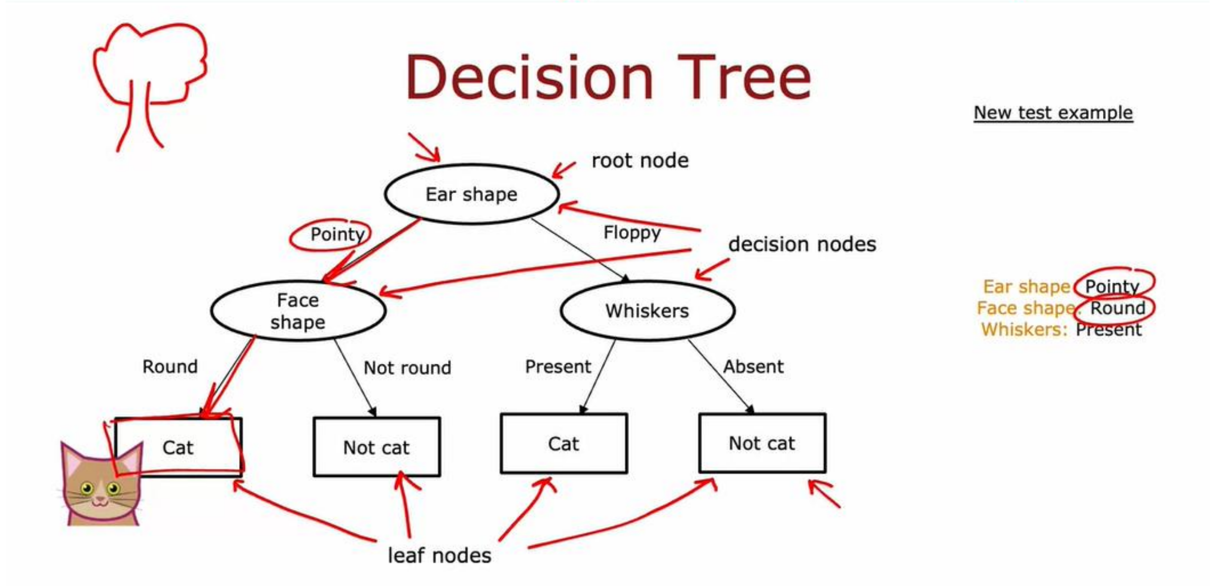
Decision trees

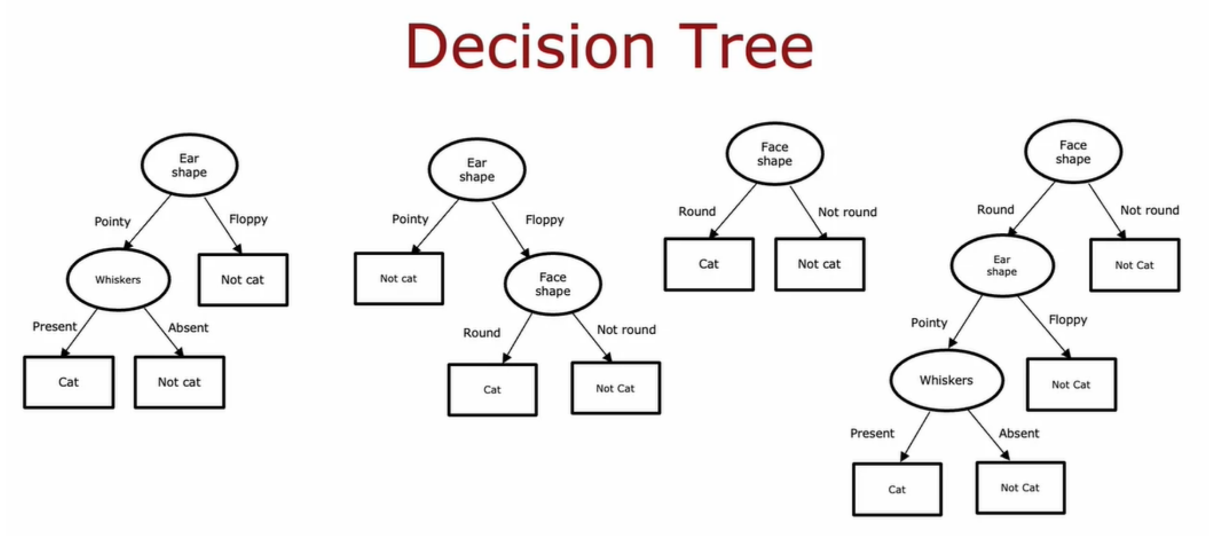
1. Decision tree model

Cat classification example

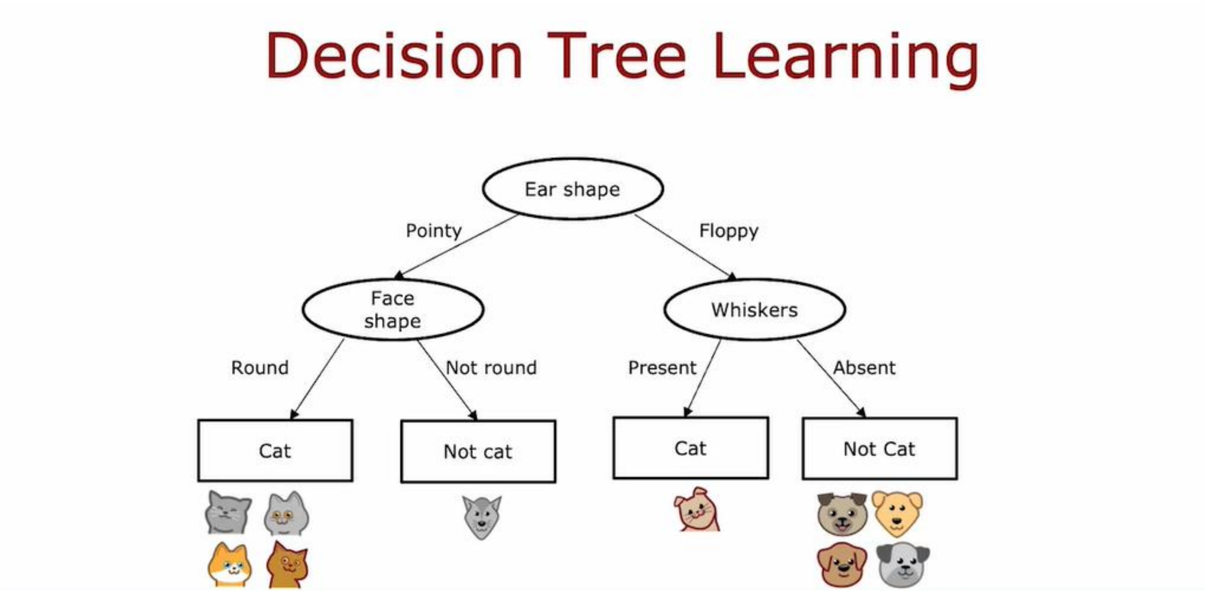
Ear shape (x_1)	Face shape (x_2)	Whiskers (x_3)	Cat
Pointy	Round	Present	1
Floppy	Not round	Present	1
Floppy	Round	Absent	0
Pointy	Not round	Present	0
Pointy	Round	Present	1
Pointy	Round	Absent	1
Floppy	Not round	Absent	0
Pointy	Round	Absent	1
Floppy	Round	Absent	0
Floppy	Round	Absent	0

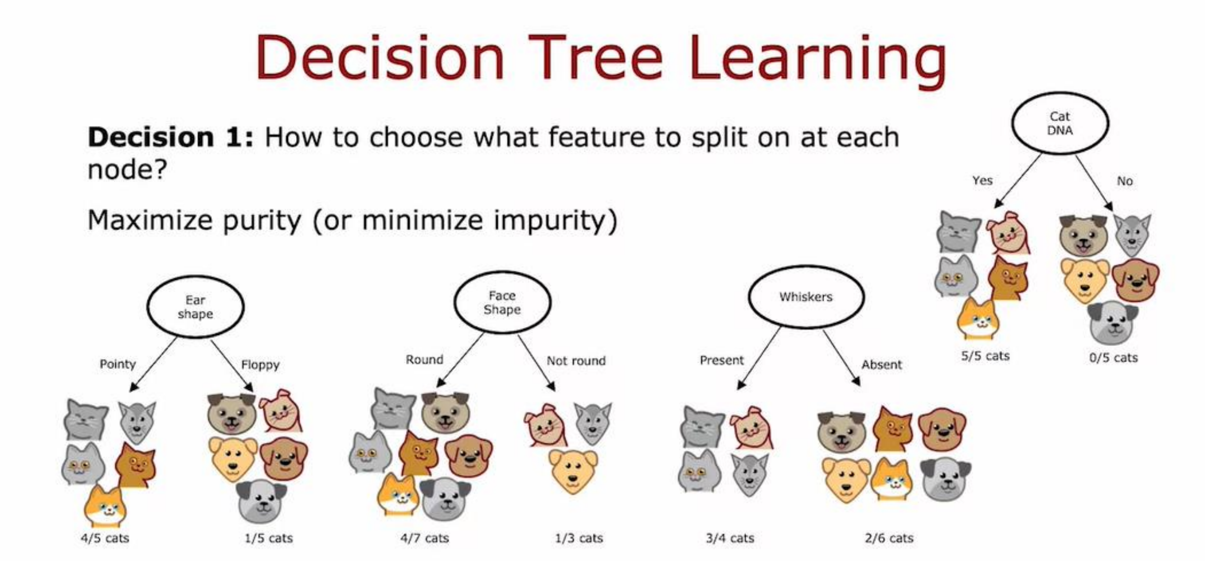
Categorical (discrete values) X y

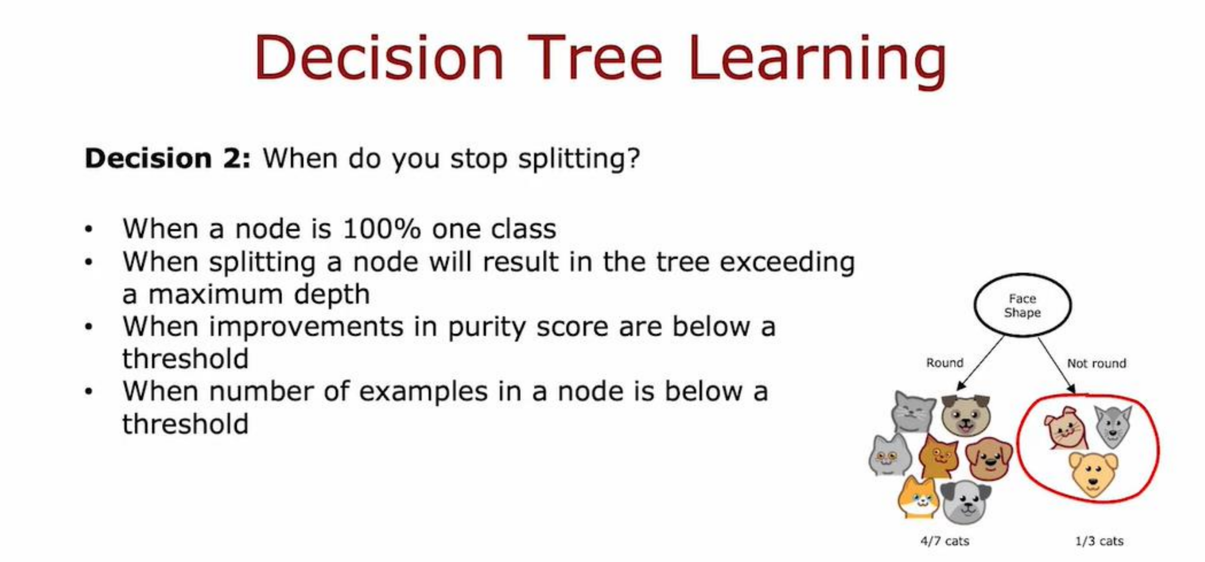




2. Learning process







3. Practice quiz 1

Based on the decision tree shown in the lecture, if an animal has floppy ears, a round face shape and has whiskers, does the model predict that it's a cat or not a cat?

- ○ Not a cat
- ◉ cat

Correct
Correct. If you follow the floppy ears to the right, and then from the whiskers decision node, go left because whiskers are present, you reach a leaf node for "cat", so the model would predict that this is a cat.

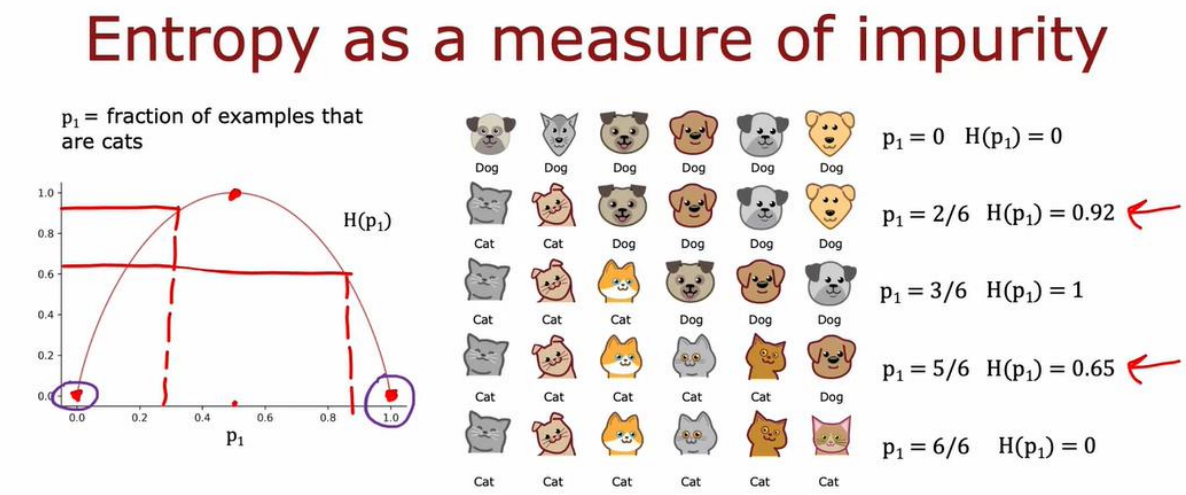
Take a decision tree learning to classify between spam and non-spam email. There are 20 training examples at the root note, comprising 10 spam and 10 non-spam emails. If the algorithm can choose from among four features, resulting in four corresponding splits, which would it choose (i.e., which has highest purity)?

- ◉ Left split: 10 of 10 emails are spam. Right split: 0 of 10 emails are spam.
- ○ Left split: 5 of 10 emails are spam. Right split: 5 of 10 emails are spam.
- ○ Left split: 2 of 2 emails are spam. Right split: 8 of 18 emails are spam.
- ○ Left split: 7 of 8 emails are spam. Right split: 3 of 12 emails are spam.

Correct
Yes!

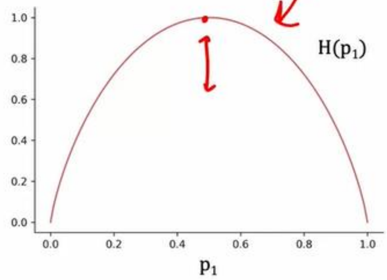
Decision tree learning

1. Measuring purity



Entropy as a measure of impurity

p_1 = fraction of examples that are cats



$$p_0 = 1 - p_1$$

$$H(p_1) = -p_1 log_2(p_1) - p_0 log_2(p_0)$$
$$= -p_1\, log_2(p_1) - (1-p_1) log_2(1-p_1)$$

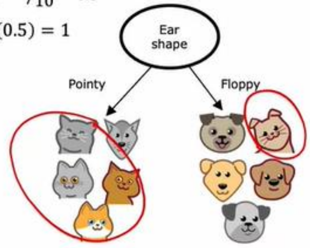
Note: "0 log(0)" = 0

2. Choosing a split: information gain

Choosing a split

$p_1 = {}^{5}/_{10} = 0.5$

$H(0.5) = 1$

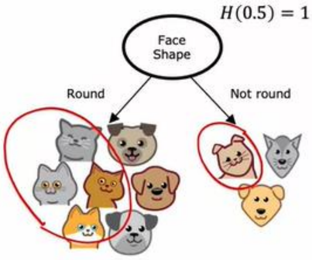


$p_1 = {}^{4}/_{5} = 0.8$ $\quad p_1 = {}^{1}/_{5} = 0.2$

$H(0.8) = 0.72 \quad H(0.2) = 0.72$

$$H(0.5) - \left(\frac{5}{10}H(0.8) + \frac{5}{10}H(0.2)\right)$$
$$= 0.28$$

$H(0.5) = 1$

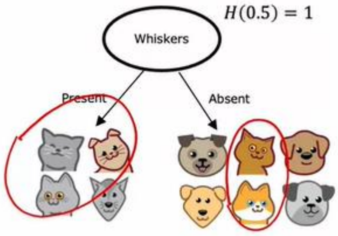


$p_1 = {}^{4}/_{7} = 0.57$ $\quad p_1 = {}^{1}/_{3} = 0.33$

$H(0.57) = 0.99 \quad H(0.33) = 0.92$

$$H(0.5) - \left(\frac{7}{10}H(0.57) + \frac{3}{10}H(0.33)\right)$$
$$= 0.03$$

$H(0.5) = 1$



$p_1 = {}^{3}/_{4} = 0.75$ $\quad p_1 = {}^{2}/_{6} = 0.33$

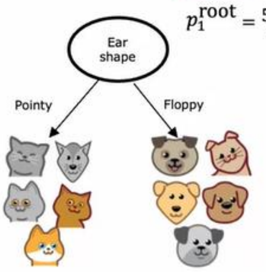
$H(0.75) = 0.81 \quad H(0.33) = 0.92$

$$H(0.5) - \left(\frac{4}{10}H(0.75) + \frac{6}{10}H(0.33)\right)$$
$$= 0.12$$

Information gain

Information Gain

$p_1^{\text{root}} = {}^{5}/_{10} = 0.5$



$p_1^{\text{left}} = {}^{4}/_{5}$ $\quad p_1^{\text{right}} = {}^{1}/_{5}$

$w^{\text{left}} = {}^{5}/_{10}$ $\quad w^{\text{right}} = {}^{5}/_{10}$

Information gain

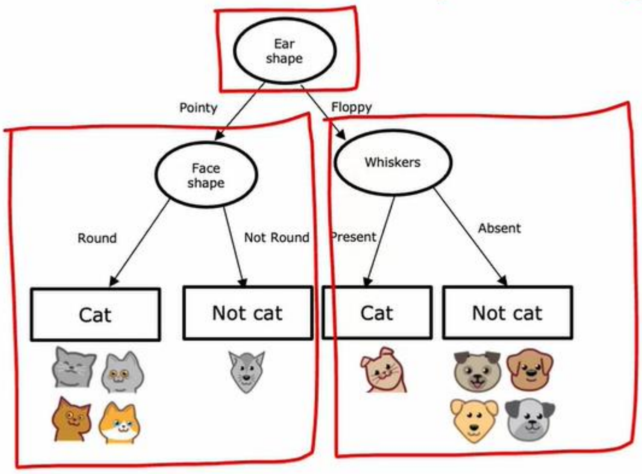
$$= H(p_1^{\text{root}}) - \left(w^{\text{left}} H\left(p_1^{\text{left}}\right) + w^{\text{right}} H\left(p_1^{\text{right}}\right)\right)$$

3. Putting it together

Decision Tree Learning

- Start with all examples at the root node
- Calculate information gain for all possible features, and pick the one with the highest information gain
- Split dataset according to selected feature, and create left and right branches of the tree
- Keep repeating splitting process until stopping criteria is met:
  - When a node is 100% one class
  - When splitting a node will result in the tree exceeding a maximum depth
  - Information gain from additional splits is less than threshold
  - When number of examples in a node is below a threshold

Recursive splitting

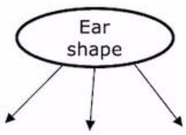


Recursive algorithm

4. Using one-hot ending of categorical features

Features with three possible values

Ear shape (x_1)	Face shape (x_2)	Whiskers (x_3)	Cat (y)
Pointy	Round	Present	1
Oval	Not round	Present	1
Oval	Round	Absent	0
Pointy	Not round	Present	0
Oval	Round	Present	1
Pointy	Round	Absent	1
Floppy	Not round	Absent	0
Oval	Round	Absent	1
Floppy	Round	Absent	0
Floppy	Round	Absent	0



3 possible values

One hot encoding

~~Ear shape~~	Pointy ears	Floppy ears	Oval ears	Face shape	Whiskers	Cat
~~Pointy~~	1	0	0	Round	Present	1
~~Oval~~	0	0	1	Not round	Present	1
~~Oval~~	0	0	1	Round	Absent	0
~~Pointy~~	1	0	0	Not round	Present	0
~~Oval~~	0	0	1	Round	Present	1
~~Pointy~~	1	0	0	Round	Absent	1
~~Floppy~~	0	1	0	Not round	Absent	0
~~Oval~~	0	0	1	Round	Absent	1
~~Floppy~~	0	1	0	Round	Absent	0
~~Floppy~~	0	1	0	Round	Absent	0

One hot encoding

If a categorical feature can take on k values, create k binary features (0 or 1 valued).

One hot encoding and neural networks

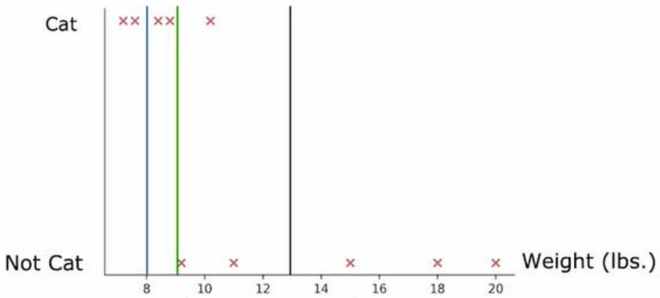
Pointy ears	Floppy ears	Round ears	Face shape	Whiskers	Cat
1	0	0	~~Round~~ 1	~~Present~~ 1	1
0	0	1	~~Not round~~ 0	~~Present~~ 1	1
0	0	1	~~Round~~ 1	~~Absent~~ 0	0
1	0	0	~~Not round~~ 0	~~Present~~ 1	0
0	0	1	~~Round~~ 1	~~Present~~ 1	1
1	0	0	~~Round~~ 1	~~Absent~~ 0	1
0	1	0	~~Not round~~ 0	~~Absent~~ 0	1
0	0	1	~~Round~~ 1	~~Absent~~ 0	1
0	1	0	~~Round~~ 1	~~Absent~~ 0	1
0	1	0	~~Round~~ 1	~~Absent~~ 0	1

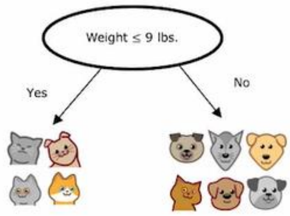
5. Continuous valued features

Continuous features

Ear shape	Face shape	Whiskers	Weight (lbs.)	Cat
Pointy	Round	Present	7.2	1
Floppy	Not round	Present	8.8	1
Floppy	Round	Absent	15	0
Pointy	Not round	Present	9.2	0
Pointy	Round	Present	8.4	1
Pointy	Round	Absent	7.6	1
Floppy	Not round	Absent	11	0
Pointy	Round	Absent	10.2	1
Floppy	Round	Absent	18	0
Floppy	Round	Absent	20	0

Splitting on a continuous variable





$$H(0.5) - \left(\frac{2}{10}H\left(\frac{2}{2}\right) + \frac{8}{10}H\left(\frac{3}{8}\right)\right) = 0.24$$

$$H(0.5) - \left(\frac{4}{10}H\left(\frac{4}{4}\right) + \frac{6}{10}H\left(\frac{1}{6}\right)\right) = 0.61$$

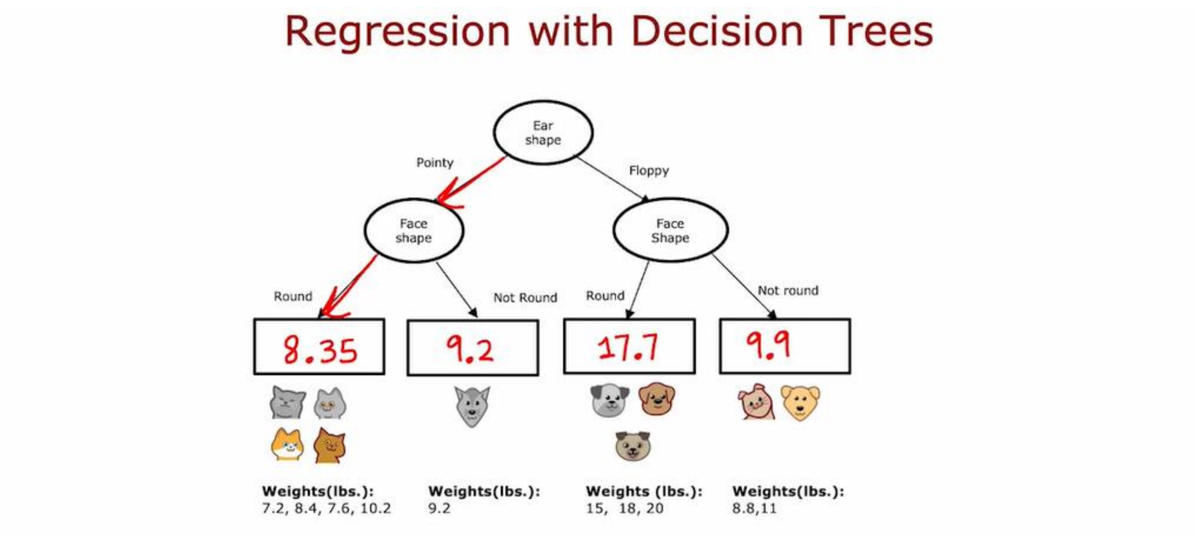
$$H(0.5) - \left(\frac{7}{10}H\left(\frac{5}{7}\right) + \frac{3}{10}H\left(\frac{0}{3}\right)\right) = 0.40$$

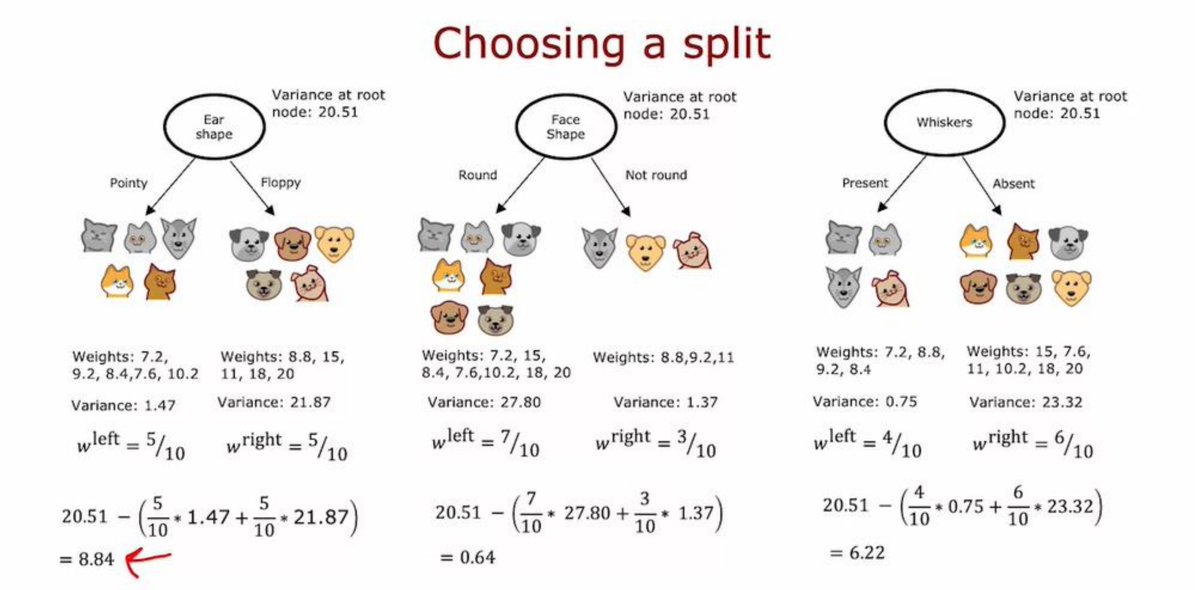
6. Regression trees

Regression with Decision Trees: Predicting a number

Ear shape	Face shape	Whiskers	Weight (lbs.)
Pointy	Round	Present	7.2
Floppy	Not round	Present	8.8
Floppy	Round	Absent	15
Pointy	Not round	Present	9.2
Pointy	Round	Present	8.4
Pointy	Round	Absent	7.6
Floppy	Not round	Absent	11
Pointy	Round	Absent	10.2
Floppy	Round	Absent	18
Floppy	Round	Absent	20

X y





7. Practice quiz

Recall that entropy was defined in lecture as H(p_1) = - p_1 log_2(p_1) - p_0 log_2(p_0), where p_1 is the fraction of positive examples and p_0 the fraction of negative examples.

At a given node of a decision tree, , 6 of 10 examples are cats and 4 of 10 are not cats. Which expression calculates the entropy $H(p_1)$ of this group of 10 animals?

- (●) $-(0.6)log_2(0.6) - (0.4)log_2(0.4)$
- ( ) $(0.6)log_2(0.6) + (0.4)log_2(0.4)$
- ( ) $-(0.6)log_2(0.6) - (1-0.4)log_2(1-0.4)$
- ( ) $(0.6)log_2(0.6) + (1-0.4)log_2(1-0.4)$

Correct

Correct. The expression is $-(p_1)log_2(p_1) - (p_0)log_2(p_0)$

Recall that information was defined as follows:

$$H(p_1^{root}) - \left(w^{left}H(p_1^{left}) + w^{right}H(p_1^{right})\right)$$

Before a split, the entropy of a group of 5 cats and 5 non-cats is $H(5/10)$. After splitting on a particular feature, a group of 7 animals (4 of which are cats) has an entropy of $H(4/7)$. The other group of 3 animals (1 is a cat) and has an entropy of $H(1/3)$. What is the expression for information gain?

- ○ $H(0.5) - (\frac{4}{7} * H(4/7) + \frac{4}{7} * H(1/3))$
- ◉ $H(0.5) - (\frac{7}{10}H(4/7) + \frac{3}{10}H(1/3))$
- ○ $H(0.5) - (H(4/7) + H(1/3))$
- ○ $H(0.5) - (7 * H(4/7) + 3 * H(1/3))$

Correct
Correct. The general expression is $H(p_1^{root}) - \left(w^{left}H(p_1^{left}) + w^{right}H(p_1^{right})\right)$

To represent 3 possible values for the ear shape, you can define 3 features for ear shape: pointy ears, floppy ears, oval ears. For an animal whose ears are not pointy, not floppy, but are oval, how can you represent this information as a feature vector?

- ○ [1,0,0]
- ○ [0, 1, 0]
- ◉ [0, 0, 1]
- ○ [1, 1, 0]

Correct
Yes! 0 is used to represent the absence of that feature (not pointy, not floppy), and 1 is used to represent the presence of that feature (oval).

For a continuous valued feature (such as weight of the animal), there are 10 animals in the dataset. According to the lecture, what is the recommended way to find the best split for that feature?

- ○ Use gradient descent to find the value of the split threshold that gives the highest information gain.
- ◉ Choose the 9 mid-points between the 10 examples as possible splits, and find the split that gives the highest information gain.
- ○ Try every value spaced at regular intervals (e.g., 8, 8.5, 9, 9.5, 10, etc.) and find the split that gives the highest information gain.
- ○ Use a one-hot encoding to turn the feature into a discrete feature vector of 0's and 1's, then apply the algorithm we had discussed for discrete features.

Correct
Correct. This is what is proposed in the lectures.

Which of these are commonly used criteria to decide to stop splitting? (Choose two.)

- [x] When the number of examples in a node is below a threshold

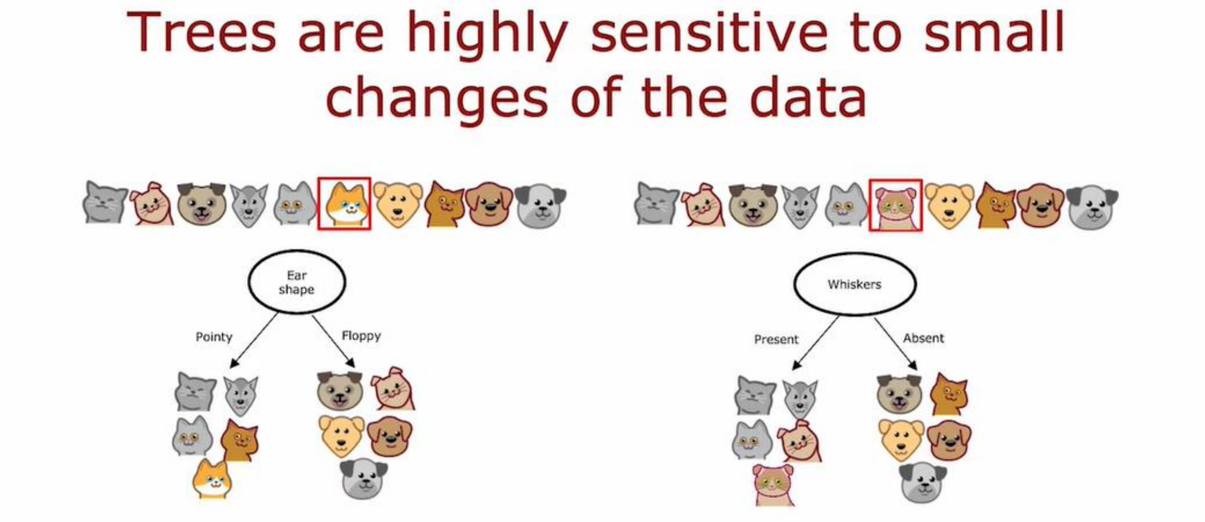
Correct
Yes!

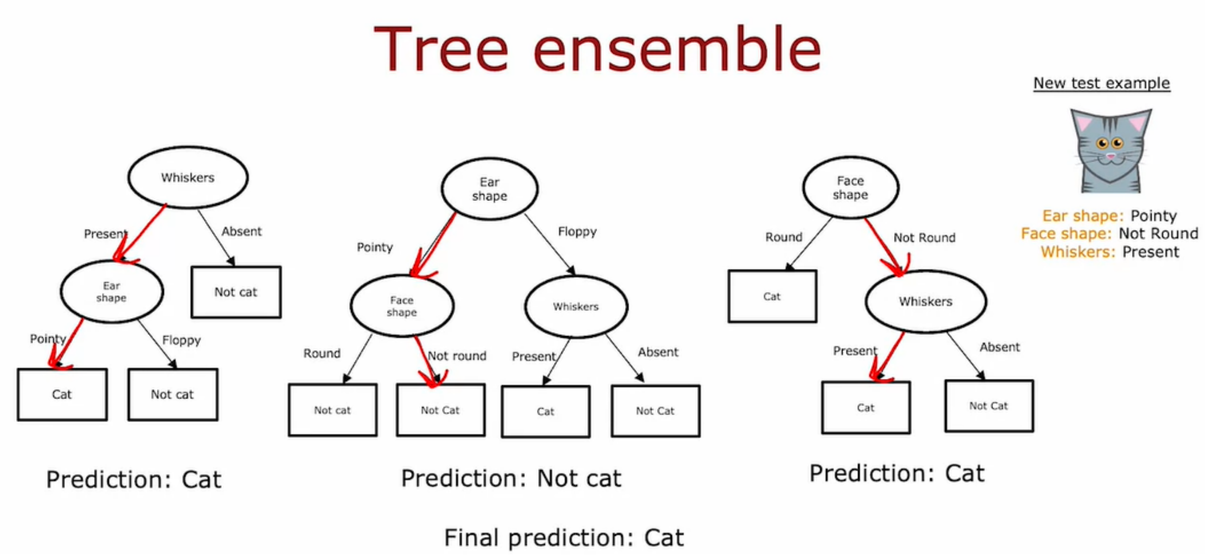
- [ ] When a node is 50% one class and 50% another class (highest possible value of entropy)
- [ ] When the information gain from additional splits is too large
- [x] When the tree has reached a maximum depth

Correct
Yes!

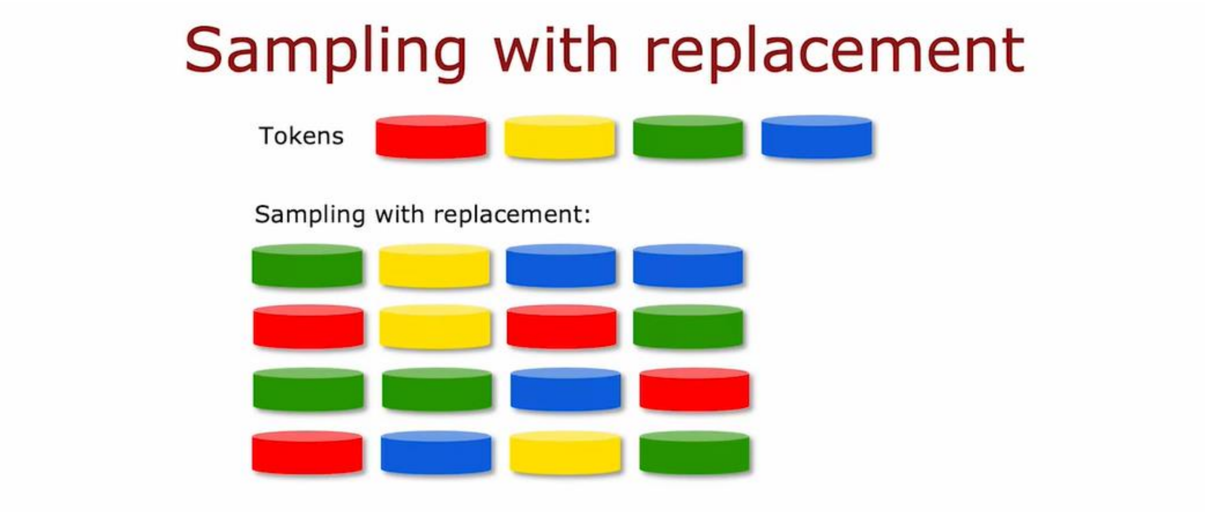
Tree ensembles

1. Using multiple decision trees

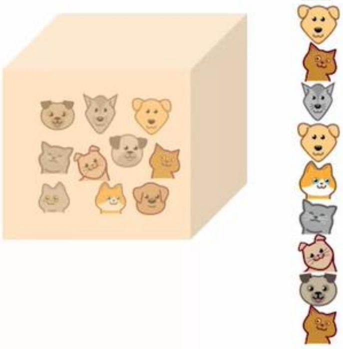




2. Sampling with replacement



Sampling with replacement



Ear shape	Face shape	Whiskers	Cat
Pointy	Round	Present	1
Floppy	Not round	Absent	0
Pointy	Round	Absent	1
Pointy	Not round	Present	0
Floppy	Not round	Absent	0
Pointy	Round	Absent	1
Pointy	Round	Present	1
Floppy	Not round	Present	1
Floppy	Round	Absent	0
Pointy	Round	Absent	1

3. Random forest algorithm

Generating a tree sample

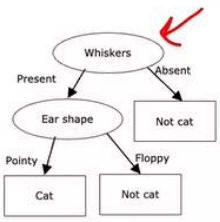
Given training set of size m

For $b = 1$ to B:

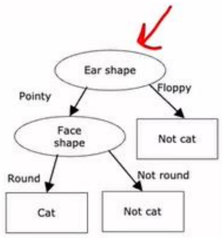
Use sampling with replacement to create a new training set of size m

Train a decision tree on the new dataset

Ear shape	Face shape	Whiskers	Cat
Pointy	Round	Present	Yes
Floppy	Round	Absent	No
Floppy	Round	Absent	No
Pointy	Round	Present	Yes
Pointy	Not Round	Present	Yes
Floppy	Round	Absent	No
Floppy	Round	Present	Yes
Pointy	Not Round	Absent	No
Pointy	Not Round	Absent	No
Pointy	Not Round	Present	Yes



Ear shape	Face shape	Whiskers	Cat
Pointy	Round	Present	Yes
Pointy	Round	Absent	Yes
Floppy	Not Round	Absent	No
Floppy	Not Round	Absent	No
Pointy	Round	Absent	Yes
Floppy	Round	Absent	No
Floppy	Round	Absent	No
Floppy	Round	Absent	No
Pointy	Not Round	Absent	No
Pointy	Round	Present	Yes



...

Bagged decision tree

Randomizing the feature choice

At each node, when choosing a feature to use to split, if n features are available, pick a random subset of $k < n$ features and allow the algorithm to only choose from that subset of features.

$K = \sqrt{n}$

Random forest algorithm

4. XGBoost

Boosted trees intuition

Given training set of size m

For $b = 1$ to $\underline{B}$:

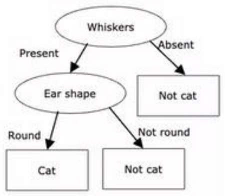
Use sampling with replacement to create a new training set of size m

But instead of picking from all examples with equal (1/m) probability, make it more likely to pick misclassified examples from previously trained trees

Train a decision tree on the new dataset

$\underbrace{1, 2, \ldots, b-1}$ b

Ear shape	Face shape	Whiskers	Cat
Pointy	Round	Present	Yes
Floppy	Round	Absent	No
Floppy	Round	Absent	No
Pointy	Round	Present	Yes
Pointy	Not Round	Present	Yes
Floppy	Round	Absent	No
Floppy	Round	Present	Yes
Pointy	Not Round	Absent	No
Pointy	Not Round	Absent	No
Pointy	Not Round	Present	Yes



Ear shape	Face shape	Whiskers	Prediction
Pointy	Round	Present	Cat ✅
Floppy	Not Round	Present	Not cat ❌ ←
Floppy	Round	Absent	Not cat ✅
Pointy	Not Round	Present	Not cat ✅
Pointy	Round	Present	Cat ✅
Pointy	Round	Absent	Not cat ❌ ←
Floppy	Not Round	Absent	Not cat ✅
Pointy	Round	Absent	Not cat ❌ ←
Floppy	Round	Absent	Not cat ✅
Floppy	Round	Absent	Not cat ✅

XGBoost (eXtreme Gradient Boosting)

- Open source implementation of boosted trees
- Fast efficient implementation
- Good choice of default splitting criteria and criteria for when to stop splitting
- Built in regularization to prevent overfitting
- Highly competitive algorithm for machine learning competitions (eg: Kaggle competitions)

Using XGBoost

Classification

```
from xgboost import XGBClassifier

model = XGBClassifier()

model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

Regression

```
from xgboost import XGBRegressor

model = XGBRegressor()

model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

5. When to use decision trees

Decision Trees vs Neural Networks

Decision Trees and Tree ensembles

- Works well on tabular (structured) data
- Not recommended for unstructured data (images, audio, text)
- Fast
- Small decision trees may be human interpretable

Neural Networks

- Works well on all types of data, including tabular (structured) and unstructured data
- May be slower than a decision tree
- Works with transfer learning
- When building a system of multiple models working together, it might be easier to string together multiple neural networks

6. Practice quiz

For the random forest, how do you build each individual tree so that they are not all identical to each other?

- ○ If you are training B trees, train each one on 1/B of the training set, so each tree is trained on a distinct set of examples.
- ○ Train the algorithm multiple times on the same training set. This will naturally result in different trees.
- ○ Sample the training data without replacement
- ◉ Sample the training data with replacement and select a random subset of features to build each tree

Correct

Correct. You can generate a training set that is unique for each individual tree by sampling the training data with replacement. The random forest algorithm further avoids identical trees by randomly selecting a subset of features when building the tree ensemble.

2.

You are choosing between a decision tree and a neural network for a classification task where the input x is a 100x100 resolution image. Which would you choose?

- ○ A decision tree, because the input is unstructured and decision trees typically work better with unstructured data.
- ○ A decision tree, because the input is structured data and decision trees typically work better with structured data.
- ○ A neural network, because the input is structured data and neural networks typically work better with structured data.
- ◉ A neural network, because the input is unstructured data and neural networks typically work better with unstructured data.

Correct
Yes!

3.

What does sampling with replacement refer to?

- ○ It refers to using a new sample of data that we use to permanently overwrite (that is, to replace) the original data.
- ○ Drawing a sequence of examples where, when picking the next example, first remove all previously drawn examples from the set we are picking from.
- ○ It refers to a process of making an identical copy of the training set.
- ◉ Drawing a sequence of examples where, when picking the next example, first replacing all previously drawn examples into the set we are picking from.

Correct
Yes!