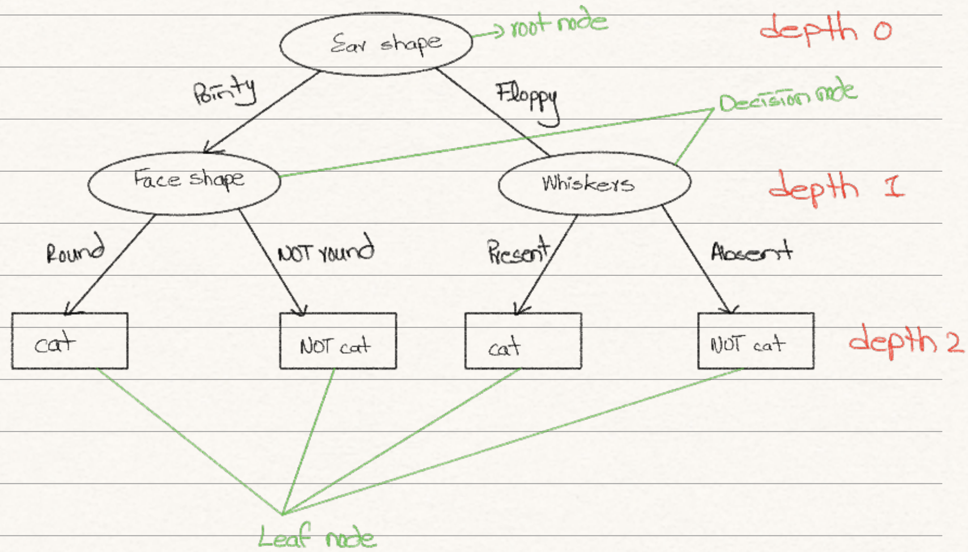
Subject: Decision tree model

Cat classification model



Learning Process:

- Which feature to use in nodes?
  - ↳ Maximize purity / Minimize impurity

- When do you stop splitting?
  - ↳ When a node is 100% one class
  - ↳ When further splitting will exceed max depth
  - ↳ When improvements in purity are below a threshold
  - ↳ When number of examples in a node is below a threshold

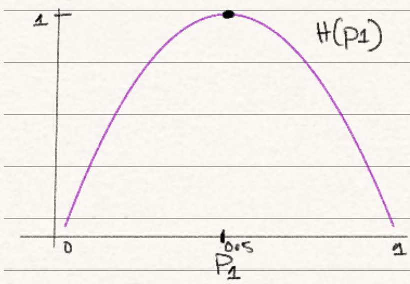
Subject:

Decision tree learning:

Entropy as measure of impurity

p_1 = fraction of cats

p_0 = fraction of NOT cats



$p_1 = \frac{3}{6}$ $\quad H(p_1) = 1$

$p_1 = \frac{5}{6}$ $\quad H(p_1) = 0.65$

$p_1 = \frac{6}{6}$ $\quad H(p_1) = 0$

$p_1 = 0$ $\quad H(p_1) = 0$

$$p_0 = 1 - p_1$$

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

$$= -p_1 \log_2(p_1) - (1-p_1)\log_2(1-p_1)$$

Choosing a split:

$p_1 = \frac{5}{10}$ $\quad H(p_1) = 1$ } Root impurity

Ear shape:

$p_1 = \frac{4}{5}$ $\quad p_1 = \frac{1}{5}$

$H(p_1) = 0.72$ $\quad H(p_1) = 0.72$

Face shape

$p_1 = \frac{4}{7}$ $\quad p_1 = \frac{1}{3}$

$H(p_1) = 0.99$ $\quad H(p_1) = 0.92$

Whiskers

$p_1 = \frac{3}{4}$ $\quad p_1 = \frac{2}{6}$

$H(p_1) = 0.81$ $\quad H(p_1) = 0.92$

Subject:

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

↳ Information gain

General form:

p_1^{left} = fraction of cats in left sub branch

w^{left} = fraction out of all examples that went to left sub branch

p_1^{right} = fraction of cats in right sub branch

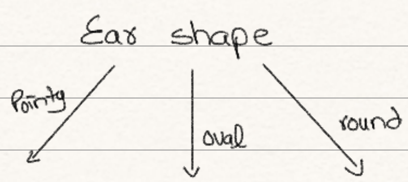
w^{right} fraction out of all examples that went to right sub branch

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

Putting it all together:

- 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

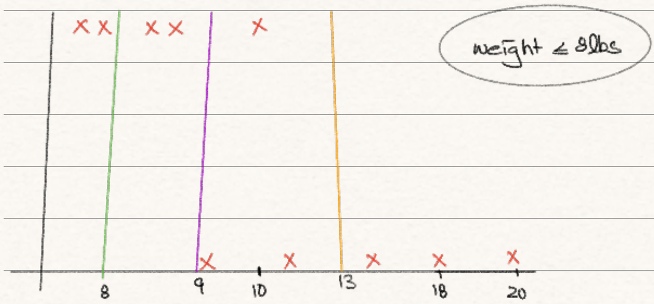
One hot encoding of categorical features:



Subject: / /

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

Continuous valued features:



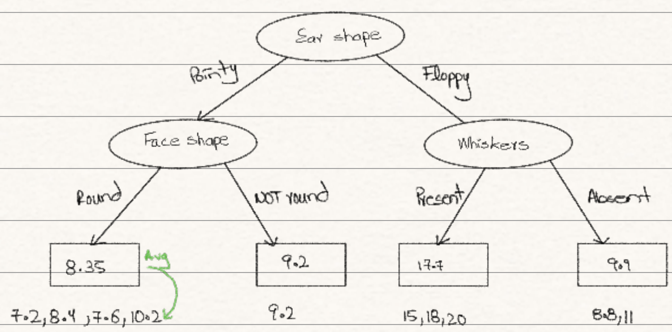
$$H(0.5) - \left(\frac{2}{10}H\left(\frac{2}{2}\right) + \frac{8}{10}H\left(\frac{3}{8}\right)\right) = 0.24$$

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

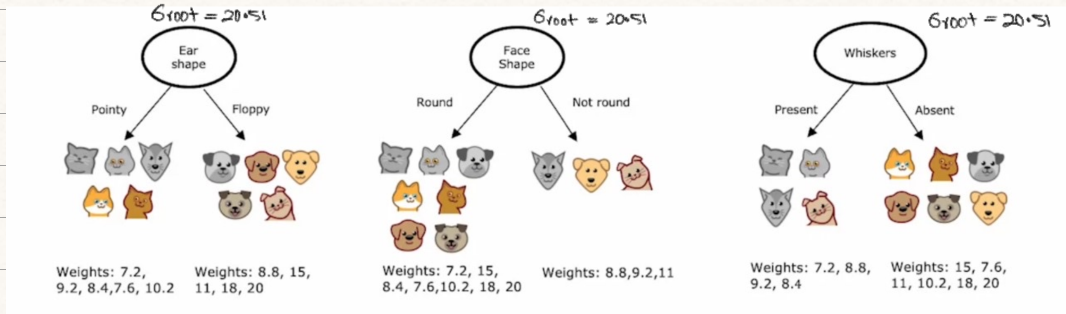
$$H(0.5) - \left(\frac{7}{10}H\left(\frac{5}{7}\right) + \frac{3}{10}H\left(\frac{0}{3}\right)\right) = 0.40$$

Regression Trees:

- Weight is target



Subject:



- Instead of reducing entropy, we reduce the variance b/w weights

Reduction in variance $= \sigma_{root} - \left(w^{left} * \sigma^{left} + w^{right} * \sigma^{right} \right)$

↳ Choose the with largest this

Tree ensembles:

- Trees are highly sensitive to small changes in data
- Building multiple trees
- We choose majority to make a prediction

Sampling with replacement:

- Pick one from data
- Put it back
- Pick again

Random forest Algorithm:

- Given a training set of size m

For b=1 to B:

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

Subject: Train the decision tree on the new dataset / /

- Setting B to larger doesn't hurt but more than 100 doesn't improve performance any further
- Above is called "Bagged decision tree"
- Sometimes, you get same splits in the some/all trees
- So, Randomize feature choice
- We choose a subset of features & then choose from the subset "k".
- For large n, $k = \sqrt{n}$
- The above is called "Random forest algorithm"

"Where does an ML engineer go camping?
a Random forest"
— Andrew Ng

XGBoost - eXtreme Gradient Boosting:

- Given a training set of size m

  For b=1 to B:
  - Use sampling with replacement to create a new training set of size m.
  - Train the decision tree on the new dataset
  - New dataset should be of the misclassified points from previous tree.

When to use decision tree?

- Works well on structured (tabular) data.