# Decision Tree Quiz

1. Decision tree algorithms: C4.5, ID3, CaRT

2. ID3 Algorithm
   split (node, {examples}):
   1. A ← the best attribute for splitting the {examples} ↗ e.g. windy, rainy
   2. Decision attribute for this node ← A
   3. For each value of A, create new child node
   4. Split training {examples} to child nodes
   5. For each child node / subset:
      if subset is pure: STOP
      else: split (child_node, {subset})

3. The DT algorithm builds a tree of nodes, using N training instances. If we prune the tree at level L, (i.e. L levels down from the root node, where L is zero for the root node), then the total no of training instances we find at the leaves (level L, and higher if a set at a node is pure) is:
   N. All the training instances are represented at every level, it is how they are divided up that changes.

4. When the DT algorithm is building the DT, it looks at a node and creates a number of child nodes by splitting on an attribute. A good algorithm tries to pick an attribute to split on where ...
   The purity of the child nodes is highest.

5. For the entropy measure we use to evaluate the purity of a split.
   - Entropy indicates the purity of the subgroup
   - Entropy indicates the randomness of the subgroup.
   - Entropy at a node can only ever decrease when we split.
   - The maximum entropy is 1, minimum 0.
   - Entropy is low when split is pure and high when it is impure.

6. The entropy of a node which has a proportion $p$ of ⊕ training instances and a proportion of $q$ ⊖ training instances is (so $(p+q)=1$):
   $$-p\log_2 p - q\log_2 q = -(p\log_2 p + q\log_2 q)$$

7. When computing the purity of a group of instances, entropy is measured in bits.

8. The formulation of the info. gain for node S on att. A is:

   $$Gain(S,A) = H(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} H(S_v)$$

   - $H(S)$: entropy of node S
   - $Gain(S,A)$: the decrease in entropy due to the split of S on att. A
   - $|S_v|$: the no. of instances in the child node corresponding to value v for att. A.
   - $|S|$: the no. of instances in node S
   - $\sum$: the weighted average of entropies of child nodes

9. We can use information gain to pick the attribute to split a node on. We pick the attribute with the highest IG. However, the information gain tends to pick the attributes w/ lots of values. To avoid this, we can use the gain ratio.

10. The DT alg (e.g. ID3) are prone to overfitting because:
    - if you run the algorithm to conclusion it will fit the training data perfectly
    - whenever you have an impure node, the algorithm will split it until all the leaves are pure, so you will have a perfect fit.
    - the training data is not representative of the whole population, so fitting it perfectly will bias your decision towards its peculiarities.

11. To avoid overfitting, do the following:
    1. Obtain a validation set of instances
    2. Evaluate tree on this set and remove the node (and its descendants) which when removed shows the greatest increase in performance on this set.
    3. Repeat until no further improvement happens no matter which node is removed.

    → subtree removal pruning procedure.

12. Att. w/ continuous value ~ included in DT by splitting using > or <
    Att. will be splitted multiple times to produce ranges.

13. DT can be used for multiclass regression!

14. A multiclass DT predicts the most frequent class the leaf node reached.
    * The multiclass problem is just like two classes, but the entropy is the sum over as many terms as there are classes.
      → instead over ⊕ or ⊖

15. When using a DT for a regr. problem, we can
    - use minimisation of variance in the subsets instead of max. of gain, in order to choose what you split on.
    - use the average of the training examples in the subset as the prediction.
    - perform lin. regr. on the training examples in the subset in order to build a linear model that predicts the value.

16. Random forest method
    - build many DTs, each one based on a different random subset of training data.
    - take a majority vote of the trees in the forest to get the prediction