Laboratory 2 – Decision trees

1. Supervised vs. unsupervised learning.
2. Classification problem: assigning a decision class label to a set of unclassified objects described by a fixed set of attributes (features).
3. Train and test paradigm.

   Creating a classifier is a multi-step approach. First, the classifier is generated from the given set of learning examples. Then, it is evaluated on the testing data. In the end it can be used for classification of new incoming examples.
4. Evaluation measures – based on confusion matrix:
   - $Classification\ accuracy = \frac{number\ of\ correctly\ classsififed\ examples}{number\ of\ all\ classified\ examples}$
   - $Sensitivity\ = \frac{TP}{TP+FN}$

     It measures the proportion of actual positives which are correctly identified as such.
   - $Specificity\ = \frac{TN}{TN+FP}$

     It measures the proportion of negatives which are correctly identified as such.

   The following confusion matrix is provided:

classified as ↓	cat	dog	rabbit
cat	29	7	0
dog	1	13	0
rabbit	0	0	10

   $$Classification\ accuracy\ = \frac{29+13+10}{29+1+7+13+10} = \frac{52}{60}$$

   To calculate sensitivity and specificity we need to choose a positive class (the rest are negative). Assume that in this case the class of interest is the cat.

classified as ↓	positive	negative
positive	29 (TP)	7 (FP)
negative	1 (FN)	23 (TN)

   $$Sensitivity = \frac{29}{29+1} = \frac{29}{30}$$

   $$Specificity = \frac{23}{23+7} = \frac{23}{30}$$
5. Experimental estimation of classification accuracy:
   - Hold-out: use two independent data sets: learning set (2/3) and testing set(1/3)
   - k-fold cross-validation: randomly divide set into k subsets; use k-1 subsets as learning set and 1 as a testing set; repeat until every subset was used for learning and testing (k-times)
   - Leave-one-out: similar to k-fold cross validation; size of the subset is set to 1

6. Decision tree is a directed graph, where nodes corresponds to some tests on attributes, a branch represents an outcome of the test and a leaf corresponds to a class label. Basic algorithm that induces a simple decision tree is a greedy algorithm that constructs the model in a top-down recursive divide-and-conquer manner.
7. Key issues in decision trees induction:
    - splitting criterion - splitting examples in the node / how to choose attribute / test for this node
    - stopping criterion - when should one stop growing the branch of the tree
    - pruning - avoiding overfitting of the tree and improving classification performance for the difficult data
8. ID3 algorithm informally:
    - Determine the attribute with the highest information gain on the training set (node or its subset in sub-nodes). Use this attribute as the root and create a branch for each of the values the attribute can have. Split training examples to branches depending on their attribute value.
    - For each branch (splitted subsets):
      If training examples are perfectly classified then stop and assign a class label to this leaf, else repeat the process with subset of the training set that is assigned to that branch.
9. How to calculate information gain:
    - $Information\_Gain = Entropy(S) - Entropy(S|attribute\_A)$
    - $Entropy(S) = \sum_{i \in class\ label\ in\ S}(-p_i \times log_2 p_i)$
    - $Entropy(S|attribute\_A) = \sum_{v \in values\ of\ attribute\_A}\left(\frac{|S_v|}{|S|} \times Entropy(S_v)\right)$
10. Attributes with a large number of values are very problematic (e.g. age, temperature). Subsets are more likely to be pure if there is a large number of values. Moreover, information gain is biased towards choosing attributes with a large number of values. This may result in overfitting (selection of an attribute that is non-optimal for prediction). There are two general solutions: discretization in a pre-processing step (transforming numeric values into ordinal ones by finding sub-intervals) or algorithms adaptation (binary tree or new splitting criterions).
11. How to find the best split:
    - Sort the values of splitting attribute.
    - Calculate the entropy between the points of different labels of decision class. Those are the potential optimal breakpoints.
    - Choose the split with minimum value of entropy measure.
    - Place split points halfway between the values.