

Classification

Jordi Villà i Freixa

Universitat de Vic - Universitat Central de Catalunya
Study Abroad

jordi.villa@uvic.cat

course 2023-2024

Preliminary note

The material in these slides is strongly based on [?]. When other materials are used, they are cited accordingly.

Mathematical notation follows as good as it can a [good practices proposal](#) from the Beijing Academy of Artificial Intelligence.

What to expect?

In this session we will discuss:

- Classification methods
- Zero-one loss
- Bayes error rate
- Classification metrics

Regression is a supervised learning method

Supervised methods in which a categorical response variable Y takes one of the possible c values which is to be predicted from a vector of \mathbf{X} explanatory variables, using a prediction function g .

As g classifies the input \mathbf{X} into one of the classes, we call g a classification function or, simply, a *classifier*.

As with any supervised learning technique, the goal is to minimize the expected loss or risk

$$\ell(g) = \mathbb{E}\text{Loss}(Y, g(\mathbf{X})) \quad (1)$$

for some loss function $\text{Loss}(Y, g(\mathbf{X}))$ that quantifies the impact of classifying a response y with $\hat{y} = g(\mathbf{x})$.

Zero-one loss

The zero-one or *indicator* loss function is the natural choice:
 $\text{Loss}(y, \hat{y}) := \mathbb{I}\{y \neq \hat{y}\}$: this is: there is no unit loss for a correct classification and a unit loss for wrong one.

This leads to the fact that we aim at taking $g(\mathbf{x})$ to be equal to the class label y for which $\mathbb{P}[Y = y | \mathbf{X} = \mathbf{x}]$ is maximal.

The error we generate in this process is linked to the so-called **Bayes error rate**.

Pre-classifier

For a given training set τ , a classifier is often derived from a pre-classifier g_τ , which is a prediction function (learner) that can take any real value, rather than only values in the set of class labels.

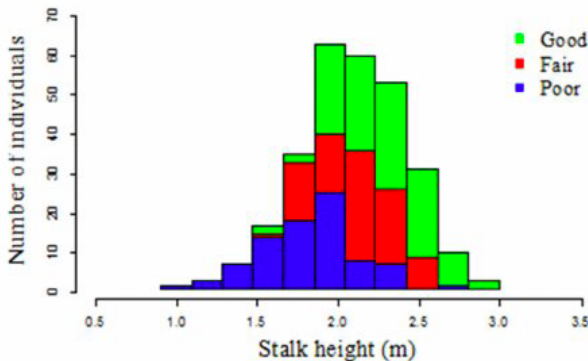


Figure 1: Adapted from [here](#)

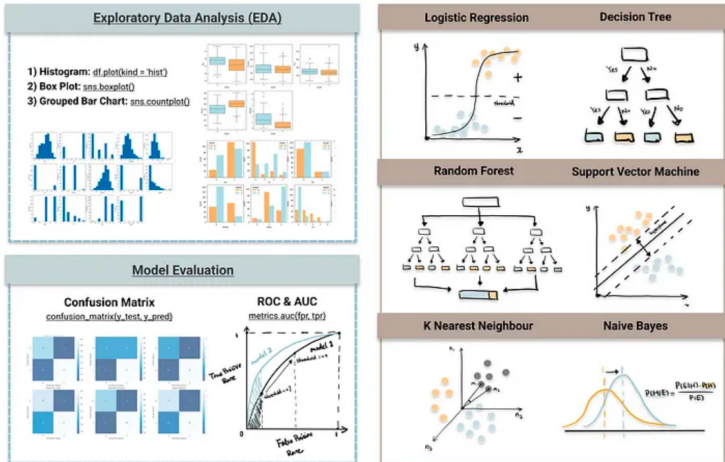


Figure 2: Check the [source](#) for a plain explanation of the different classification methods.

Training and test sets. Loss and confusion matrices

Theoretically, we should be measuring the risk in Eq. ?? and minimizing such equation over some class of functions \mathcal{G} . However, as the training loss is often a poor estimate of the risk, this is usually estimated from the test set τ' .

Loss matrix \mathbf{L} : for the indicator loss function, it contains 0 in the diagonal and 1 everywhere else.

Confusion matrix \mathbf{M} : counts the number of times that, for the training or test data, the actual (observed) class is i whereas the predicted class is j .

The training/test loss of the classifier in terms of \mathbf{L} and \mathbf{M} is $\frac{1}{n} \sum_{i,j} [\mathbf{L} \odot \mathbf{M}]_{ij}$. In the case of the indicator loss, the missclassification error is $1 - \text{tr}(\mathbf{M})/n$.

Confusion matrix

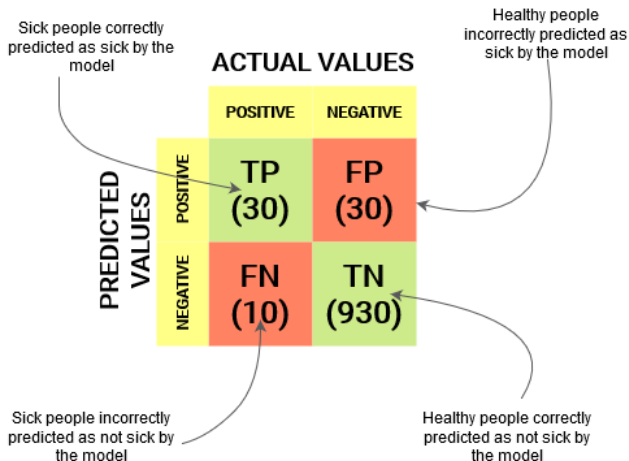


Figure 3: Adapted from [here](#).

Missclassification error and accuracy

In the binary classification case ($c = 2$), and using the indicator loss function, the missclassification error can be written as:

$$\text{error}_j = \frac{\text{fp}_j + \text{fn}_j}{n}$$

and the accuracy can be calculated by measuring the fraction of correctly classified objects:

$$\text{accuracy}_j = 1 - \text{error}_j = \frac{\text{tp}_j + \text{tn}_j}{n}$$

We can do better than this in many situations:

- we can modify the loss matrix and make it different from the indicator
- we can modify the the way we measure the classification beyond the accuracy

- precision: $\text{precision}_j = \frac{tp_j}{tp_j + fp_j}$

- recall or sensitivity: $\text{recall}_j = \frac{tp_j}{tp_j + fn_j}$

- specificity: $\text{specificity}_j = \frac{tn_j}{tp_j + fp_j}$

- F_β score: $F_\beta = \frac{(\beta^2 + 1)tp_j}{(\beta^2 + 1)tp_j + \beta^2 fn_j + fp_j}$