UNIT 4:
Supervised Learning

Prepared by Nima Dema

K- Nearest Neighbours

K- Nearest Neighbours

- Used for both regression as well as classification problem.
- To make prediction for new data point, the algorithm find the closest data points in the training dataset-its "nearest neighbours."
- Consider an arbitrary number, k, of neighbours, K-nearest Neighbours.
- Distance measures are used to determine which of the K instances in the training dataset is most similar to the new input.

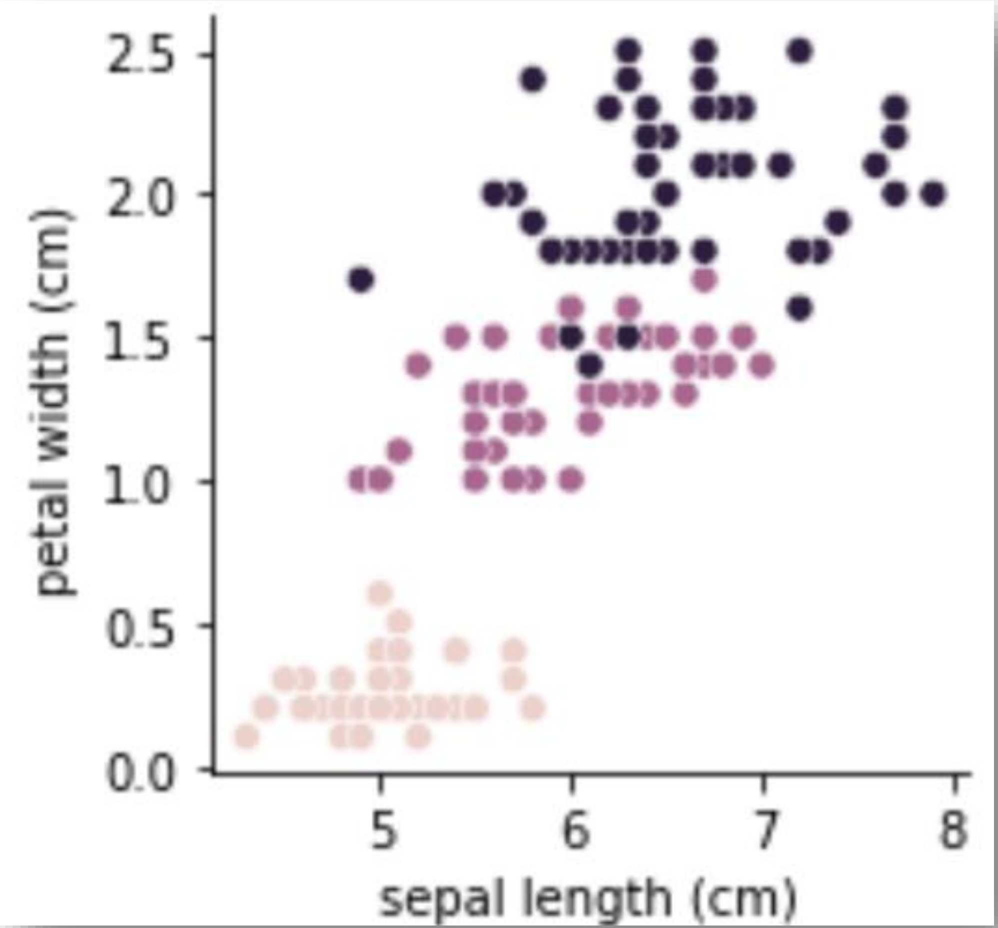
KNN for classification

When k = 1

Probability = $\frac{\text{no of first class}}{\text{total k}}$

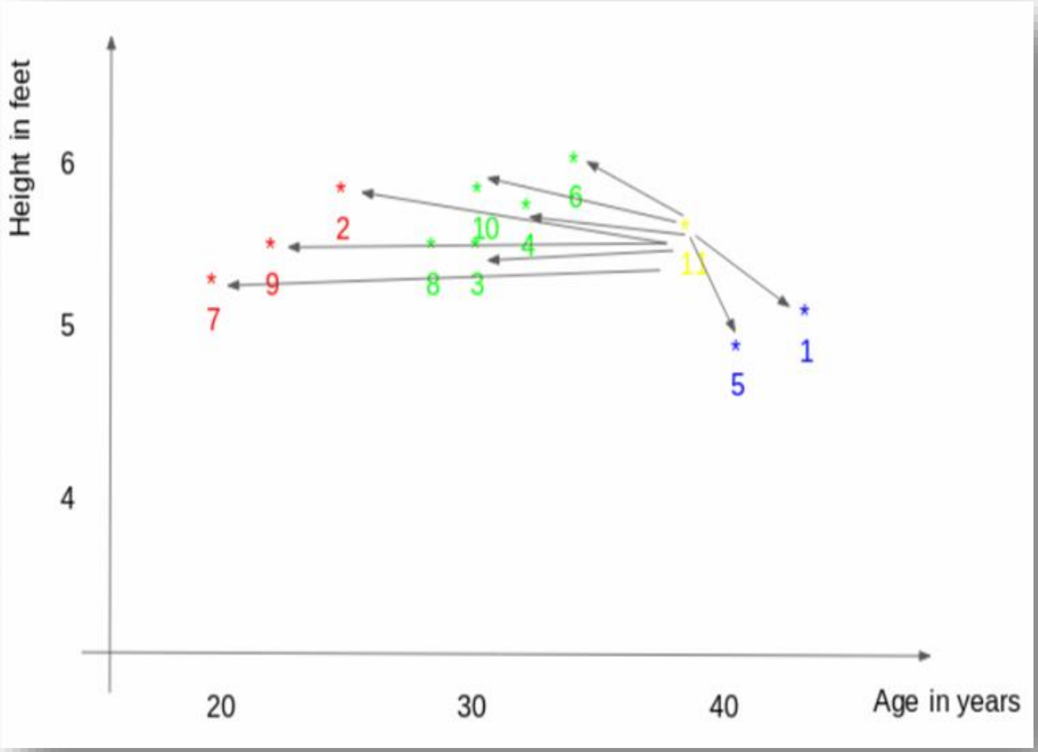
When k = 3

probability(class setosa) = 2/3



KNN for Regression

- First, the distance between the new point and each training point is calculated.
- The closest k data points are selected (based on the distance)
- The average of these data points is the final prediction for the new point.



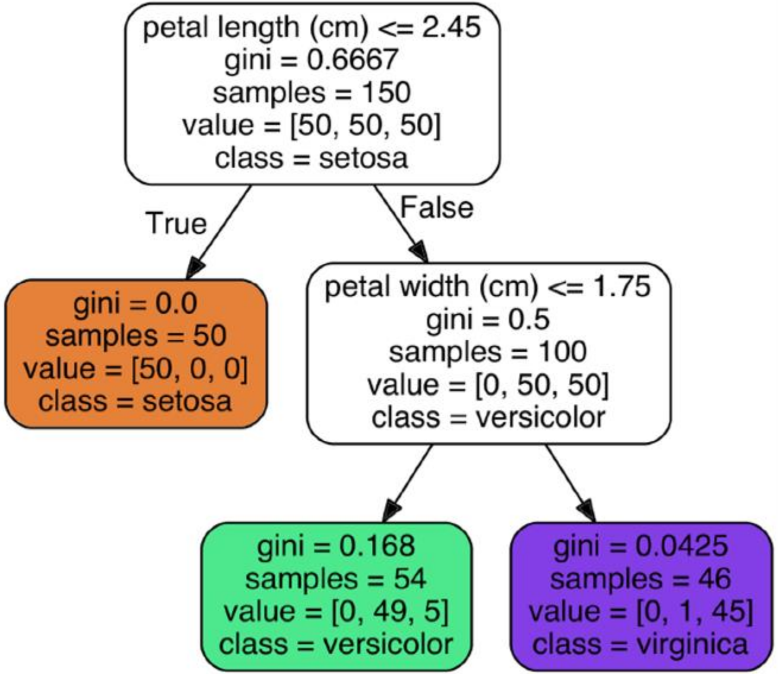
Advantage and Disadvantages

- Advantages:
  - No training period
  - Easy Implementation
- Disadvantages
  - High cost in calculating distance for large datasets
  - Becomes difficult for the algorithm to calculate the distance in high dimension.
  - Need feature scaling

Decision Tree

Decision Trees (CART)

- Decision trees can be used for classification as well as regression problems.
- The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features
- The name itself suggests that it uses a flowchart like a tree structure to show the predictions that result from a series of feature-based splits.
- It starts with a root node and ends with a decision made by leaves.

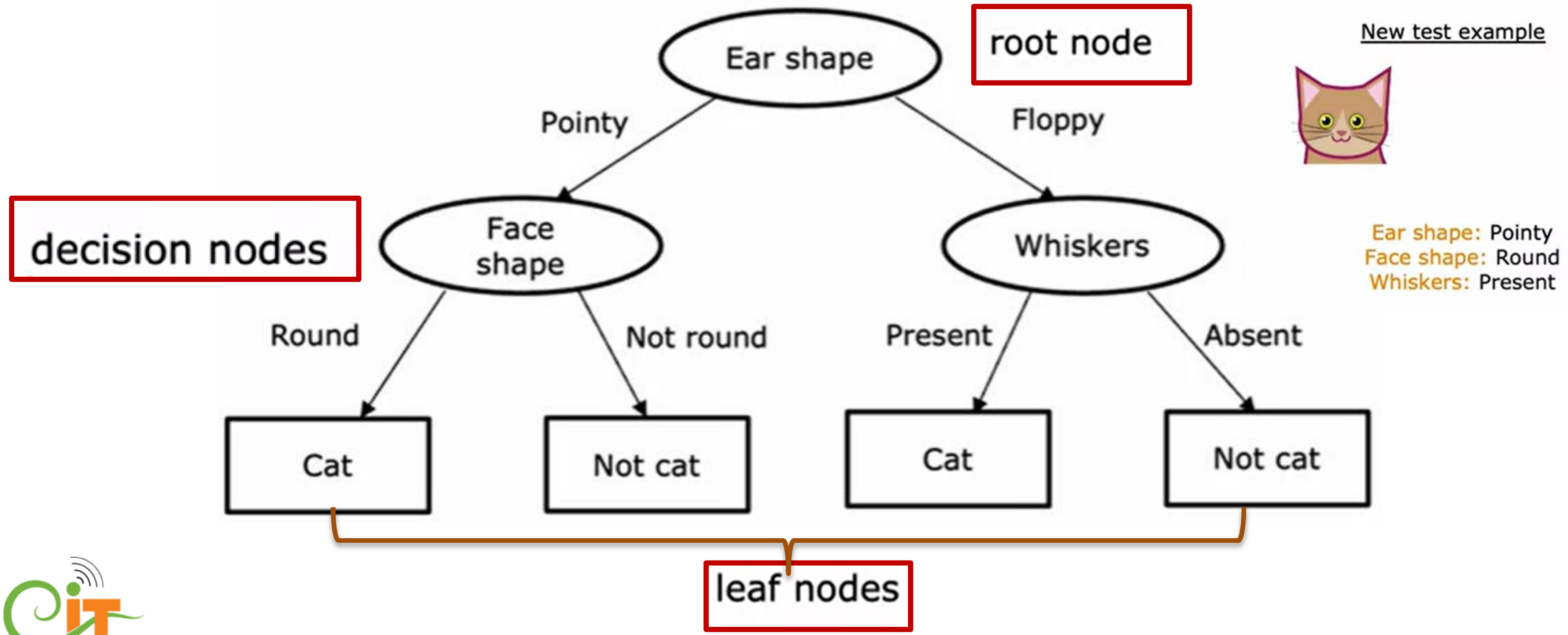


Building Decision Tree for Classification Problem

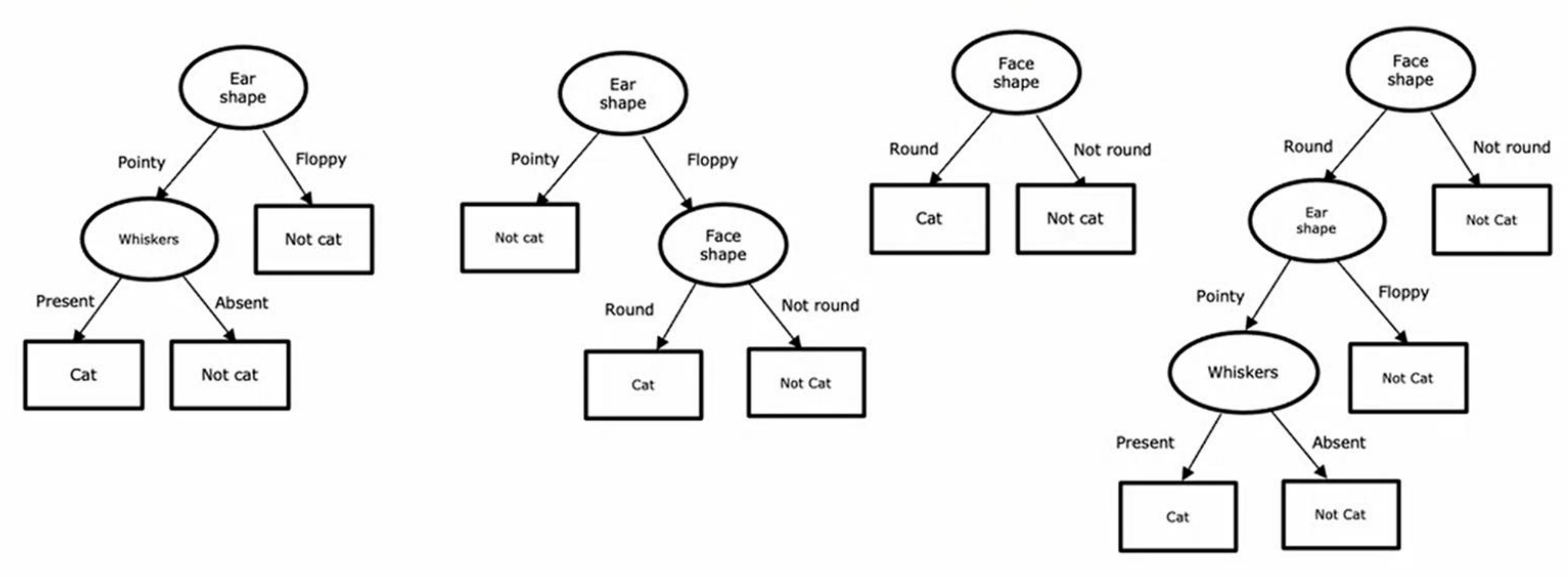
Ear shape	Face shape	Whiskers	Cat
Pointy	Round	Present	1
Floppy	Not round	Present	1
Floppy	Round	Absent	0
Pointy	Not round	Present	0
Pointy	Round	Present	1
Pointy	Round	Absent	1
Floppy	Not round	Absent	0
Pointy	Round	Absent	1
Floppy	Round	Absent	0
Floppy	Round	Absent	0

X y

Building Decision Tree for Classification Problem

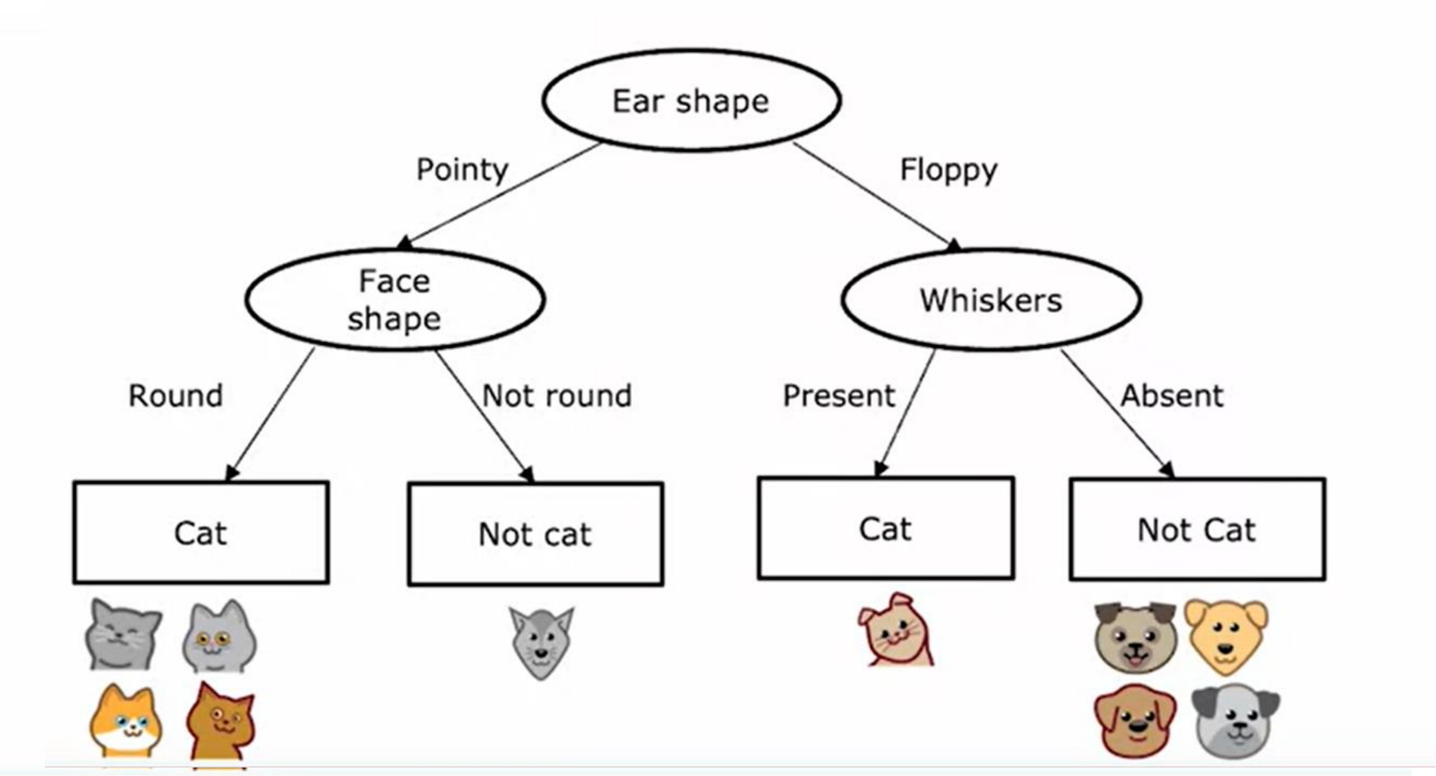


Building Decision Tree for Classification Problem



There are several possible way to build decision tree. Job of the decision tree model is to get best possible model that does well on training set as well as test set.

Decision Tree Building Process

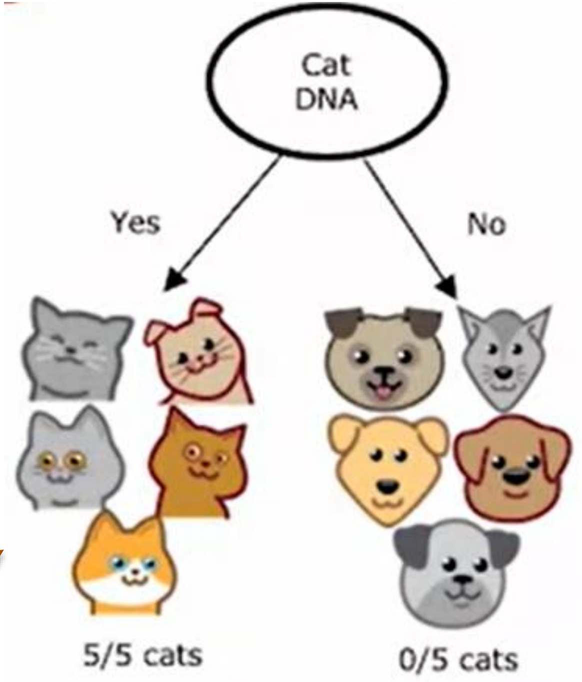


Decision Tree Learning

- While building the decision tree, there were couple of key decision that algorithm had to make.

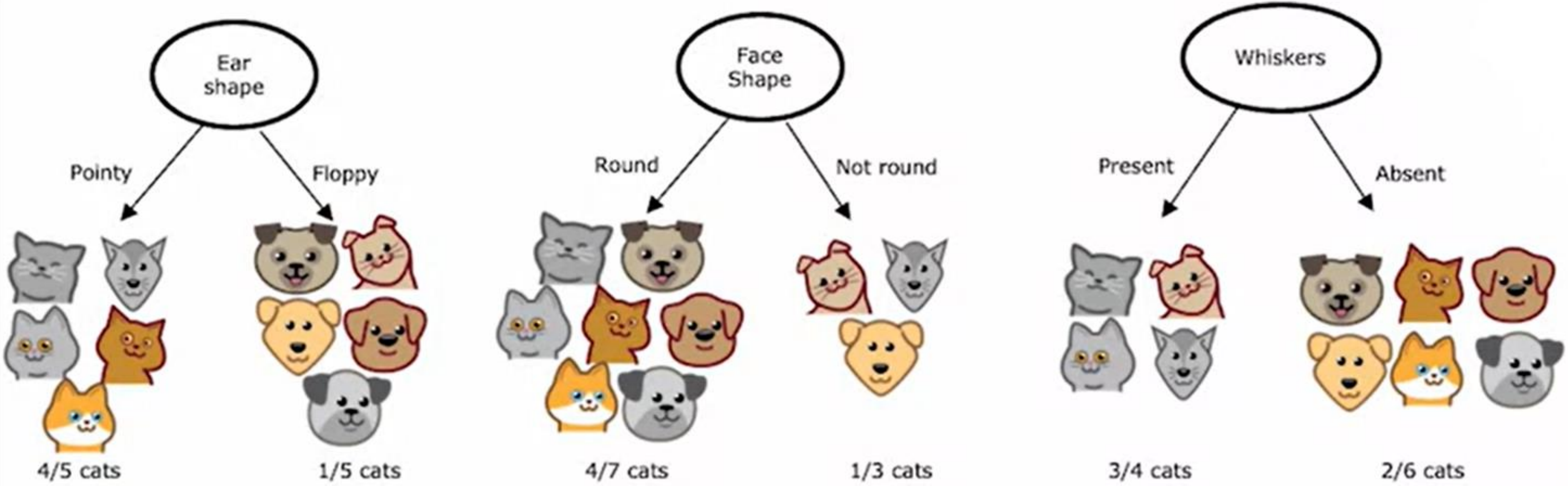
1. **Decision 1:** How to choose what feature to split on at each node?
    - Decision tree will choose feature to **maximize purity (Minimize impurity)**

Example of Pure node.



Decision Tree Learning

Choose best feature which gives more purity at both side of the node



Decision Tree Learning

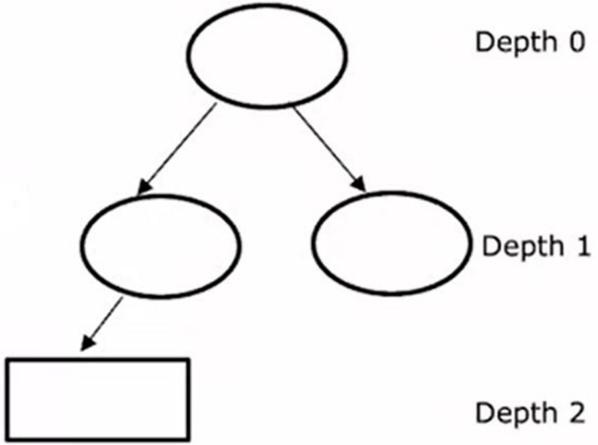
2. **Decision 2:** When do you stop splitting?

- when a node is 100% one/single class.

- When splitting a node will result in the tree exceeding a maximum depth.

- When improvements in purity score are below a threshold

-when number of examples in a node is below a threshold



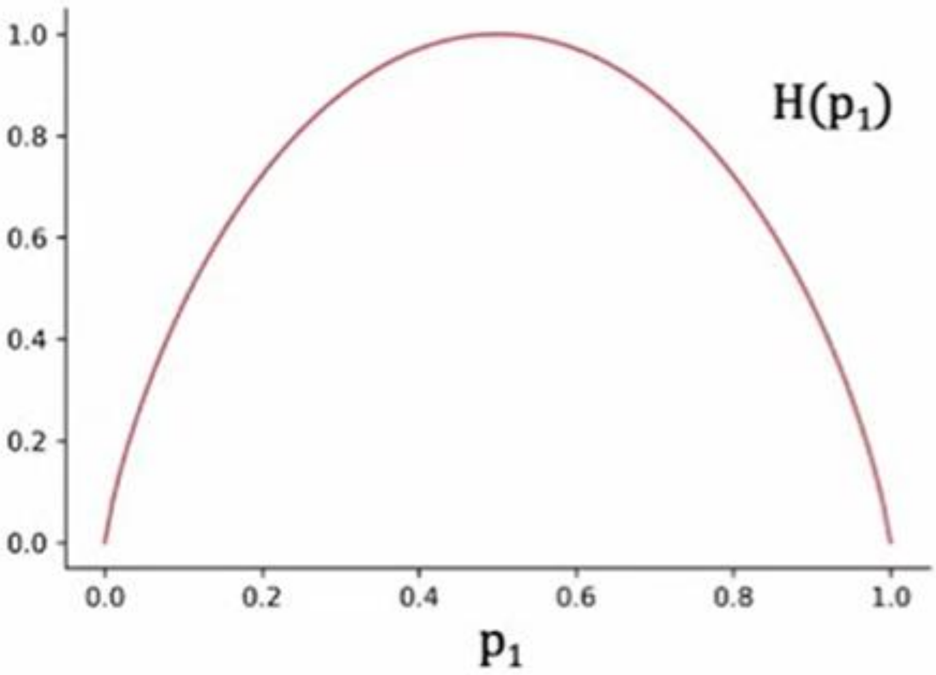
Why limit the tree depth?

- To make sure that tree doesn't become too big and complex

- By keeping tree small, its less prone to overfitting

GYALPOZHING
COLLEGE OF INFORMATION TECHNOLOGY

Entropy as a measure of Impurity



$$I_H = -\sum_{j=1}^{c} p_j log_2(p_j)$$

- Entropy ranges from 0 to 1.
- Entropy value 0 shows purity
- Entropy value 1 shows highest Impurity.
- When p = 3/6 = 0.5, the Entropy value correspond to 1 showing total impure node.
- P = 0/5 or p = 5/5 corresponds to entropy value 0 showing pure nodes.

Entropy as a measure of Impurity

- To decide feature in each node, there is a metric called "Entropy" which is the amount of uncertainty/impurity in the dataset.
- Entropy is an information theory metric that measures the impurity or uncertainty in a group of observations. It **determines how a decision tree chooses to split data.**

P1 = fractions of examples that are cats

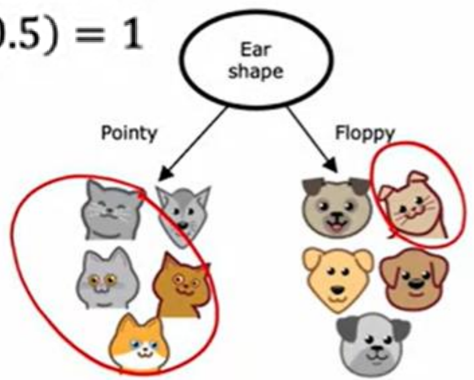
Measure impurity of set of example using a function called **Entropy**

Choosing a split: Information Gain

- Choice of feature at each node depends on choice of feature that reduces the entropy the most or reduces impurity, or maximizes purity.
- In decision tree learning reduction of entropy is called **Information gain**.
- Information gain is useful, when you need to decide on which attributes tells you the most information about the variable upon being presented with sets of features about your random variable.
- When building decision trees, placing attributes with the highest information gain at the top of the tree will lead to the highest quality decision tree.

Choosing a split: Information Gain

$H(0.5) = 1$



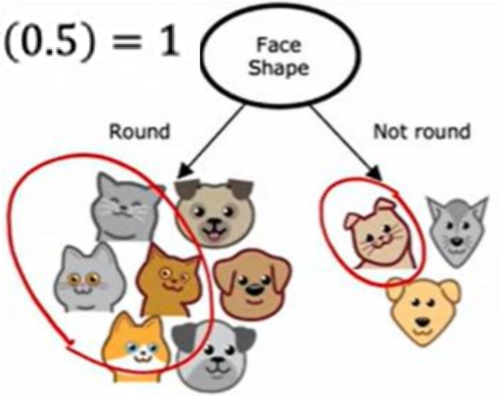
$p_1 = {}^4/_5 = 0.8$ $\quad p_1 = {}^1/_5 = 0.2$

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

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

$$= 0.28$$

$H(0.5) = 1$



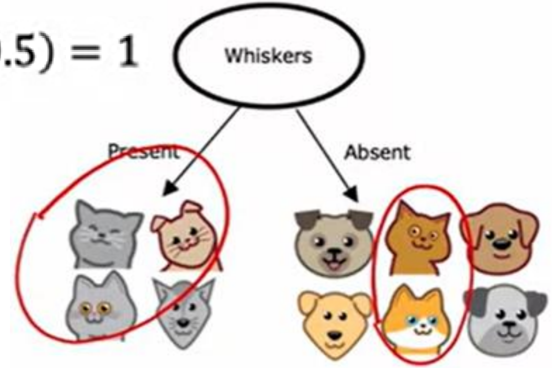
$p_1 = {}^4/_7 = 0.57$ $\quad p_1 = {}^1/_3 = 0.33$

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

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

$$= 0.03$$

$H(0.5) = 1$



$p_1 = {}^3/_4 = 0.75$ $\quad p_1 = {}^2/_6 = 0.33$

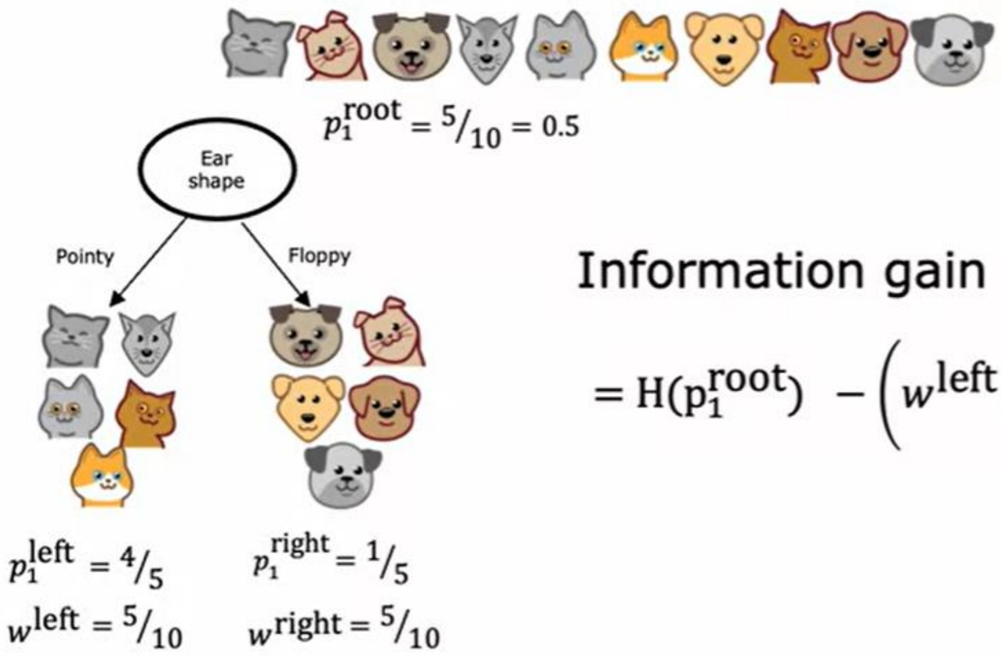
$H(0.75) = 0.81$ $\quad H(0.33) = 0.92$

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

$$= 0.12$$

$$\underbrace{0.28 \qquad 0.03 \qquad 0.12}_{\text{Information Gain}}$$

Information Gain



Information gain

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

Decision Tree Learning

- Start with all examples at the root node.
- Calculate information gain for all possible features, pick the one with highest information gain.
- Split the 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.

Gini Impurity

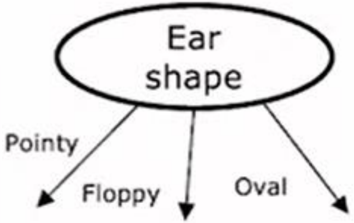
- Gini Impurity is a measurement used to build Decision Trees to determine how the features of a dataset should split nodes to form the tree.

- Gini Impurity for a feature = Weighted average of Gini impurities for the leaves node.

$$\boxed{I_G = 1 - \sum_{j=1}^{c} p_j^2}$$

Multiple labels categorical Features

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



Multiple labels categorical Features

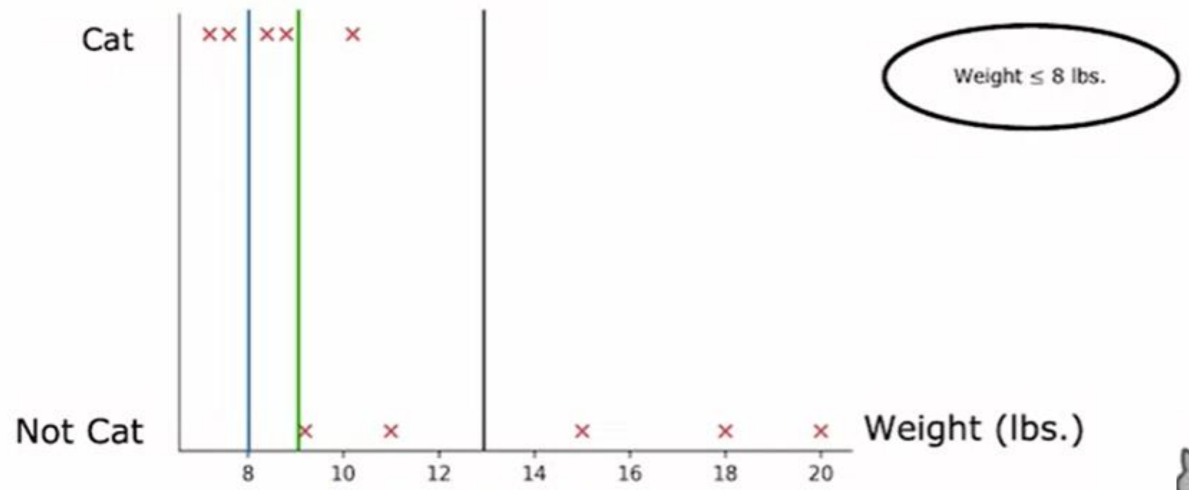
- Apply one- hot encoding

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

Continuous value 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

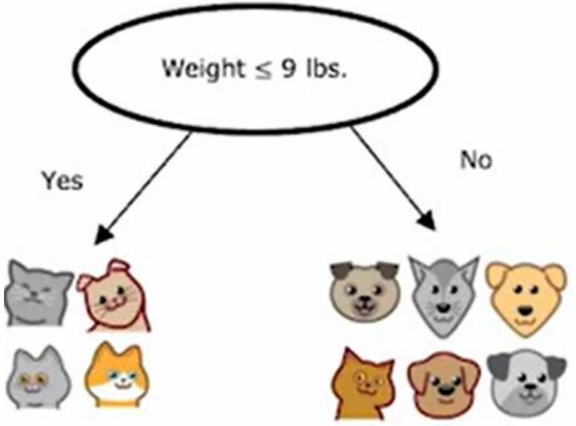
Continuous value features?



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

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

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



Decision Tree for classification from sklearn

```
class sklearn.tree.DecisionTreeClassifier(*, criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, class_weight=None, ccp_alpha=0.0)
```

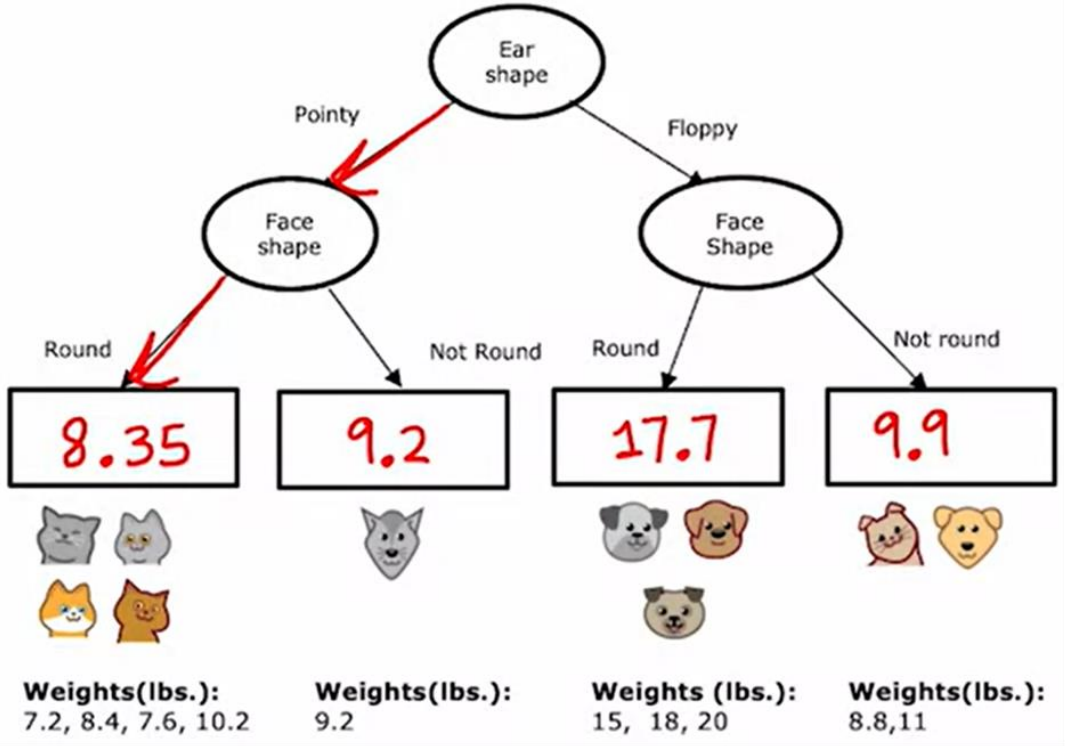
Decision Tree for Regression problem (Regression Tree)

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

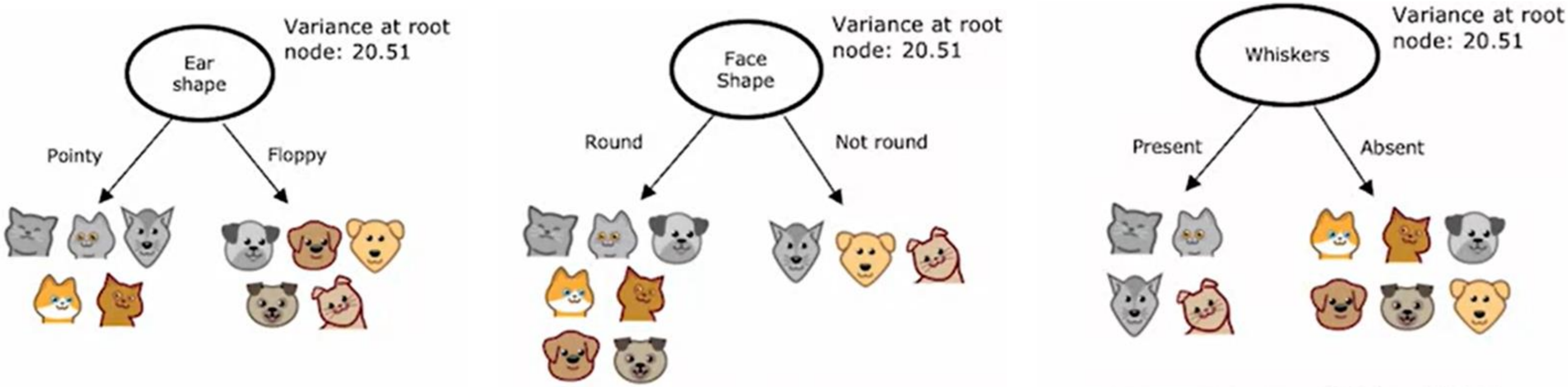
Decision Tree for Regression problem (Regression Tree)



New test example

Ear shape: Pointy
Face shape: Round
Whiskers: Present

Choosing a Split: Variance Reduction



Ear shape: Pointy	Ear shape: Floppy	Face Shape: Round	Face Shape: Not round	Whiskers: Present	Whiskers: Absent
Weights: 7.2, 9.2, 8.4,7.6, 10.2	Weights: 8.8, 15, 11, 18, 20	Weights: 7.2, 15, 8.4, 7.6,10.2, 18, 20	Weights: 8.8,9.2,11	Weights: 7.2, 8.8, 9.2, 8.4	Weights: 15, 7.6, 11, 10.2, 18, 20
Variance: 1.47	Variance: 21.87	Variance: 27.80	Variance: 1.37	Variance: 0.75	Variance: 23.32
$w^{\text{left}} = {}^{5}/_{10}$	$w^{\text{right}} = {}^{5}/_{10}$	$w^{\text{left}} = {}^{7}/_{10}$	$w^{\text{right}} = {}^{3}/_{10}$	$w^{\text{left}} = {}^{4}/_{10}$	$w^{\text{right}} = {}^{6}/_{10}$

$$20.51 - \left(\frac{5}{10} * 1.47 + \frac{5}{10} * 21.87\right) = 8.84$$

$$20.51 - \left(\frac{7}{10} * 27.80 + \frac{3}{10} * 1.37\right) = 0.64$$

$$20.51 - \left(\frac{4}{10} * 0.75 + \frac{6}{10} * 23.32\right) = 6.22$$

GYALPOZHING
COLLEGE OF INFORMATION TECHNOLOGY

Decision Tree for Regression problem (Regression Tree)

- Decision Tree can also be applied to regression problems, using the DecisionTreeRegressor class.

- The final prediction is the average of the value of the dependent variable in that leaf node.

```
class sklearn.tree.DecisionTreeRegressor(*, criterion='squared_error', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None, max_leaf_nodes=None, min_impurity_decrease=0.0, ccp_alpha=0.0)
```

More Example: https://www.saedsayad.com/decision_tree.htm

THANK YOU ☺