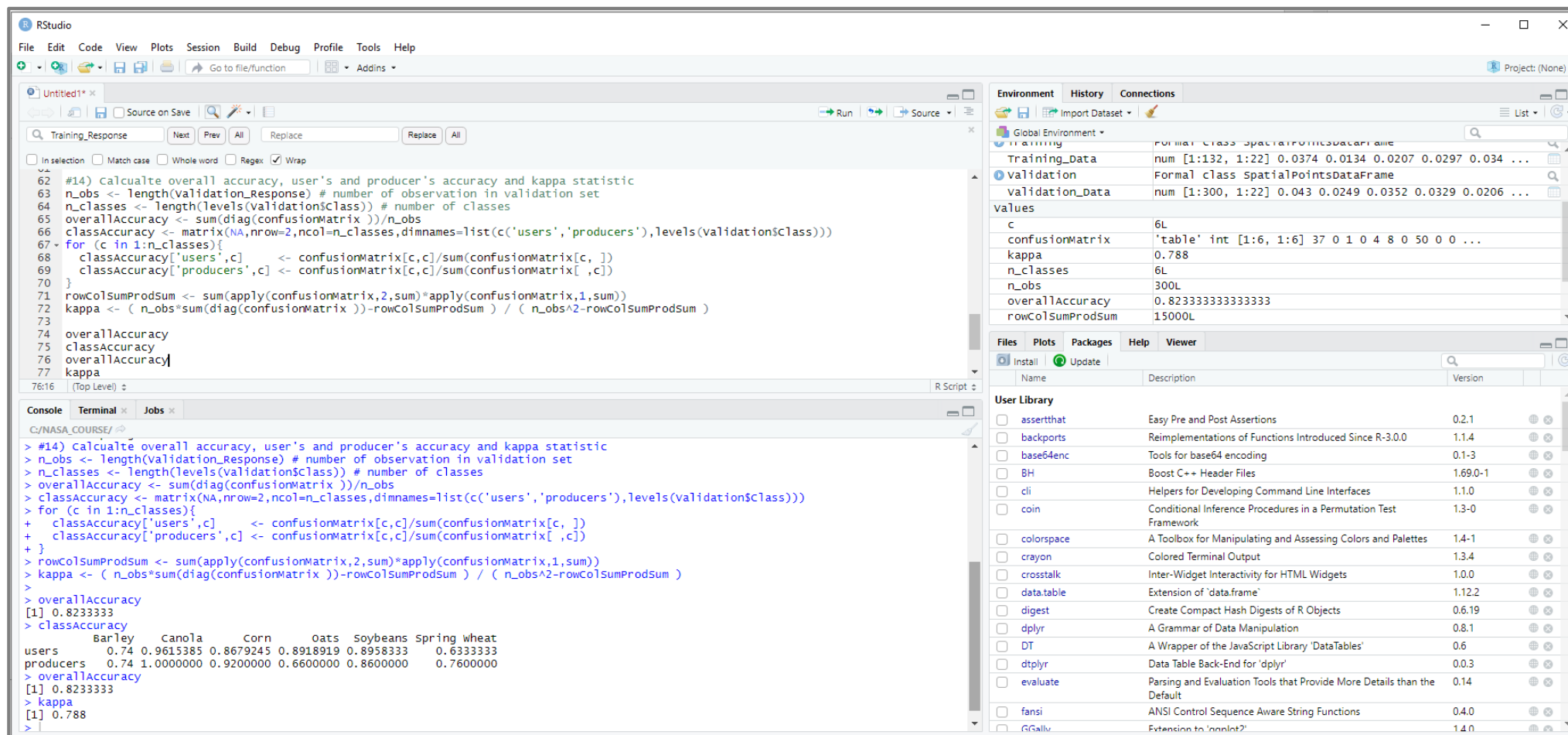
- Source Editor:** Contains R code for generating a confusion matrix from validation data.
- Environment:** Lists objects in the global environment, including `inraster`, `Predictor_Data`, `r_tree`, `Training`, `Training_Data`, `validation`, and `validation_Data`.
- Console:** Shows the execution of the code and the resulting confusion matrix.

```
#11) Extract values to be used for independent validation
> validation_Data <- raster::extract(inraster, validation)[,selection]
> validation_Response <- as.factor(validation$class)
#12) Classify the independent validation data
> validation_Predictions <- predict(r_tree, validation_Data)
#13) Generate a confusion matrix from the independent validation data
> validation_Response <- as.factor(validation$class)
> confusionMatrix <- table(validation_Predictions, validation_Response)
> confusionMatrix
```

	validation_Response					
validation_Predictions	Barley	Canola	Corn	Oats	Soybeans	Spring wheat
Barley	37	0	0	4	0	9
Canola	0	50	0	1	0	1
Corn	1	0	46	1	4	1
Oats	0	0	2	33	1	1
Soybeans	4	0	1	0	43	0
Spring wheat	8	0	1	11	2	38

# Random Forests: Hands on Demonstration

## 18. Calculate the independent accuracies and kappa statistic



```
#14) Calculate overall accuracy, user's and producer's accuracy and kappa statistic
n_obs <- length(validation_response) # number of observation in validation set
n_classes <- length(levels(validation$class)) # number of classes
overallAccuracy <- sum(diag(confusionMatrix)) / n_obs
classAccuracy <- matrix(NA, nrow=2, ncol=n_classes, dimnames=list(c('users', 'producers'), levels(validation$class)))
for (c in 1:n_classes){
  classAccuracy['users',c] <- confusionMatrix[c,c] / sum(confusionMatrix[,c])
  classAccuracy['producers',c] <- confusionMatrix[c,c] / sum(confusionMatrix[,c])
}
rowColSumProdSum <- sum(apply(confusionMatrix, 2, sum) * apply(confusionMatrix, 1, sum))
kappa <- (n_obs * sum(diag(confusionMatrix)) - rowColSumProdSum) / (n_obs^2 - rowColSumProdSum)

overallAccuracy
classAccuracy
classAccuracy
overallAccuracy
kappa
```

Console Output:

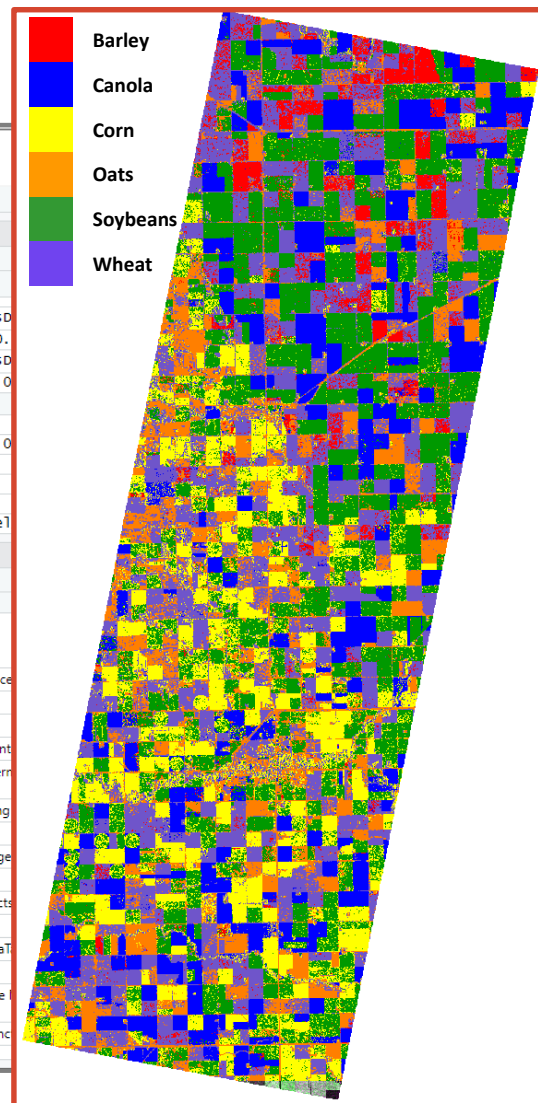
```
> #14) Calculate overall accuracy, user's and producer's accuracy and kappa statistic
> n_obs <- length(validation_response) # number of observation in validation set
> n_classes <- length(levels(validation$class)) # number of classes
> overallAccuracy <- sum(diag(confusionMatrix)) / n_obs
> classAccuracy <- matrix(NA, nrow=2, ncol=n_classes, dimnames=list(c('users', 'producers'), levels(validation$class)))
+ for (c in 1:n_classes){
+   classAccuracy['users',c] <- confusionMatrix[c,c] / sum(confusionMatrix[,c])
+   classAccuracy['producers',c] <- confusionMatrix[c,c] / sum(confusionMatrix[,c])
+ }
> rowColSumProdSum <- sum(apply(confusionMatrix, 2, sum) * apply(confusionMatrix, 1, sum))
> kappa <- (n_obs * sum(diag(confusionMatrix)) - rowColSumProdSum) / (n_obs^2 - rowColSumProdSum)
>
> overallAccuracy
[1] 0.8233333
> classAccuracy
      Barley  Canola  Corn  Oats  Soybeans  Spring  Wheat
users  0.74  0.9615385  0.8679245  0.8918919  0.8958333  0.6333333
producers 0.74  1.0000000  0.9200000  0.6600000  0.8600000  0.7600000
> overallAccuracy
[1] 0.8233333
> kappa
[1] 0.788
```

# Random Forests: Hands on Demonstration

## 19. Classify the whole raster

```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Source on Save
Training_Response
69 }
70 }
71 rowColSumProdSum <- sum(apply(confusionMatrix,2,sum)*apply(confusionMatrix,1,sum))
72 kappa <- ( n_obs*sum(diag(confusionMatrix))-rowColSumProdSum ) / ( n_obs^2-rowColSumProdSum )
73
74 overallAccuracy
75 classAccuracy
76 overallAccuracy
77 kappa
78
79
80 #15) Classify the whole raster
81 OutputRaster <- '3_RESULTS/RS2_TSX_Sentinel1_1_30m_UTM_RandomForest.tif'
82 Predictor_Data <- subset(inraster, Selection)
83 predictions <- predict(inraster[[Selection]],model=r_tree,file=OutputRaster)
84
```

```
> #15) Classify the whole raster
> OutputRaster <- '3_RESULTS/RS2_TSX_Sentinel1_1_30m_UTM_RandomForest.tif'
> Predictor_Data <- subset(inraster, Selection)
> predictions <- predict(inraster[[Selection]],model=r_tree,file=OutputRaster)
```



# References

1. Breiman, Leo. "Random forests." *Machine learning* 45.1 (2001): 5-32.
2. Liaw, Andy, and Matthew Wiener. "Classification and regression by randomForest." *R news* 2.3 (2002): 18-22.
3. <https://victorzhou.com/blog/gini-impurity/>
4. Genuer, Robin, Jean-Michel Poggi, and Christine Tuleau-Malot. "Variable selection using random forests." *Pattern Recognition Letters* 31.14 (2010): 2225-2236.
5. Behnamian, Amir, et al. "A systematic approach for variable selection with random forests: achieving stable variable importance values." *IEEE Geoscience and Remote Sensing Letters* 14.11 (2017): 1988-1992.
6. Banks, Sarah, et al. "Assessing the potential to operationalize shoreline sensitivity mapping: Classifying multiple Wide Fine Quadrature Polarized RADARSAT-2 and Landsat 5 scenes with a single Random Forest model." *Remote Sensing* 7.10 (2015): 13528-13563.
7. Banks, Sarah, et al. "Contributions of Actual and Simulated Satellite SAR Data for Substrate Type Differentiation and Shoreline Mapping in the Canadian Arctic." *Remote Sensing* 9.12 (2017): 1206.
8. Banks, Sarah, et al. "Wetland Classification with Multi-Angle/Temporal SAR Using Random Forests." *Remote Sensing* 11.6 (2019): 670.
9. White, Lori, et al. "Moving to the RADARSAT constellation mission: Comparing synthesized compact polarimetry and dual polarimetry data with fully polarimetric RADARSAT-2 data for image classification of peatlands." *Remote Sensing* 9.6 (2017): 573.
10. Millard, Koreen, and Murray Richardson. "On the importance of training data sample selection in random forest image classification: A case study in peatland ecosystem mapping." *Remote sensing* 7.7 (2015): 8489-8515.
11. Millard, Koreen, and Murray Richardson. "Wetland mapping with LiDAR derivatives, SAR polarimetric decompositions, and LiDAR-SAR fusion using a random forest classifier." *Canadian Journal of Remote Sensing* 39.4 (2013): 290-307.
12. Planet Team (2017). *Planet Application Program Interface: In Space for Life on Earth*. San Francisco, CA. <https://api.planet.com>
13. RADARSAT-2 Data and Products © Maxar Technologies Ltd. (2018) – All Rights Reserved. RADARSAT is an official mark of the Canadian Space Agency.

# Contributors

- Sarah Banks, Environment and Climate Change Canada
- Dr. Amir Behnamian, Environment and Climate Change Canada
- Dr. Koreen Millard, Carleton University, Department of Geography and Environmental Studies
- Dr. Cai, Carleton University, Department of Mathematics and Statistics
- Dr. Liaw, Merck