



# Lab2: Decision trees, support vector machines & neural networks.

Ana Mestre Borges

Machine Learning course

Master in Intelligent Interactive Systems

## The dataset

[https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+\(original\)](https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(original))

- Source: UCI / Wisconsin Breast Cancer
- Number of classes: 2
- Number of data: 683
- Number of features: 10

### Attribute Information:

1. Sample code number: id number
2. Clump Thickness: 1 - 10
3. Uniformity of Cell Size: 1 - 10
4. Uniformity of Cell Shape: 1 - 10
5. Marginal Adhesion: 1 - 10
6. Single Epithelial Cell Size: 1 - 10
7. Bare Nuclei: 1 - 10
8. Bland Chromatin: 1 - 10
9. Normal Nucleoli: 1 - 10
10. Mitoses: 1 - 10
11. Class: (2 for benign, 4 for malignant)

## Decision trees

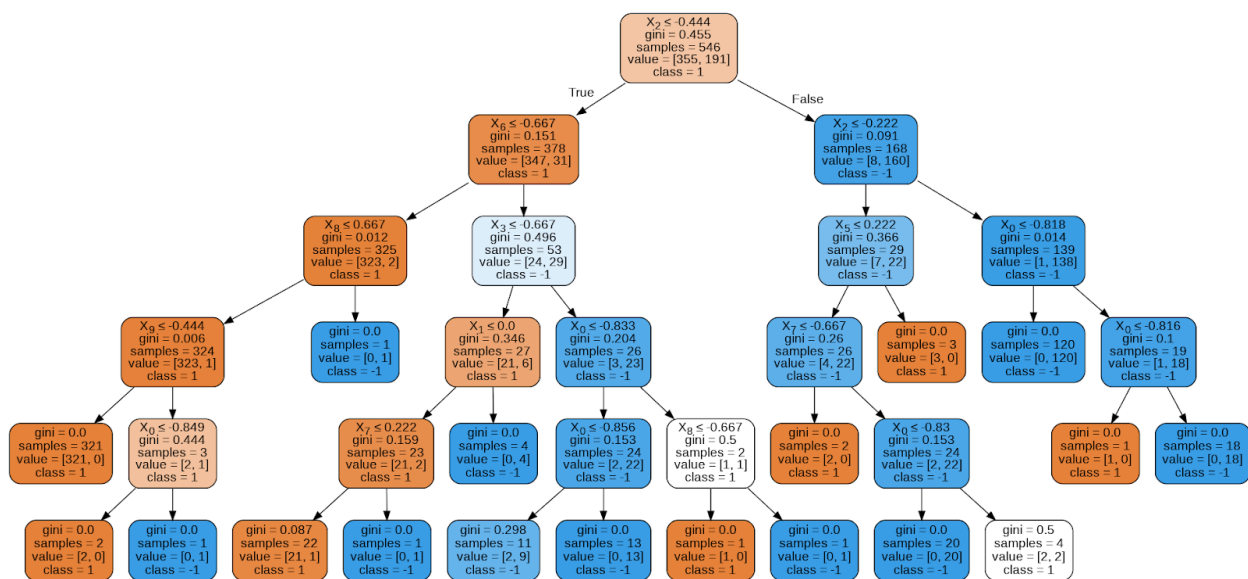
- Partition the dataset into training and testing sets.
- Run a decision tree learning algorithm using the training set.
- Test the decision tree on the testing dataset and report the total classification error (i.e. 0 = 1 error).
- Repeat the experiment with a different partition.
- Plot the resulting trees.
- Are they very similar, or very different? Explain why.

*Advice: it can be convenient to set a maximum depth for the tree.*

## Results

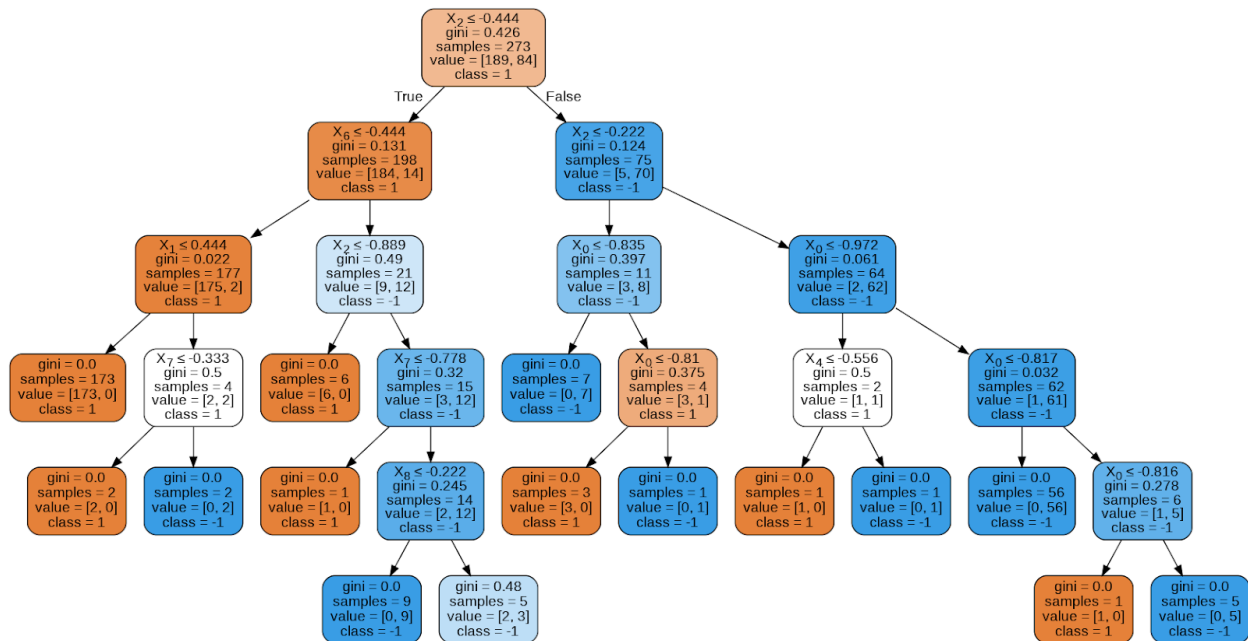
Generation of a decision tree using 20% of the dataset for testing and 80% for training. There has been defined a max depth of 5.

Accuracy: 0.92



Generation of a decision tree using 60% of the dataset for testing and 40% for training. There has been defined a max depth of 5.

Accuracy: 0.94



**Are they very similar, or very different? Explain why.**

Regarding to how the trees obtained by being trained with different datasets (same data but splitted differently and also same max tree depth), we can see they are different. This can be observed by looking at both the plots and the accuracy score obtained for each of them. The changes between them are produced because decision trees are quite sensitive to data changes.

## Support Vector Machines

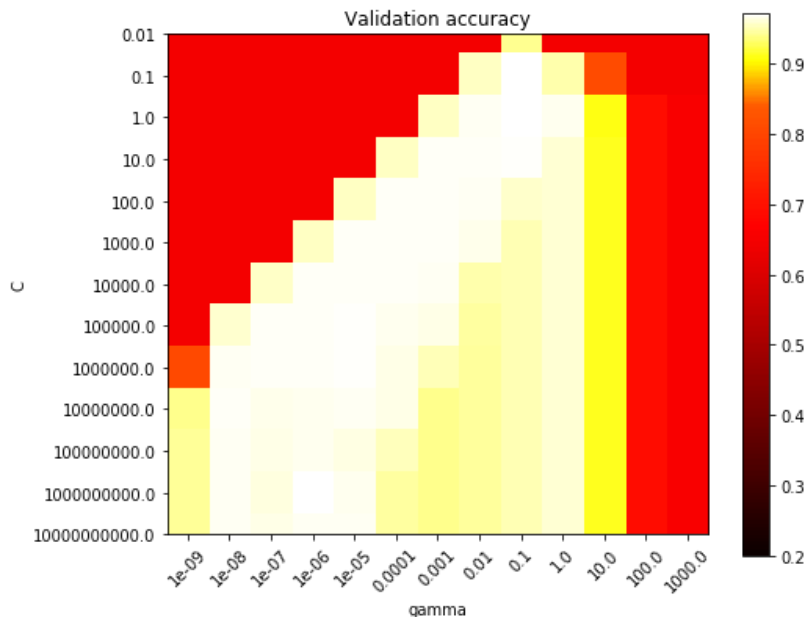
- Run SVM to train a classifier, using radial basis as kernel function.
- Apply cross-validation to evaluate different combinations of values of the model hyper-parameters (box constraint C and kernel parameter  $\gamma$ ).
- How sensitive is the cross-validation error to changes in C and  $\gamma$ ?
- Choose the combination of C and  $\gamma$  that minimizes the cross-validation error, train the SVM on the entire dataset and report the total classification error.

*Advice: use a logarithmic range*

## Results

### How sensitive is the cross-validation error to changes in C and Gamma?

It is quite sensitive. By looking at the following chart we can see that, on the one hand when the gamma parameter gets smaller, the model cannot capture the complexity of the data. But on the other hand, when it gets larger, the C parameter has no effect on the overfitting



The best parameters found to minimize the cross-validation error are:

$C = 0.695$ ,  $\gamma = 0.162$  with score = 0.9726

The accuracy for the entire dataset is: 0.9736.

## Neural Networks

- Train a Multi-Layer perceptron using the cross-entropy loss with l-2 regularization (weight decay penalty). In other words, the activation function equals the logistic function.
- Plot curves of the training and validation error as a function of the penalty strength  $\alpha$ .
- How do the curves behave? Explain why.
- Experiment with different sizes of the training/validation sets and different model parameters (network layers).

*Advice: use a logarithmic range for hyper-parameter  $\alpha$ .*

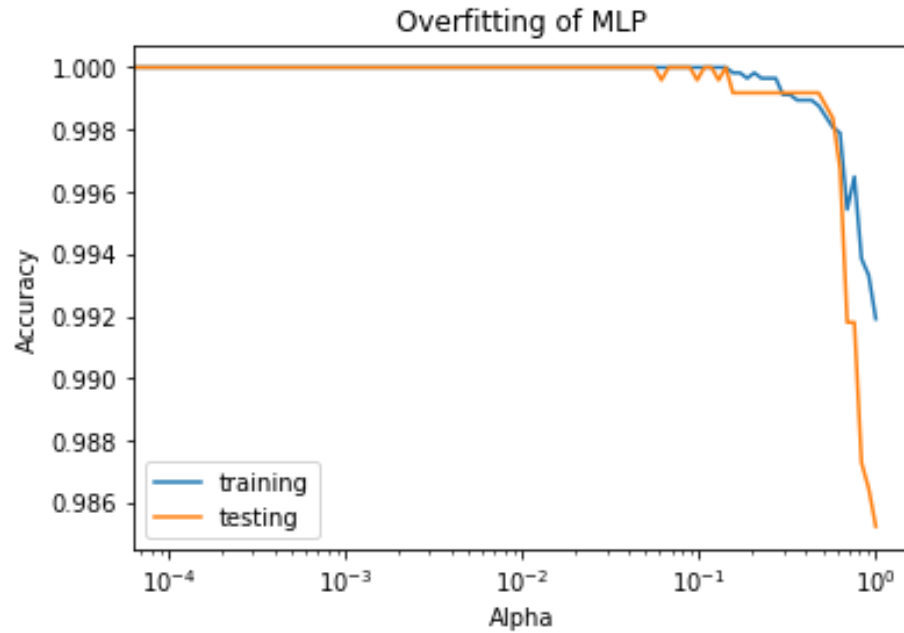
## Results

Regarding the neural networks experiment, I couldn't achieve satisfactory results with the breast cancer dataset, I assume that maybe the problem was that this dataset was too small to see the results we were looking for. Therefore, I used the mushroom dataset.

Mushroom dataset information:

- # of classes: 2
- # of data: 8124
- # of features: 112

Once the new dataset was selected, some experiments had to be done by changing the model parameters and trying to figure out how the  $\alpha$  values affected our model. However, here are the final results:



It can be seen that the accuracy for both models (training and testing) are high until the alpha becomes a higher value. At this point the accuracy falls significantly. This behaviour change happens because the overfitting has been reduced.