1 Cross Validation

Let's assume to be in one of the following situations:

- the prediction algorithm of choice has a variety of parameters that need to be tuned, which means optimised in order to improve accuracy (or any other measure of goodness)
- performances of a group of completely different models need to be assessed and ranked for selection purposes

Cross validation is a widely (if not the most) used class of methods to assign to each different model an estimate of their "goodness". A great asset of cross validation is that it offers the flexibility to choose the most appropriate measure of goodness, depending on the situation. Such measures can be the very intuitive accuracy, the ever popular AUC or, as in this paper's case study, a customised function depending on a loss matrix (in this case Table **??**).
The class of C.V. methods contains, among others:

- K-fold cross validation
- leave-one-out cross validation (limit case of K-fold for $K := \#\{training\ \ set\}$)
- stratified K-fold C.V.
- N-repeated stratified K-fold C.V.
- Monte Carlo C.V.
- Generalised CV

The method that was chosen for this study is a 6-repeated stratified 3-fold C.V.

1.1 N-repeated stratified K-fold C.V.

Suppose a training set T is given so that every $(v_i, y_i) \in T$ is of the form $(v_{1,i}, .., v_{u,i}, y_{1,i}, .., y_{v,i})$ where u is the number of explanatory variables, v is the number of outcomes.

Definition 1.1. *A subset $T_k \subset T$ is called a stratified K-fold of T if*

1. *their cardinalities follow relation*

$$\#T_k = \left\lfloor \frac{\#T}{K} \right\rfloor$$

2. *is picked randomly from the family of subsets of* T *such that*

$$\mathbb{E}_{T_k}[y] = \mathbb{E}_T[y]$$

First of all $\forall k \in 1, .., K$ a stratified K-fold T_k of T is taken. A model prediction will be called $\hat{M}(\cdot, \theta)$ for simplicity and in order to stress its dependence from the parameter θ to be tuned. Note that $\hat{M}(\cdot, \theta)$ could be interpreted as a total different model prediction $\hat{M}_\theta$ as well.
The second step consists in training the model on $T \backslash T_k$ $\forall \theta_i \in \Theta$ the parameter candidates set (from now on we use the simplified notation $\hat{M}_k(\cdot, \theta) := \hat{M}_{T \backslash T_k}(\cdot, \theta)$). Finally $\forall (v_j, y_j \in T_k$

$$L(\hat{M}_k(v_j, \theta_i), y_j)$$

is calculated and the results summarised in the following way (C.f. **?**)

$$CV(\theta) := \frac{1}{\#T} \sum_{k=1}^{K} \sum_{j \in C_k} L(\hat{M}_k(v_j, \theta), y_j)$$

For minimisation and maximisation reasons the multiplication by $\frac{1}{\#T}$ is irrelevant so it can be omitted.