INF283 - Exercise - 3

(**Exercise 3 deadline: 14th of september, 23.59**).
Submission details: Computer written or scanned pages.
Deliver here: https://mitt.uib.no/courses/12791/assignments

Weekly exercises are a compulsory part of the course. You will need to complete at least half of them. Weekly exercises give a total of 16 points to the final grade, of the total 8 exercises each then gives 2 points to you final grade, as long as you upload your answer to MittUIB.no/assignments before 23.59 on Fridays. They will then be reviewed, and points are added to your total grade score (If we see you have made an effort. So no score loss if your answer had some error in the calculation etc). If you have completed only a fraction of the tasks then you will get a fraction of 2 points.

If you follow these exercises and ask for help when you need it during the group sessions, it will help you a lot, especially through the more difficult parts of the course.

In this exercise we will prepare for project 1. We will look at model selection and evaluation (k-nn and logistic regression on iris), b) Pruning by hand on a simple artificial dataset, c) Decision trees.

Model selection and validation

Which machine learning algorithm should we choose ?
How to assess its quality ?

Model selection is the task of selecting a statistical model from a set of candidate models, given data. In the simplest cases, a pre-existing set of data is considered. However, the task can also involve the design of experiments such that the data collected is well-suited to the problem of model selection. Given candidate models of similar predictive or explanatory power, the simplest model is most likely to be the best choice (Occam's razor).

The two algorithms we have learned most about so far is KNN and regression.

Model validation is the process of deciding whether the results quantifying hypothesized relationships between variables, obtained from analysis (regression analysis), are acceptable as descriptions of the data.

Two common ways of validation are:

Hold out set:
It is either done by sampling an additional set of examples, independent of the training set, then we use the empirical error on this validation set as our estimator. Sometimes this is difficult, because we can not sample an additional dataset (i.e. no resources to fund it, or people are no longer alive etc.) If this is the case, the original data can be split into two sets.

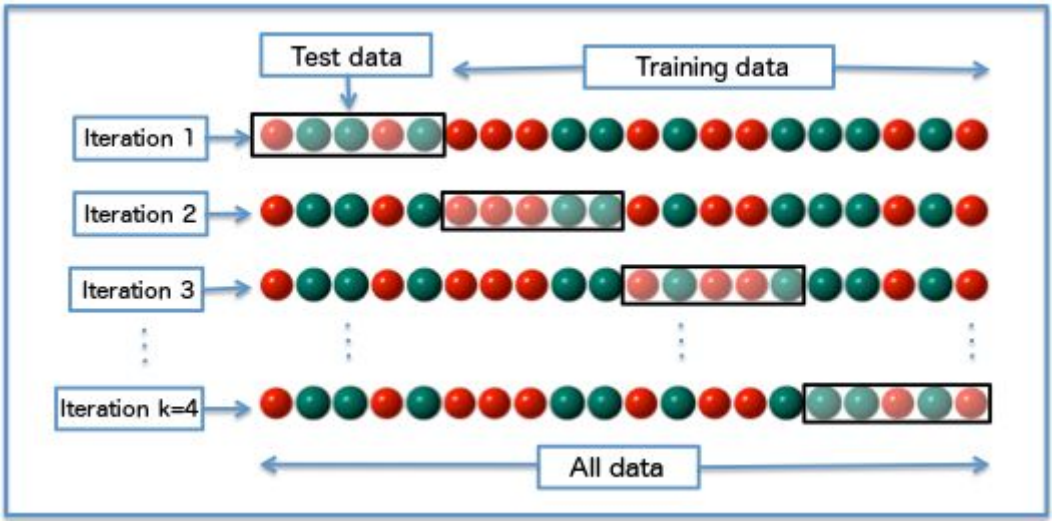

Figure 1. Cross validation iterations

Cross-validation:
The goal of cross-validation is to test the model's ability to predict new data that was not used in estimating it, in order to flag problems like overfitting and to give an insight on how the model will generalize to unseen examples.
One round of cross-validation involves partitioning a sample of data into complementary subsets (see figure 1), performing the analysis on one subset (called the *training set*), and validating the analysis on the other subset (called the *validation set*). To reduce variability, in most methods multiple rounds of cross-validation are performed using different partitions, and the validation results are combined (e.g. averaged) over the rounds to give an estimate of the model's predictive performance (estimate for the performance on unseen data points).
1.1:
Shortly describe difference between training set and validation set. Why do we need validation sets ?
1.2:
Use a programming language to select one of our two candidate algorithms (KNN and logistic regression) by cross validation on the IRIS data set, with 6 iteration splits.
Steps:

1. Choose your language
2. Import the IRIS data set and functions needed.
3. Create a function/script that outputs the **mean accuracy score** of the cross validation of the IRIS data-set on the two algorithms when tuning the hyperparameters (Set "K" for KNN and "C" Inverse of regularization strength for logistic regression.

4. From the 10 models, pick the best and evaluate its performance on the test set (accuracy)

NOTE: Try 5 meaningful values each for K and C, that gives you 10 models.
HINT for python:

```
>>> import numpy as np
>>> from sklearn.model_selection import train_test_split
>>> from sklearn import datasets
>>>from sklearn.model_selection import cross_val_score # here is cross val. in python
>>> # here import KNN and logistic regression

>>> iris = datasets.load_iris() # subset a part of this as test set for question 1.2.4
# clf =(KNN OR LOGISTIC algorithm)() # you need a for loop here for each hyperparameter
>>> scores = cross_val_score(clf, iris.data, iris.target, cv=?)
>>> scores # some array like this is output (but not the same)
array([ 0.96...,  1.  ...,  0.96...,  0.96...,  1.        ]) # Mean accuracy score is mean of these
0.96…
# write down which K and C gave the best mean accuracy score for KNN and logistic
#regression.
# pick the best of of the 10 models, which c or k did this have
```

HINT for R:

```
library(caret)
train_control <-  trainControl(method="cv", number=?) # cross val
model <- train() # function for training
```

Decision Tree learning

A **decision tree** is a decision support tool that uses a tree or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

Several decision trees algorithms exist, most famous are ID3 and CART. The first project in this course will be to create a structure like this from scratch, so we need some understanding of what is going on.

2.1:
Explain each of these words for decision trees (1-2 sentences each):
a node, a leaf, a root, a branch/split, entropy, gini index, information gain.
NOTE: Remember to define mathematically if you can, e.g. entropy.

2.2:
Now lets see if you understand what information means:

Entropy is measured in bits of information (you get 1 bit of information if you know the result of a fair coin flip).
How many bits of information do you get from drawing a hearts in a fresh card deck ?
p(hearts) = 13/52

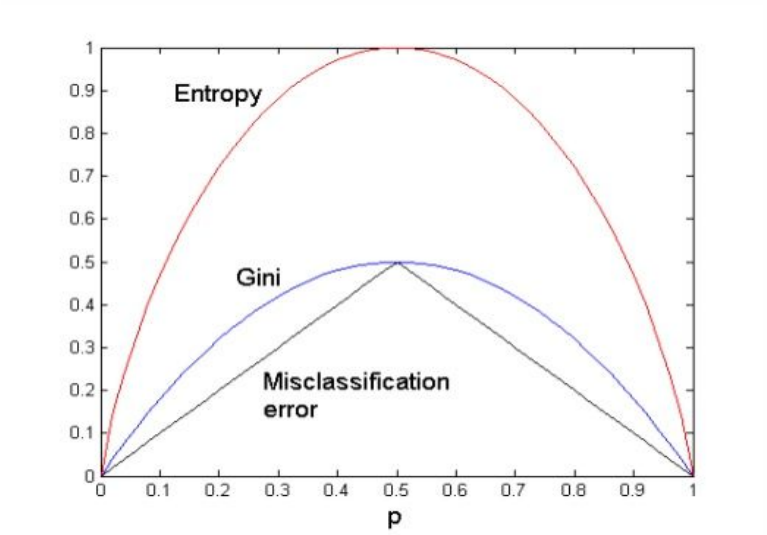

Figure 2: *The impurity measures Gini index and entropy, functions at different probabilities p.*

2.3
Given a solar system S, we split the objects in S {stars and planets(circles)} into two sets according to how far they are from solar systems center point (close and far away).
Calculate Information gain (IG) of this split when using impurity measure:
A: entropy (E): IG(solar system, distance) = E(solar system) - E(solar system, distance)
B: Gini index(G): IG(solar system, distance) = G(solar system) - G(solar system, distance)

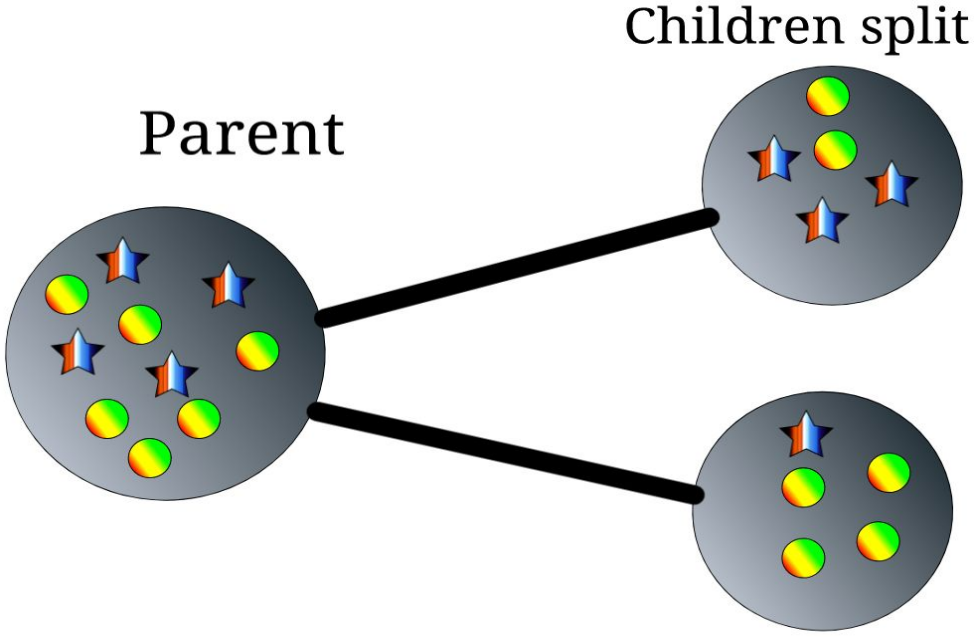

Figure 3

Pruning decision trees

Pruning is a technique that reduces the size of decision trees by removing sections of the tree that provide little power to classify instances. Pruning reduces the complexity of the final classifier, and hence can improve predictive accuracy by the reduction of overfitting.

We are going to focus on reduced-error pruning here:

- Consider each node for pruning. Start on leaves and move upwards.
- For a subtree S of the tree, if replacing S by a leaf does not make more prediction errors on the pruning set than the original tree, replace S by a leaf.

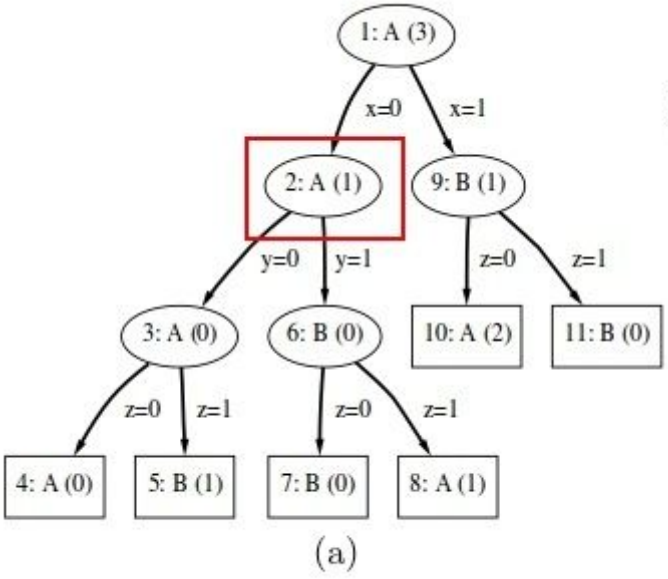

Figure 4: *Decision tree predicted on pruning data, red square is an example of node (node 2 split for majority class A, with 1 error on the pruning set A(1). B(2) means majority class B with 2 errors on the pruning set.*

We now have the tree with errors for pruning data. The pruning data looks something like in figure 5 (just an example, not needed for questions):
Majority class: The class with most items in the set. In figure 5 there are 3 B's and 2 A's, therefor the majority class is B in the pruning set. Node 1 in figure 4 have majority class A, which means there are more A's in the whole data set, since node 1 is the root node.

x	y	z	class
0	0	1	A
0	1	1	B
1	1	0	B
1	0	0	B
1	1	1	A

Figure 5: *Subsample of pruning set. Red square is an example of data of three categories x,y,z. You can see that for this case x = 0, y = 1, and z = 1, class is then B.*

2.1:
A: Write a pseudo code for this pruning
B: Simulate the pruning algorithm and show the steps. Describe which nodes are kept in this decision tree after minimum error pruning. (Either draw it or name the node integers kept after pruning)

Hint: See node 4 and 5 are children of 3, and the sum of errors for 4 and 5 are 1, while 3 have 0 errors. Since 1 error > 0 error, these children should be removed, etc.

Preparation to project-1 for non-programmers

This part is intended to help students who don't know the basic data-structures well. If you have done a few courses in informatics, you are most likely familiar with these topics.

The upcoming project will be a very free task, such that if you struggle with basic containers like a dictionary or classes, it might be hard to do. This is a informatics course, therefore we do programming.

If you feel like you need some help, we will be focusing to help you on the group sessions. Here are the things you should know how to do:

Create a class that depends on another class (child and parent etc)
Create a dictionary (linking id to value)
The method of recursion (method of solving a problem where the solution depends on solutions to smaller instances of the same problem)

You should try to make each of these structures in your programming language.