EXERCISE 1

1. Explain the ID3 algorithm for Decision Tree learning: **formally** describe input, output and the main steps of the algorithm.
2. Describe the concept of pruning in Decision Tree learning, highlighting the main motivation and an operational procedure to perform it.

In the ID3 the input is a dataset $D:\{(x_i, t_i)\}_{i=1}^{N}$ and the output a decision tree T that is the learned Function, the main steps are the Following:

- Choose an attribute (the one who maximize the information gain)
- For each value, create a node (root of the subtree)
- If in the reduced dataset, all samples are in class 1 (or 0), then create a leaf with 1 (or 0)
- Else, recursively run the algorithm on the reduced dataset.

Pruning is a procedure to reduce overfitting, it works as Follows:

- We have a tree T trained on D and a validation set $V : D \cap V = \emptyset$.
- do until term. condition
    - let n a subtree of T, picked randomly
    - let T' the tree where n is substituded with a leaf (most common value)
    - if $accuracy_V(T') > accuracy_V(T)$, then $T = T'$

At some point, the accuracy will stop decreasing.

EXERCISE 2

Consider a classification problem $f : X \times Y \times Z \to \{T, F\}$, with $X = \{a, b, c\}, Y = \{0, 1\}, Z = \{u, v, w\}$ and the data set in the table on the right

X	Y	Z	f
a	1	v	F
b	1	w	T
c	0	u	T
b	0	w	F
a	0	w	T
c	1	u	T

1. Describe a solution of this problem based on Naive Bayes. Which terms are computed in the Naive Bayes model and how?
2. Describe how to predict the class for the new sample $(a, 0, u)$ based on Naive Bayes (just provide the formula, no need to compute the numerical result).

In the naive bayes classificator we assume that the attributes are statistically independent from each other, so new points x are classified as follows:

$$c^* = \underset{c}{\operatorname{argmax}}\ \mathbb{P}(c|D) \prod_{i=1}^{d} \mathbb{P}(a_i | c, D)$$

$\uparrow$ attributes of the point

to predict $(a, 0, u)$ we consider the estimations:

$\mathbb{P}(T|D) = 4/6$, $\mathbb{P}(F|D) = 2/6$, in general $\hat{\mathbb{P}}(c|D) = \dfrac{|\{(x,t) \in D : t = c\}|}{|D|}$

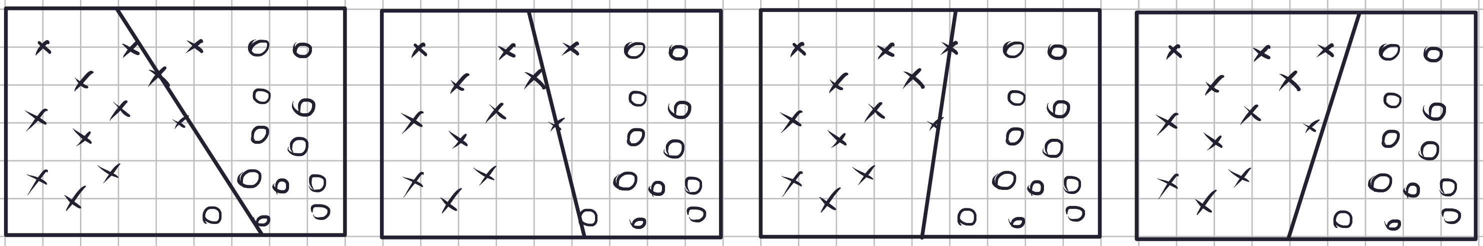
and For an attribute a_i:

$\hat{\mathbb{P}}(a_i | c, D) = \dfrac{|\{(x,t) \in D : t = c \wedge a_i \in x\}|}{|\{(x,t) \in D : t = c\}|}$. $\curvearrowleft$ this set migth be empty, like For $\mathbb{P}(u|F, D)$, so we add a small virtual probability (else, the total product would be zero)

EXERCISE 3

1. Describe the perceptron model for classification and its training rule.
2. Draw a graphical representation of a linearly separable 2D data set for binary classification and provide a qualitative graphical example of a possible evolution of perceptron training (4 images showing a possible temporal evolution of the solution of the algorithm on the sketched data set, with the last image showing a possible final solution).

Is a linear model in the parameters $w_0, w_1 \cdots w_d$ where a sample $x \in \mathbb{R}^d$ is classified as -1 or +1 as follows: $y(x) = \text{sign}(w^T x + w_0)$.
It's trained by considering $E(w) = \sum_{i=1}^{N} (t_i - w^T x + w_0)^2$ and applying the gradient descent $w \leftarrow w - \eta \nabla E$ until convergence (where $\eta \in \mathbb{R}^+$).



EXERCISE 4

Consider the following Convolutional Neural Network (CNN) acting on images of dimension 96 × 96 × 3:

conv1	Conv2D 3 × 3 kernel, 8 feature maps with padding 1 and stride 1
relu1	ReLU activation function
pool1	2 × 2 max pooling with stride 2
conv2	Conv2D 3 × 3 kernel and 5 feature maps with padding 1 and stride 1
relu2	ReLU activation function
conv3	Conv2D 3 × 3 kernel and 3 feature maps with padding 1 and stride 1
relu3	ReLU activation function
pool3	2 × 2 max pooling with stride 2
flatten	flatten operation
fc1	30 units
relu4	ReLU activation function
do1	dropout with rate 0.5
fc2	10 units
relu5	ReLU activation function
do2	dropout with rate 0.5
fc3	10 units
output	softmax

1. What kind of problem and which dimension of the problem it is able to solve?
2. Explain what is the role of the dropout layer in a CNN and in particular the meaning of the dropout rate.
3. Explain and motivate what is a suitable loss function to train this network.

From the input and the output, we see how this network classifies RGB images of 96×96 px in 10 different classes. the dropout layer is used to reduce overfitting, during training a portion specified by the rate is randomly picked and ignored. Since it's a classification problem, the loss function will be the cross entropy:

$$E(w) = -\sum_{i=1}^{N} t_i \cdot \ln \text{softmax}(\underset{\uparrow \text{net}}{f(x_i)})$$

EXERCISE 5

Given a data set $D = \{(x_n, t_n)_{n=1}^N\}$, denoted with its design matrix $\mathbf{X}$ and its output vector $\mathbf{t}$, for a classification problem $f : \Re^d \to C$, with $|C| = K$, and a linear model $y(x; w)$ with parameters w,

1. Define a regularized squared error function to solve the problem
2. Describe a kernelized linear model obtained by solving the above problem, and provide the solution of such a problem including a formal definition of the Gram matrix.
3. Provide the dimensions of all the elements of the solution

we consider the function $E(\underset{\substack{\uparrow \\ \text{matrix of} \\ \text{param.}}}{W}) = \frac{1}{2}\text{trace}\{(t - WX)^T(t - WX)\} + \lambda \|W\|$ where $\lambda \in \mathbb{R}^+$
and $\|W\| = \sum_i \sum_j |W_{ij}|$

The problem can be solved analytically and the resulting solution is $y(x) = \sum_{i=1}^{N} \alpha_i x_i^T x$ where α_i is a vector that depends on W and λ, we can substitute the dot product with:

$$y(x) = \sum_{i=1}^{N} \alpha_i K(x_i, x)$$ where K is a kernel function. The GRAM matrix is an $\underset{\substack{\uparrow \\ \text{number of samples}}}{N \times N}$ matrix: $K = \begin{pmatrix} x_1^T x_1 & \cdots & x_1^T x_N \\ \vdots & \ddots & \vdots \\ x_N^T x_1 & \cdots & x_N^T x_N \end{pmatrix}$. α_i is a $\overset{\text{num classes}}{\overset{\downarrow}{K}}$ dim. vector and there are N of them.

EXERCISE 6

Given a data set D for binary classification and a set of learners $y_m(x)$, $m = 1, \dots, M$,

1. Describe the general approach of AdaBoost and provide details on its sequential training.
2. Provide the final classifier that combines the given learners $y_m(x)$.

In the general approach the classifiers are trained sequentially on a set of weigthed samples, the sample on which y_t fails will have a greater weigth for y^{t+1}. These weigths are influenced by a coefficient α_i that measures the "goodnes" of each model, a new point are classified by:

$$y(x) = \underset{C}{\arg\max} \sum_{i=1}^{M} \alpha_i \cdot \delta(y_i(x) = c)$$