Name and surname: _____

Degree and Group: _____

NIA and Signature: _____

1. Data download

Each student must work with a different data file that can be downloaded from the following url (do not forget to replace your student number when copying the url in your browser):

```
http://www.tsc.uc3m.es/~jarenas/TMDE1617_E1/YourNIA.zip
```

You will find in the compressed file two .mat files for the two parts of the exam.
If you are using python, the following code lines will allow you to read the data in a .mat file:

```
>>> import scipy.io
>>> data = scipy.io.loadmat('file_to_load.mat')
>>> xTrain = data['xTrain']
>>> sTrain = data['sTrain']
>>> ...
```

2. Regression

You are given a dataset with K labeled D-dimensional observations $\{(\mathbf{x}^{(k)}, s^{(k)})\}, k = 1, \ldots, K\}$. The observations are stored in matrix `xTrain` and vector `sTrain`, in such a way that the k-th row contains $\mathbf{x}^{(k)} = (x_1^{(k)}, \ldots, x_D^{(k)})^\top$. Vector `sTrain` contains the target variable (the k-th element of `sTrain` is $s^{(k)}$).

1. Compute the sample variance for each component of the training data in `xTrain`. Save the result in vector variable `vTrain`.

2. Normalize the training data in such a way that all components have average value 0 and sample variance 1. Apply the same normalization to the test data in `xTest`. Save the result in matrices `xnTrain` and `xnTest`, respectively. In the following, use the normalized datasets.

3. Compute the least squares regression coefficient for the estimator in the form

$$\hat{s} = \mathbf{w}_e^T \mathbf{x}_e.$$

where $\mathbf{w}_e = (w_0, w_1, \ldots, w_{D-1})^\top$ and $\mathbf{x}_e = (1, \mathbf{x}^\top)^\top$. Save the result in variable `we`.

4. Compute the least squares regression coefficients for the estimator in the form

$$\hat{s} = w_0 + \sum_{i=1}^{D} w_i x_i^i \tag{1}$$

and save the coefficient vector in variable `w`

5. For the obtained estimator, commpute the average absolute error over the training datase and save the result in variable `EAP`. The average absolute error is defined as

$$EAP = \frac{1}{K} \sum_{k=1}^{K} |\hat{s}^{(k)} - s^{(k)}|$$

3. Classification

Consider a binary classification problem characterized by Gaussian likelihoods:

$$p_{\mathbf{X}|H}(\mathbf{x}|0) \sim \mathcal{N}(m_0\mathbf{1}, 2\mathbf{I})$$

$$p_{\mathbf{X}|H}(\mathbf{x}|1) \sim \mathcal{N}(m_1\mathbf{1}, 2\mathbf{I})$$

where $\mathbf{x} = (x_1, \ldots, x_D)^\top$ is a D dimensional vector of observations, $\mathbf{1}$ is an all-ones vector, and $\mathbf{I}$ is the identity matrix.

In the provided data file you can find matrices `Xtrain` and `Xtest` containing observation vectors stored in a row-wise manner, and variable `ytrain` that contains the true hypothesis corresponding to each element of `Xtrain`, i.e., $y^{(k)}$ is the true hypothesis responsible for the generation of the k-th row of `Xtrain`.

Consider for this exercise classifiers that implement the following decision criterion:

$$t \underset{D=0}{\overset{D=1}{\gtrless}} \eta,$$

where η is a threshold for the classification and $t = \sum_{m=1}^{d} x_m$, with $d \leq D$. For instance, if $d = 2$, t is the sum of variables x_1 and x_2.

You have to use data in this section as they are provided. Do not carry out any normalization processes.

1. Estimate m_0 and m_1 as the average of the available training data belonging to classes 0 and 1. Save the result for class 0 in variable `m0`.

2. Obtain the threshold η of a Neyman-Pearson classifier based just on observation x_1 satisfying $P_{\text{FA}} \leq 0{,}1$. Save your result in variable `etaNPx1`.

3. Compute the probability of detection that would be achieved by the classifier designed in the previous section. Save your result in variable `PDx1`.

4. Compute the mean of and variance of T, conditioned on $H = 1$, for $d = 1, 2, \ldots, D$. Save your results in variables `tm` and `tv`. I.e., both variables should represent length-D vectors, and the i-th element of `tm` and `tv` should contain the mean and variance of T (under hypothesis 1) when T is the sum of the first i variables.

5. Obtain the threshold of the ML classifier for $d = 3$, and compute the classifier predictions for the data in `Xtest`. Save your results in variable `ytest`.

4. Saving and uploading results

Save (at least) the variables mentioned in the exercises in a file called `results.mat`. The following matlab command performs this task for you:

```
save('results.mat', 'mTrain', 'xnTrain', 'xnTest', 'we', 'w', 'RECP', 'm0', ...
    'etaMLx1', 'PFAx1', 'tm', 'tv', 'etaNPd3')
```

If you are using python, use instead:

```
>>> scipy.io.savemat('results.mat', {'mTrain': mTrain, 'xnTrain': xnTrain, ... })
```

Zip file `results.mat` together with your code in a file called `Lab12.zip`, and upload the .zip file to Aula Global before the deadline expires.