

# Article CREPS Anglais

GRENOUILLAT Alexia, LELIEVRE Hugo, AFIFI Mohamed Yasser

April 2022

## 1 Abstract

Data from the CREPS of Toulouse shows that nearly 80% of the athletes who enter a high-level athletic programme injure themselves during the first year. This impedes training, and consequently leads to a decline in performance. This study presents statistical tools to enhance athletic performance by predicting athletes' injuries. We work hand-in-hand with the CREPS of Toulouse, a public training centre for high-level athletes. We base our models on the data provided by their test protocol, called Mobility and Stability Athlete Testing (TM2S). Such test protocols already described in the literature, as for instance the Global Mobility Condition (GMC). The aim of this study is to build our own model based on the results of this test. We want to predict how likely an athlete is to be injured, depending on variables such as the score of each exercise on the TM2S, the sport practiced and if the athlete is new in the high-level program. We start by examining dependencies between significant variables to determine which ones are important for our prediction. We then build the following models to explain the qualitative variable "Injury" by the variables previously described: logistical regression, random forest, regression trees, naive Bayes and clusters of variables. Thanks to these models, we are able to compute the probability of injury for each athlete. We create a R interface; the coaches enter the athlete's characteristics, and the interface gives them the personalized injury risks. Nevertheless, some limits need to be underlined: the AUC (a performance indicator) of our models never reaches more than 0.8, which can be explained by the poor size of our dataset.

**Keywords:** Sport performance enhancement, Injury prediction, Machine learning algorithms, Logistic regression, LASSO penalisation, Chi-squared tests, Clustering of variables, Shiny interface.

## 2 Introduction

When athletes enter a high-level structure, injuries can be a crippling obstacle to their improvement. Since injuries can make them unavailable for a long period of time, their sport performance is significantly impacted. Therefore, preventing such injuries can considerably enhance the overall performance of the athletes. The CREPS of Toulouse, a public athletic training establishment, has noticed that 85% of their high-level athletes are injured during their first year in the structure. In order to reduce the number of injuries, fitness coaches of the CREPS are looking for a way to detect factors which can increase significantly the risk of injury. One popular procedure of performance evaluation that has been used for years is the Functional Movement Screen (FMS, 1997), a worldwide scientifically recognized test. It is a set of seven physical exercises of coordination and strength. Athletes are tested on extreme mobility and stability positions and assessed on a scale from 0 to 3 points, depending on the completion of these exercises. The FMS is now in widespread use and scientifically recognized around the world. However, it does not completely satisfy the needs of the CREPS. Even if the FMS has a complex evaluation for each exercise, there is only one threshold to assess the ability of the athlete. This threshold is not precise enough if we try to prevent athletes' from injury. Another procedure, named "Global Mobility Condition"

(GMC, 2011), have been created by doctors and physiotherapists especially for the preparation of French rugby men for the World Cup. The GMC is a set of twenty physical binary-scored tests, divided into four categories : lower and upper limb flexibility, strength and functional tests. Contrary to the FMS, the GMC provides five thresholds for the global grade, which are listed in Figure 1.

GMC score	Patient's condition	Diagnosis
$GMC \geq 19$	Excellent	To maintain
$17 \leq GMC < 19$	Normal	To optimise
$15 \leq GMC < 17$	Insufficient	To improve
$10 \leq GMC < 15$	Significant risk of injuries	To work

Table 1: Global Mobility Condition (GMC) rating scale (<https://www.global-mobility-condition.fr/>)

These thresholds are useful for fitness coaches to adapt their corrective routines based on the athlete's global grade. Nevertheless, this classification has not been validated in the literature, so CREPS' coaches cannot use the GMC to prevent injuries efficiently. Due to the lack of precision of these methods, the CREPS of Toulouse conceived its own test, named TM2S ("Athletes' Mobility and Stability Testing"). It is inspired by the FMS and the GMC: it includes 13 exercises, has a binary notation and is more focused on injury detection than performance evaluation. Since the TM2S is more adapted to the CREPS' needs, we used it to build a statistical model which can predict athlete's injuries efficiently. First, we have to analyse dependencies between our variables to determine those which can be useful for our model. Our second task is to perform a selection of variables to reduce the problem's dimension and get a model more interpretable. Once these steps achieved, we have to build several algorithms to find the best statistical model to predict athlete's injuries. To compare the different methods, we use performance indicators such as the AUC and the rate of correct predictions. Finally, we create an user interface with the R software so the CREPS' fitness coaches can add new data and visualize the probability of injury in real time. We conclude by explaining the main limitations of the project and giving some further improvements. The paper is organised as follow :

- **Section 2:** Materials of the CREPS project
- **Section 3:** Statistical tools and machine learning algorithms
- **Section 4:** Main results with each statistical model and selection of the best one
- **Section 5:** Presentation of the R interface
- **Section 6:** Conclusion and discussion

## 3 Materials of the CREPS project

### 3.1 Definition of the term 'injury'

Before starting a statistical analysis, we first have to define clearly the notion of injury. Since our entire study depends on this variable, its definition is crucial. With the collaboration of the CREPS coaches, we define the term 'injury' as follow: "Injury results in downtime and inability to practice". This is consistent with the desire of performance, since even a week of downtime prevents an athlete from training. Because of the lack of data and the significant bias present in our dataset, we will not study the precise number of injuries each athlete had in a season. In the following sections, we will only take into account the fact that an athlete has been injured during the sport season. Moreover, due to the difficulty of prediction, we will not deal with the downtime of an injury.

### 3.2 The Athlete’s Mobility and Stability Test (TM2S in French)

It is a scored protocol test created by the CREPS of Toulouse to reduce the risk of injuries among the athletes and their downtime. Twenty exercises compose the TM2S, which are split into four categories :

- Lower mobility, scored on 8 points;
- Upper mobility, scored on 4 points;
- Lower stability, scored on 6 points;
- Upper stability, scored on 2 points.

The TM2S is binary scored, and points are attributed by the fitness coach depending on the athlete’s achievement of the exercises. If the athlete poorly performs an exercise, or if he cannot complete it, the coach gives a score of zero for this exercise. If the athlete performs it correctly (or at least with minor compensation), the coach gives a score of one. Depending on the total score, the CREPS sets up corrective routines to improve athlete’s weaknesses. The main aim of these routines is to prevent athletes from injuries by optimising their mobility and stability. The score sheet and the TM2S exercises are available in Appendix - Figure 13.

### 3.3 Presentation of the dataset

We consider a sample composed of athletes from eight sports : Baseball, Basketball, Bowling, French Boxing, Rowing, Rugby, Swimming and Synchronised Swimming. We use the datasets provided by the fitness coaches of the CREPS, which gather information for the 2020-2021 season. The sample is composed of 119 individuals; each of them is a CREPS’ athlete, practice one of the eight sports listed above and had completed the TM2S test at the beginning of the 2020-2021 season. For each athlete, the score per exercise, per category and the global score are given. We choose to consider the score per category as variable of our statistical models instead of the score per exercise. Even if the last one seems to be more precise, it leads to an excessive number of variables. Therefore, it is a better strategy to condense the information given by the test before building a statistical model.

To combine information on both TM2S scores and athletes’ injuries, we have to join two datasets on the anonymity number given to each athlete. Since the size of our final dataset is very small, we cannot take into account the detail of the injuries’ description and its precise localisation. Therefore, we have to classify these injuries and group them into three different classes:

- Lower Limbs injuries (LL), which groups the entire area extending from the top of the thigh to the end of the feet;
- Upper Limbs injuries (UL), which groups all injuries located at the level of the arm (from the collarbone to the fingertips);
- Torso and Spine injuries (TS), regarding torso, bust, lower back and spine injuries.

This classification is validated by the CREPS coaches, Marine and Thomas. Then, we consider two types of response variables, which are both binary:

- A response variable which take in account all types of injuries (*Ex: “Yes” if the athlete was injured one or more times during the season, “No” otherwise*);
- A more targeted response variable at one of the three parts of the body explained previously (*Ex: “Yes” if the athlete has injured his spine at least once during the season, “No” otherwise*).

Once the response variable is clearly defined, we need to identify the explanatory variables, *ie* the variables of interest that are more or less related to the risk of injury. Indeed, if our response variable is independent of one/several of our explanatory variables, they are not useful to write our model. We can consider the following variables:

- TM2S scores (by category);
- The sport practiced;
- The fact that an athlete is new / old in the structure;
- The athlete's past (*ie* if he had injuries in the past which could favor a relapse);
- Whether an athlete is currently experiencing pain in a particular body area.

Now that we have the basic mathematical tools, we can start by a statistical analysis to build our models.

## 4 Statistical models and machine learning algorithms

### 4.1 Chi-squared test of independence

In order to determine if the variables previously described are related to athlete injuries, we use the Chi-square test of independence. According to Minhaz Fahim Zibran, "*Chi-square ( $\chi^2$ ) test is a nonparametric statistical analyzing method often used in experimental work where the data consist in frequencies or 'counts' [...] as distinct from quantitative data obtained from measurement of continuous variables [...]. The most common use of the test is to assess the probability of association or independence of facts*" (1). We need to test the independence between:

- The number of athlete's injuries in a year and the sport played;
- The sport practiced and the areas of the body most frequently injured;
- The fact that an athlete has been injured during the year and his status (new / old in the structure);

For the LL, UL and Spine body parts, we also tested the independence between:

- The number of injuries and the past of the athletes (*Ex (LL): Is an athlete's number of LL injuries during the year independent of the number of past LL injuries ?*)
- The number of injuries and the presence of current pain. (*Ex (LL) : Is an athlete's number of LL injuries over the year independent of the presence of a current LL pain ?*)

For each test of independence, we look at the p-value obtained at the output. We choose to adopt a risk level of 5% (*ie* the probability of saying that two variables are not independent when they are is equal to 0.05). We have to be careful, because with this type of test, we only control the first kind error (*ie* we control the error of rejecting independence when it is verified). If the p-value is less than 0.05, independence is rejected (*ie* the explanatory variable must be kept when we will write the model afterward). If the p-value is greater than 0.05, we cannot conclude anything about the independence between the variables tested. In this case, we cannot determine thanks to the Chi-squared test whether the explanatory variable is interesting for the prediction of injuries. In the following part, we designate by  $\mathcal{H}_0$  the null hypothesis of our test (*ie* the hypothesis we wish to control).

Results of Chi-squared tests of independence are available in Table 2.

<sup>1</sup>*CHI-Squared Test of Independence* , Minhaz Fahim Zibran, Department of Computer Science (2007)

Test	p-value	Conclusion
Nb of injuries indep. of sport practiced	$0.016 < 0.05$	Nb of injuries depends on the sport practiced
Injured areas indep. of sport practiced	$0.008 < 0.05$	Injured areas depend on the sport practiced
Nb of injuries indep. of athlete's status	$0.02 < 0.05$	Nb of injuries depend on athlete's status
LL injuries indep. of previous LL injuries	$0.1 > 0.05$	We cannot conclude
LU injuries indep. of previous LU injuries	$0.003 < 0.05$	LU injuries depend on previous LU injuries
TS injuries indep. of previous TS injuries	$0.005 < 0.05$	TS injuries depend on previous TS injuries
LL injuries indep. of current LL pain	$0.01 < 0.05$	LL injuries depend on current LL pain
LU injuries indep. of current LU pain	$0.02 < 0.05$	LU injuries depend on current LU pain

Table 2: Results of the 8 Chi-squared tests of independence performed on the CREPS dataset.

Thanks to this preliminary analysis, we know which variables can be useful in the construction of our statistical models.

## 4.2 Logistic model

Since our response variable is binary ( “Yes” if the athlete has injured himself during the season, “No” otherwise), we first choose to use a logistic regression. Our goal is to explain the response variable “Injury” (named  $Y$  in the following section) thanks to the  $p$  regressors, noted  $x^{(1)}, \dots, x^{(p)}$  defined as the variables of our dataset (here,  $p = 12$ ). Among the 12 regressors:

- 8 qualitative variables: “New / old” (1 if the athlete is new, 0 otherwise), “Sport”, “Old LL injuries” (OLLI), “Old UL injuries” (OULI), “Old TS injuries” (OTSI), “Actual LL pain” (ALLP), “Actual UL pain” (AULP), “Actual TS pain” (ATSP);
- 4 quantitative variables: “Lower Mobility Sum” (LMS), “Upper Mobility Sum” (UMS), “Pelvic Stability Sum” (PSS), “Scapular Sum and Core Stability” (SSCS).

The response variable  $Y_i|x_i \sim \mathcal{B}(\pi(x_i))$  must ensure that :

$$\mathbb{P}(Y_i = 1|x_i) = \pi(x_i) \text{ where } \pi(x_i) = \mathbb{E}[\pi(x_i)]$$

Then, we could write our logistic regression as follow :

$$\begin{aligned}
\text{logit}(\pi(x_i)) &= \ln \left( \frac{\pi(x_i)}{1 - \pi(x_i)} \right) \\
&= \theta_0 + \theta_1 LMS_i + \theta_2 UMS_i + \theta_3 PSS_i + \theta_4 SSCS_i + \theta_5 \mathbb{1}_{New_i=1} + \theta_6 \mathbb{1}_{OLLI_i=True} + \theta_7 \mathbb{1}_{OULI_i=True} \\
&\quad + \theta_8 \mathbb{1}_{OTSI_i=True} + \theta_9 \mathbb{1}_{ALLP_i=True} + \theta_{10} \mathbb{1}_{AULP_i=True} + \theta_{11} \mathbb{1}_{ATSP_i=True} + \theta_{12} \mathbb{1}_{Sport_i=Baseball} \\
&\quad + \theta_{13} \mathbb{1}_{Sport_i=Basketball} + \theta_{14} \mathbb{1}_{Sport_i=Bowling} + \theta_{15} \mathbb{1}_{Sport_i=FrenchBoxing} + \theta_{16} \mathbb{1}_{Sport_i=Rowing} \\
&\quad + \theta_{17} \mathbb{1}_{Sport_i=Rugby} + \theta_{18} \mathbb{1}_{Sport_i=Swimming} + \theta_{19} \mathbb{1}_{Sport_i=SynchronisedSwimming}
\end{aligned}$$

We suppose here that there isn't any interaction between our variables. It is a strong hypothesis; however, since our logistic regression already has a lot of variables, it will not be easily understandable if we add interactions. Moreover, a model with too many variables will not be computable on R. We then estimate the vector of parameters  $\theta \in \mathbb{R}^{20}$  by maximum of likelihood. To make the

computation easier, we use the log-likelihood function instead of the likelihood one, which is defined as follow:

$$\theta \mapsto l(Y; \theta) = \sum_{i=1}^n \ln[\text{logit}(\pi_i)]$$

The estimator of maximum of likelihood is then:

$$\hat{\theta} = \arg \max_{\theta} l(Y; \theta)$$

Since it is impossible to find an analytical solution to this optimisation problem, we use a Fisher-scoring algorithm to compute  $\hat{\theta}$ .

Once the model adjusted, we have an estimation for each linear predictor  $\eta_i$  by  $\hat{\eta}_i$  with :

$$\begin{aligned}\eta_i &= \theta_0 + \theta_1 LM S_i + \dots + \theta_{19} \mathbb{1}_{Sport_i = SynchronisedSwimming} \\ \hat{\eta}_i &= \hat{\theta}_0 + \hat{\theta}_1 LM S_i + \dots + \hat{\theta}_{19} \mathbb{1}_{Sport_i = SynchronisedSwimming}\end{aligned}$$

We also have an estimation for each parameter  $\hat{\pi}(x_i) = \hat{\pi}_{\theta}(x_i)$ .

If we apply the Bayes rule on the  $\hat{\pi}(x_i)$ , we get the adjusted values  $\hat{Y}_i$  of  $Y_i$  :  $Y_i = \begin{cases} 1 & \text{if } \hat{\pi}(x_i) > 0.5 \\ 0 & \text{otherwise} \end{cases}$

In our case, we have : Injury =  $\begin{cases} Yes & \text{if } \hat{Y}_i = 1 \\ No & \text{if } \hat{Y}_i = 0 \end{cases}$

### 4.3 LASSO regression

In the previous part, we highlighted that our logistic model has a significant number of variables. This is a problem for different reasons :

- in terms of computation time: the more variables we have, the longer the algorithm will take to compute;
- from a prediction perspective: if  $x^{(0)}$  is a new vector of the explanatory variables, we know that the quality (in the sense of squared deviation) of the prediction  $\hat{Y}_0$  of the response  $Y_0$  is decomposed into the squared bias and the variance. In other words, for any estimator  $\hat{\theta}$  of  $\theta$ ,

$$\mathbb{E}[(Y - X\hat{\theta})^2] = \text{Bias}[X\hat{\theta}] + \text{Var}(X\hat{\theta})$$

where X is the design matrix (ie each columns corresponds to one of our variables),

$$\text{Bias}[X\hat{\theta}] = \mathbb{E}[X\hat{\theta}] - X\theta \text{ and } \text{Var}(X\hat{\theta}) = \mathbb{E}[(X\hat{\theta} - \mathbb{E}[X\hat{\theta}])^2]$$

Thus, to improve the prediction, one may prefer a slight increase in the bias to have a decrease in the variance.

In this context, we will try to use so-called LASSO penalized regression method to overcome these difficulties. The idea of the LASSO (ie *Least Absolute Selection and Shrinkage Operator*) regression proposed by Tibshirani is to cancel some coefficients of the vector  $\theta$  to have a sparse estimator. This leads to the selection of variables leading to a more understandable model and a matrix of explanatory variables with better properties than  $X^T X$ .

To force the cancellation of theta coordinates, we constrain its 1 norm:  $\|\theta\|_1 = \sum_{i=1}^p \|\theta_i\|$

The first step is to center-reduce the explanatory variables ( $X$  becomes  $\tilde{X}$ ) and at least center the response vector ( $Y$  becomes  $\tilde{Y}$ ). Therefore, we define the LASSO estimator as:  $\forall \in \mathbb{R}_+^*$ ,

$$\hat{\theta}_{lasso} \in \arg \min_{\theta \in \mathbb{R}^p} \|\tilde{Y} - \tilde{X}\theta\|_2^2 + \lambda \|\theta\|_1$$

The resulting vector of fitted values  $\tilde{X}\hat{\theta}_{lasso}$  is always unique. We can highlight some kinds of behaviour, which depend on the value of the penalisation parameter  $\lambda$ :

- If  $\lambda = 0$ , we get the linear regression estimate  $\hat{\theta}$ ;
- A larger value of  $\lambda$  leads to a sparse solution :  $\lim_{\lambda \rightarrow +\infty} \hat{\theta}_{lasso}(\lambda) = 0$

The penalisation  $\lambda$  have to be choosen carefully. Since it is impossible to find the perfect  $\lambda$  *a priori*, we go through a cross-validation process to stabilize the choice of  $\lambda$  (Figure 1).

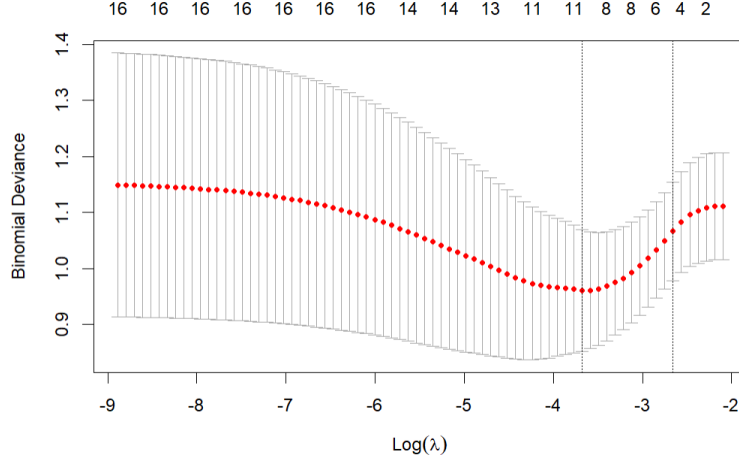


Figure 1: Mean square error as a function of  $\lambda$ .

By minimizing this function, we obtain the optimal value of  $\lambda$ , which is marked by the dotted line.

Thanks to the cross-validation, we are now able to cancel some coefficients of  $\theta$  to perform the variable selection (Figure 2).

Figure 2 shows that the LASSO regression keeps only the nonzero variables.

We perform the LASSO regression for our 4 response variables *Injury*, *LL Injury*, *LU Injury* and *TS Injury*. For each of them, we get the following models:

- $Injury = \theta_0 + \theta_1 SSCS_i + \theta_2 \mathbb{1}_{OLLI_i=True} + \theta_3 \mathbb{1}_{New_i=1} + \theta_4 \mathbb{1}_{Sport_i=Baseball} + \theta_5 \mathbb{1}_{Sport_i=Basketball} + \theta_6 \mathbb{1}_{Sport_i=Bowling} + \theta_7 \mathbb{1}_{Sport_i=FrenchBoxing} + \theta_8 \mathbb{1}_{Sport_i=Rowing} + \theta_9 \mathbb{1}_{Sport_i=Rugby} + \theta_{10} \mathbb{1}_{Sport_i=Swimming} + \theta_{11} \mathbb{1}_{Sport_i=SynchronisedSwimming}$
- $LL\ Injury = \theta_0 + \theta_1 SSCS_i + \theta_2 PSS_i + \theta_3 \mathbb{1}_{OLLI_i=True} + \theta_4 \mathbb{1}_{ALLP_i=True} + \theta_5 \mathbb{1}_{New_i=1} + \theta_6 \mathbb{1}_{Sport_i=Baseball} + \theta_7 \mathbb{1}_{Sport_i=Basketball} + \theta_8 \mathbb{1}_{Sport_i=Bowling} + \theta_9 \mathbb{1}_{Sport_i=FrenchBoxing} + \theta_{10} \mathbb{1}_{Sport_i=Rowing} + \theta_{11} \mathbb{1}_{Sport_i=Rugby} + \theta_{12} \mathbb{1}_{Sport_i=Swimming} + \theta_{13} \mathbb{1}_{Sport_i=SynchronisedSwimming}$

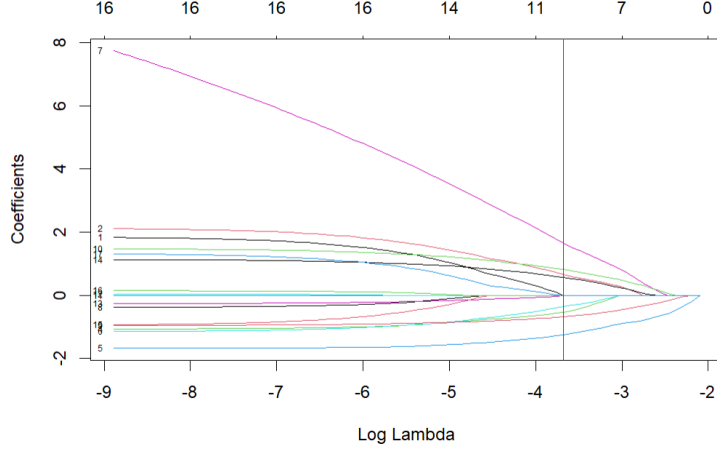


Figure 2: LASSO regression with the optimal value of  $\lambda$  marked by the red line.

- LU Injury =  $\theta_0 + \theta_1 UMS_i + \theta_2 \mathbb{1}_{OULL_i=True} + \theta_3 \mathbb{1}_{Sport_i=Baseball} + \theta_4 \mathbb{1}_{Sport_i=Basketball} + \theta_5 \mathbb{1}_{Sport_i=Bowling} + \theta_6 \mathbb{1}_{Sport_i=FrenchBoxing} + \theta_7 \mathbb{1}_{Sport_i=Rowing} + \theta_8 \mathbb{1}_{Sport_i=Rugby} + \theta_9 \mathbb{1}_{Sport_i=Swimming} + \theta_{10} \mathbb{1}_{Sport_i=SynchronisedSwimming}$
- TS Injury =  $\theta_0 + \theta_1 SSCS_i + \theta_2 \mathbb{1}_{OLLI_i=True} + \theta_3 \mathbb{1}_{New_i=1} + \theta_4 \mathbb{1}_{Sport_i=Baseball} + \theta_5 \mathbb{1}_{Sport_i=Basketball} + \theta_6 \mathbb{1}_{Sport_i=Bowling} + \theta_7 \mathbb{1}_{Sport_i=FrenchBoxing} + \theta_8 \mathbb{1}_{Sport_i=Rowing} + \theta_9 \mathbb{1}_{Sport_i=Rugby} + \theta_{10} \mathbb{1}_{Sport_i=Swimming} + \theta_{11} \mathbb{1}_{Sport_i=SynchronisedSwimming}$

#### 4.4 Clustering of variables

Another useful method to reduce dimension is the clustering of variables. It enables to group variables which are strongly related to each other and thus create homogeneous clusters. Clustering of variables (ClustOfVar) is an alternative to LASSO regression. Where LASSO cancels parameters (meaning we do not take them into account in our final model), ClustOfVar arrange variables into meaningful structures and finally select one metavariable from each group. This metavariable is synthetic; moreover, each variable of each group bring some information, so ClustOfVar might build models more complete than LASSO.

Our approach for clustering our 20 variables is to calculate the dissimilarities between them, and then to apply a cluster analysis method to this matrix of dissimilarities. Since we have a lot of qualitative variables, we can use the correlation ratio as a measure of dissimilarity. Here, we will use a hierarchical clustering because it enable us to choose the number of clusters regarding on the dendrogram we get.

The aim of clustering is to maximise an homogeneity criterion. To define it, we need to introduce some notations: let  $\{x_1, \dots, x_{p_1}\}$  be a set of our  $p_1 = 4$  quantitative variables and  $\{y_1, \dots, y_{p_2}\}$  a set of our  $p_2 = 8$  qualitative variables. We denote by  $X$  and  $Y$  the corresponding quantitative and qualitative data matrices, which are  $n \times p_1$  and  $n \times p_2$  (where  $n = 119$  is the number of individuals). To make the following formulas more understandable, let  $x_j \in \mathbb{R}^n$  be the  $j$ -th column of  $X$ , and  $y_j \in \mathbb{M}_{n,j}(\mathbb{R})$  the  $j$ -th column of  $Y$  (where  $\mathcal{M}_j$  is the set of categories of  $y_j$ ).

We denote by  $\mathcal{P}_K = (\mathcal{C}_1, \dots, \mathcal{C}_K)$  the partition into  $K$  clusters of the  $N = p_1 + p_2$  variables. The synthetic variable of a cluster  $\mathcal{C}_k$  is defined as the quantitative variable  $c_k \in \mathbb{R}^n$  which is the "most



linked” to all the variables in  $\mathcal{C}_k$ :

$$c_k = \arg \max_{u \in \mathcal{R}^n} \left\{ \sum_{x_j \in \mathcal{C}_k} r_{u, x_j}^2 + \sum_{y_j \in \mathcal{C}_k} \eta_{u|y_j}^2 \right\}$$

where  $r^2$  is the square Pearson correlation, and  $\eta^2$  the correlation ratio.  $\eta_{u|y_j}^2 \in [0, 1]$  measures the part of the variance of  $u$  explained by the categories of  $y_j$ .

According to M. Chavent, V. Kuentz Simonet, B. Liquet and J. Saracco, ”a cluster of variables is defined as homogeneous when the variables in the cluster are strongly linked to a central quantitative synthetic variable”. Homogeneity can be mathematically defined as a measure of adequacy between the variables in cluster and its central synthetic quantitative variable  $c_k$ :

$$H(\mathcal{C}_k) = \sum_{x_j \in \mathcal{C}_k} r_{x_j, c_k}^2 + \sum_{y_j \in \mathcal{C}_k} \eta_{c_k|y_j}^2$$

- The first term measures the link between the quantitative variables in  $\mathcal{C}_k$  and  $c_k$ , independently of the sign of the relationship;
- The second term measures the link between the qualitative variables in  $\mathcal{C}_k$  and  $c_k$ .

We can notice that the homogeneity of a cluster is maximum when all the quantitative variables are correlated (positively or negatively) to  $c_k$ , and all the correlation ratios of the qualitative variables are equal to 1. In this case, all the variables in the cluster  $\mathcal{C}_k$  bring the same information. We finally define the homogeneity of a partition  $\mathcal{P}_K$ :

$$\mathcal{H}(\mathcal{P}_K) = \sum_{k=1}^K H(\mathcal{C}_k)$$

Now that we have defined the main terms, our goal is to find a partition of a set of quantitative/qualitative variables such that variables within a cluster  $\mathcal{C}_k$  are strongly linked to each other. It means that we are looking for a partition  $\mathcal{P}_K$  which maximises the homogeneity function  $\mathcal{H}$  previously defined. We use the hierarchical clustering algorithm of the R package ClustOfVar. We get the following dendrogram:

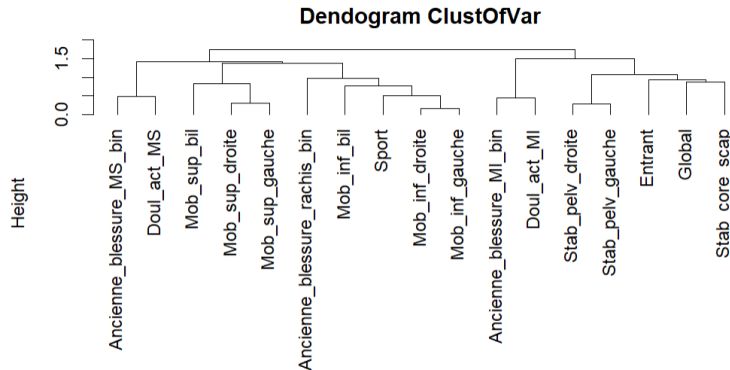


Figure 3: Clustering of Variables dendrogram.

We use a bootstrap procedure to evaluate the stability of the partitions into  $K = 2, 3, \dots, 20$  clusters. First, we draw  $B$  bootstrap samples of our  $n$  observations and get the corresponding  $B$  dendrograms. Then, we compare the partitions of these  $B$  dendrograms with the partitions of the initial hierarchy and evaluate their stability. Finally, we choose the number of clusters which maximise the stability of the partitions without create too many clusters. In our case, 6 clusters seems to be the better choice for 2 reasons:

- The stability of the partitions is higher than if we only take 5 clusters (or less);
- Our clusters are consistent, and there isn't any cluster with NA variables (contrary to the partitions in 7 or more clusters).

The clusters obtained are available in the following table :

Cluster	Variables
Cluster 1	LMS + Sport + OTSI
Cluster 2	UMS
Cluster 3	SSCS + New/Old
Cluster 4	PSS
Cluster 5	OLLI + ALLP
Cluster 6	OULI + AULP

Table 3: Clusters obtained from the Clustering of Variables dendrogram.

Thanks to this clustering, we can write another logistic regression model (different from the one introduced previously):

$$\text{Injury} = \beta_0 + \beta_1(\text{Cluster1})_i + \beta_2(\text{Cluster2})_i + \beta_3(\text{Cluster3})_i + \beta_4(\text{Cluster4})_i + \beta_5(\text{Cluster5})_i + \beta_6(\text{Cluster6})_i$$

We also perform a LASSO regression on the ClustOfVar's logistic model to compare the quality of prediction later. We get the following results:

- $\text{Injury} = \beta_0 + \beta_1(\text{Cluster1})_i + \beta_2(\text{Cluster3})_i + \beta_3(\text{Cluster5})_i + \beta_4(\text{Cluster6})_i$
- $\text{LL Injury} = \beta_0 + \beta_1(\text{Cluster1})_i + \beta_2(\text{Cluster3})_i + \beta_3(\text{Cluster5})_i$
- $\text{LU Injury} = \beta_0 + \beta_1(\text{Cluster1})_i + \beta_2(\text{Cluster2})_i + \beta_3(\text{Cluster6})_i$
- $\text{TS Injury} = \beta_0 + \beta_1(\text{Cluster1})_i + \beta_2(\text{Cluster3})_i + \beta_3(\text{Cluster4})_i + \beta_4(\text{Cluster5})_i$

## 4.5 Classification and Regression Trees

The Classification and Regression Trees (CART) algorithm is "a non parametric method to build estimators in a multidimensional framework" (2). For this project, we will only use classification trees because our response variable is qualitative. The aim here is to partition the space of input variables; to achieve it, trees use a recursive sequence of division rules which are based on a single explanatory variable. We choose to use classification trees because they are really understandable.

<sup>2</sup>Beatrice Laurent Bonneau hands-out.

Most of all, since our supervisors are not used to mathematical approaches, we think that trees would be a simple and effective way for them to determine quickly if an athlete has a high risk of injury. Let's explain the functioning of regression trees. We observe a sample of  $n = 119$  individuals,  $p = 20$  qualitative explanatory variables noted  $X_j$ , and a qualitative variable  $Y$  to predict.  $Y$  can belong to 2 classes:  $\mathcal{C}_1$  if the athlete is injured,  $\mathcal{C}_2$  otherwise.

The construction of a binary discrimination tree can be decomposed as follow:

1. We determine a sequence of nodes; they are defined by the choice of one variable among the  $p$  explanatory and a division which leads to a partition into 2 classes. If the selected variable is qualitative, the division is defined by a split into 2 groups of modalities; otherwise, we use a threshold value to divide. The initial node corresponds to the whole sample, and we iterate the process on each of the subsets.
2. We need to define a criterion thanks to which we select the best division among all admissible ones. Our goal is to divide the observations which compose a node into two more homogeneous groups regarding the response variable  $Y$ . Let's note  $\eta$  a node of our tree. Iterating one division of  $\eta$  creates 2 son nodes,  $\eta_{left}$  and  $\eta_{right}$ . Among all the possible divisions of  $\eta$ , the algorithm keeps the one which minimises the sum of the heterogeneities of  $\eta_{left}$  and  $\eta_{right}$ . In other words, for each node  $\eta$ , we have to solve:

$$\max_{\{\text{divisions of } X_j, j \in \{1, \dots, n\}\}} \mathcal{D}_\eta - (\mathcal{D}_{\eta_{left}} + \mathcal{D}_{\eta_{right}})$$

where  $\mathcal{D}_\eta$  is the heterogeneity at the node  $\eta$  defined as  $\mathcal{D}_\eta = |\eta| \sum_{l=1}^p p_\eta^l (1 - p_\eta^l)$ .  $p_\eta^l$  denotes proportion of the class  $\mathcal{C}_k$  of  $Y$  in the node  $\eta$ .

3. We determine a rule to decide if a node is terminal: it thus becomes a leaf. There are 3 cases which stops the growth of the tree at a given node:
  - it is homogeneous, *ie* all the individuals of the node have the same value for  $Y$ ;
  - there isn't any admissible partition left;
  - the number of observations in the node is less than 5; we define 5 as a prescribed value to avoid overfitting.
4. Finally, we assign each leaf to a class  $\mathcal{C}_k$  of  $Y$  by a majority vote ( $k = 1, 2$ ).

The maximal tree obtained is available in Figure 4.

Once we get the maximal tree, we try to prun it. However, we obtain a "*only one leaf*" tree at each try, which is not satisfying at all. Thus, we choose to keep the maximal tree even if it leads to instability, over fitting and lack of robustness.

We try to improve the cons of classification trees by reducing the variance thanks to random forests. Let  $Y$  be our qualitative response (interpreted as "*Injury*"),  $X_1, \dots, X_p$  our explanatory variables ( $p = 20$  as usual),  $x = \{x_1, \dots, x_p\} \in \mathbb{R}^p$  and  $\hat{f}(x)$  a predictor. We note  $\mathcal{S} = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$  a  $n$ -sample which have a distribution  $F$ . Let's consider  $B$  samples  $\{\mathcal{S}_b\}_{1, \dots, B}$ . Since considering  $B$  independent samples would require too much data, they are therefore replaced by  $B$  bootstrap samples. Each of them is obtained by  $n$  draws with replacement according to the empirical distribution  $\hat{F}_n$ .

In our case,  $Y$  is qualitative, so a predictor by model aggregation is defined as a majority vote:

$$\hat{f}_B(.) = \arg \max_j \text{card}\{b | \hat{f}_{Sb}(.) = j\}$$

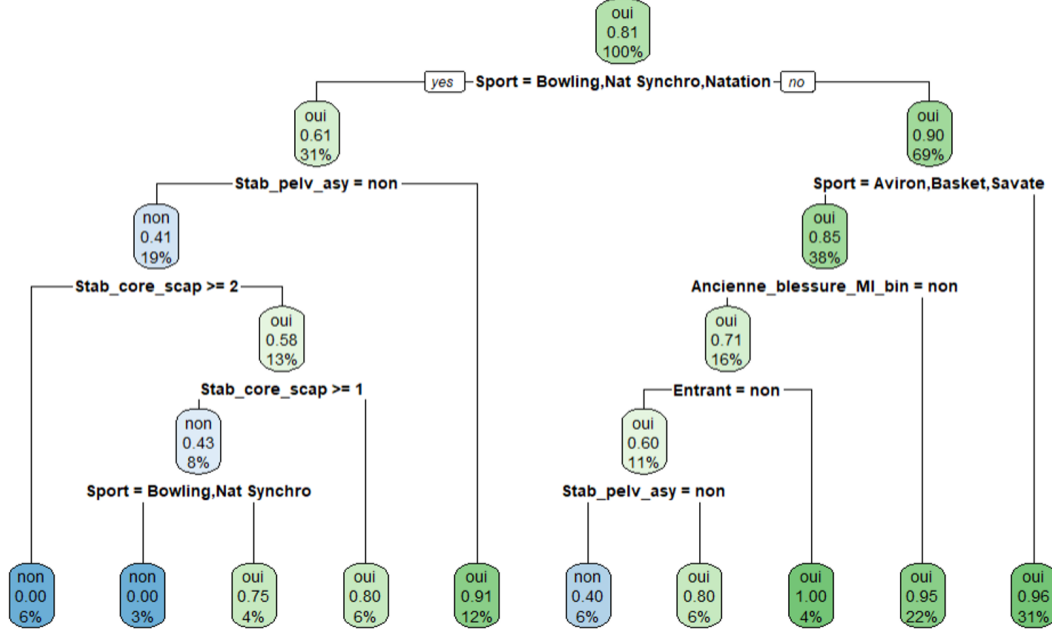


Figure 4: Maximal regression tree obtained from the CREPS dataset.

However, the  $B$  samples are built on the same learning sample  $\mathcal{S} = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$ . It leads that the estimators are not independent. Thus, we need to find low correlated predictors, which is possible thanks to Random forests. The objective is to make the aggregated previously obtained trees more independent by adding randomness in the choice of the variables involved in the prediction. The algorithm works as follow:

1. Let  $X_0$  and  $\mathcal{S} = \{(X_1, Y_1), \dots, (X_n, Y_n)\}$  a learning sample.
2. For  $b = 1$  to  $B$ :  
We take a bootstrap sample  $z_b$  and estimate a tree on it with randomization of the variables : we perform a random selection of a subset of  $p' < p$  predictors. Then, we search an optimal division.
3. Finally, we calculate the result of a majority vote.

We use the pruning strategy implemented by default by the algorithm. Indeed, the pruned trees obtained are strongly correlated because the same subset of variables appear to explain a large part of our dataset. Then, the size of the trees is only limited by the minimum number of observations observed per leaf, which is set to 5 by default (same as for regression trees). Therefore, we can aggregate complete trees; they have a quite low bias, but a high variance. The last one is caused by the random selection of  $p' < p$  potential predictors at each node. If we take each tree separately from the other, it is less optimal and efficient than a single regression trees. Nevertheless, when we aggregate all the trees, we get good results on our dataset (section **Results**). We tune only 2 parameters for random forests:

- The number  $B$  of trees in the forest. After many trials, we choose  $B = 500$  bootstrap samples to build the random forest (which is in fact the default parameter);
- The number  $p'$  of potential predictors at each node: we choose  $p' = \sqrt{p}$  as we work on a classification problem.

## 4.6 Naive Bayes

The Naive Bayes Classifier (*Friedman et al., 1997*) is "a simple probabilistic classifier that learns from training data and then predicting the class of the test instance with the highest posterior probability". The fundamental hypothesis used in this classifier is that the effect of an attribute on a given class is independent of the values of the other attributes. We name this assumption "class conditional independence". This is a strong hypothesis, since our variables are generally not independent. However, Naive Bayes Classifier has shown significant results on many classification problems, so it seems to be a good idea to test it on our dataset.

Let *Train* be a training set of samples; each has a "class label". In our case, there are 2 classes:  $\mathcal{C}_1 = \text{"Injured athletes"}$ , and  $\mathcal{C}_2 = \text{"No injured athletes"}$ . Each sample of *Train* is represented by a  $p$ -dimensional vector  $\mathbb{X} = \{x_1, \dots, x_p\}$ , where  $x_1$  is the measured value of the variable  $X_1$ , ...,  $x_p$  is the measured value of the variable  $X_p$ . With a sample  $\mathbb{X}$ , the Naive Bayes classifier will predict that  $\mathbb{X}$  belongs to the class which have the highest *a posteriori* probability (conditioned on  $\mathbb{X}$ ). In other words,  $\mathbb{X}$  is predicted to belongs to  $\mathcal{C}_k$  if and only if  $\mathbb{P}(\mathcal{C}_k|\mathbb{X}) > \mathbb{P}(\mathcal{C}_l|\mathbb{X})$ , where  $k, l \in \{1, 2\}, k \neq l$ . Thanks to the Bayes' theorem, we can easily compute the probability  $\mathbb{P}(\mathcal{C}_k|\mathbb{X})$ :

$$\mathbb{P}(\mathcal{C}_k|\mathbb{X}) = \frac{\mathbb{P}(\mathbb{X}|\mathcal{C}_k)\mathbb{P}(\mathcal{C}_k)}{\mathbb{P}(\mathbb{X})}$$

The class  $\mathcal{C}_k$  which maximises  $\mathbb{P}(\mathcal{C}_k|\mathbb{X})$  is named "the maximum *a posteriori* hypothesis". Since we previously supposed the independence of our variables, we can write  $\mathbb{P}(\mathbb{X}|\mathcal{C}_k) \approx \prod_{i=1}^p \mathbb{P}(x_i|\mathcal{C}_k)$ .

- If  $X_i$  is qualitative,  $\mathbb{P}(x_i|\mathcal{C}_k)$  is the number of samples of  $\mathcal{C}_k$  in *Train* which have the value  $x_i$  for the variable  $X_i$ , divided by the number of sample of  $\mathcal{C}_k$  in *Train*;
- If  $X_i$  is quantitative, we assume that the values have a gaussian distribution. Then, we have  $\mathbb{P}(x_i|\mathcal{C}_k) = f(x_i, m_{\mathcal{C}_k}, \sigma_{\mathcal{C}_k})$ , where  $f$  is the gaussian distribution of mean  $m_{\mathcal{C}_k}$  and standard deviation  $\sigma_{\mathcal{C}_k}$ .

As  $\mathbb{P}(\mathbb{X})$  remains the same, the Naive Bayes classifier predicts that the class label of  $\mathbb{X}$  is  $\mathcal{C}_k$  if and only if  $\mathcal{C}_k$  maximises  $\mathbb{P}(\mathbb{X}|\mathcal{C}_k)\mathbb{P}(\mathcal{C}_k)$ .

## 5 Main results with each statistical model and selection of the best one

Once we have chosen and explained the different models and algorithms we used, we need to implement them on R and present the results we get. To determine which model is the most efficient, we have to evaluate their prediction capacity thanks to performance indicators. The first step of a predictive approach is to divide randomly our dataset in two samples:

- The *train* sample, used to train the different models (by minimising the number of incorrect predictions);
- The *test* sample, used for model selection: we choose the model with the smallest number of incorrect predictions.

We choose to split our dataset in 90 individuals for the training test (which represents nearly 75% of the entire dataset), and 29 individuals for the test set. As we have a two class classification

problem, we can use a prediction method which provides an estimator of  $\mathbb{P}(Y = 1|X = x)$ . A natural prediction is to affect  $Y$  to 1 (for us,  $Y \in \mathcal{C}_1$ , *ie* the athlete is injured) if:

$$\hat{\mathbb{P}}(Y = 1|X = x) > \frac{1}{2}$$

We then have a symmetric role to classes 0 and 1. We use this natural prediction on our test sample:  $\forall i \in \{1, \dots, 29\}$ ,

$$\begin{cases} \hat{Y}_i = 1 & \text{if } \hat{\mathbb{P}}(Y_i = 1|X = x_i) > \frac{1}{2} \\ \hat{Y}_i = 0 & \text{if } \hat{\mathbb{P}}(Y_i = 1|X = x_i) \leq \frac{1}{2} \end{cases}$$

Once we have applied this prediction to the entire *test* set, we use a contingency table to visualise the number of good and bad predictions. This contingency table is obtained from a confusion matrix which crosses the modalities of the predicted variable  $Y$  with those of the observed variable. A contingency table takes the following form:

	Injured athlete ( <i>ie</i> $Y_i = 1$ )	Healthy athlete ( <i>ie</i> $Y_i = 0$ )
Positive prediction ( <i>ie</i> $\hat{Y}_i = 1$ )	True positive	False positive
Negative prediction ( <i>ie</i> $\hat{Y}_i = 0$ )	False negative	True negative

Table 4: Contingency table.

In our case, ROC ("Receiver Operating Characteristic") curves seems to be a good estimator of performance of our models. Indeed, as we want to detect if an athlete has a high risk of injury, our study can be related to other medical diagnostic tests in which ROC curves are widely spread. ROC curves also have a great advantage: they are independent of the prevalence of the injury. A ROC curve plots the True Positive Rate (TPR) versus the False Positive Rate (FPR). The first one corresponds to the sensitivity of a diagnostic test (*ie* "*detection of disease when disease is truly present*"). We note the TPR and FPR as follow:

$$TPR = \frac{\text{card}(i|\hat{Y}_i = 1, Y_i = 1)}{\text{card}(i|Y_i = 1)}$$

$$FPR = \frac{\text{card}(i|\hat{Y}_i = 1, Y_i = 0)}{\text{card}(i|Y_i = 0)}$$

The "ideal" ROC curve correspond to  $TPR = 1$  and  $FPR = 0$ , which means that our model does not provide any error of classification. A model is considered acceptable if its ROC curve is above the linear function  $x \mapsto x$  (*ie* we have one in two chance to correctly predict a presence / absence of injury). If a ROC curve is under the linear function, it means that the associated model is less performant than heads or tails, which is widely unacceptable. We show on Figure 5 the ROC curves obtained for the models we use.

All the ROC curves are above the linear function, which means that no model is too bad. Thus, we can keep all of them for further performance analysis.

In many cases (as ours), the ROC curves cannot be used directly to choose the best model overall. Indeed, most of the time, curves intersect themselves at least once, so it is hard to determine with the naked eye which one is the best. Therefore, we use the Area Under the Curve (AUC) criterion to compare our model. We estimate it on our *test* sample because the AUC is too optimistic on the learning sample.

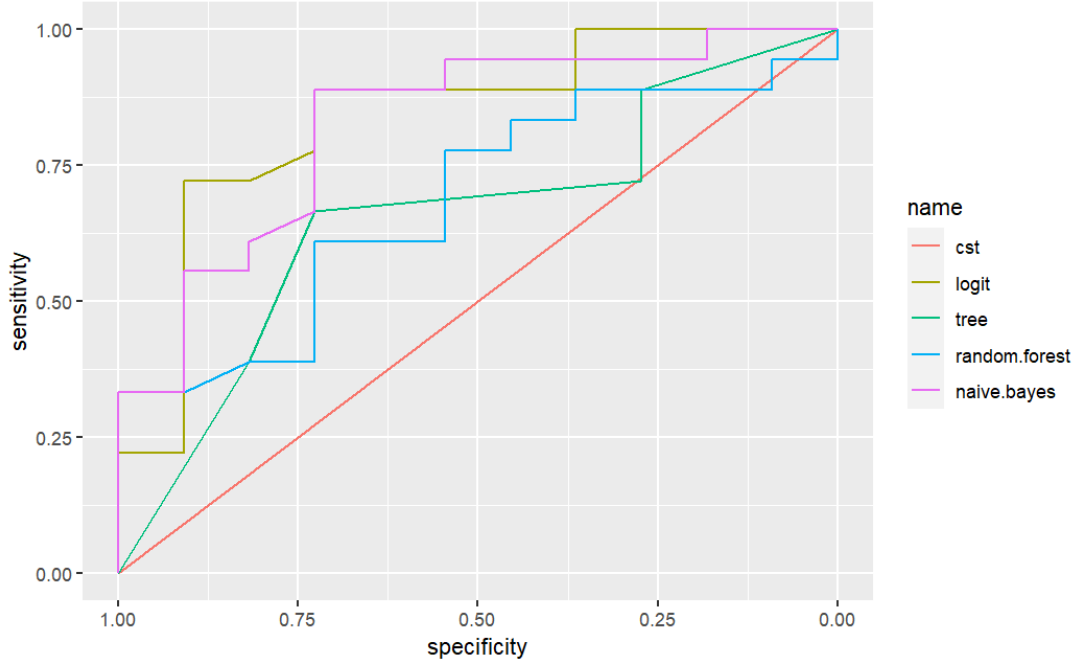


Figure 5: ROC curves of logistic regression, regression tree, random forest and naive Bayes.

For the general category "*Injury*", we first print the AUC boxplots to compare models. As results are not very satisfying, we use the LASSO regression of **Section 4** on all our models and print the new boxplots (Figure 6).

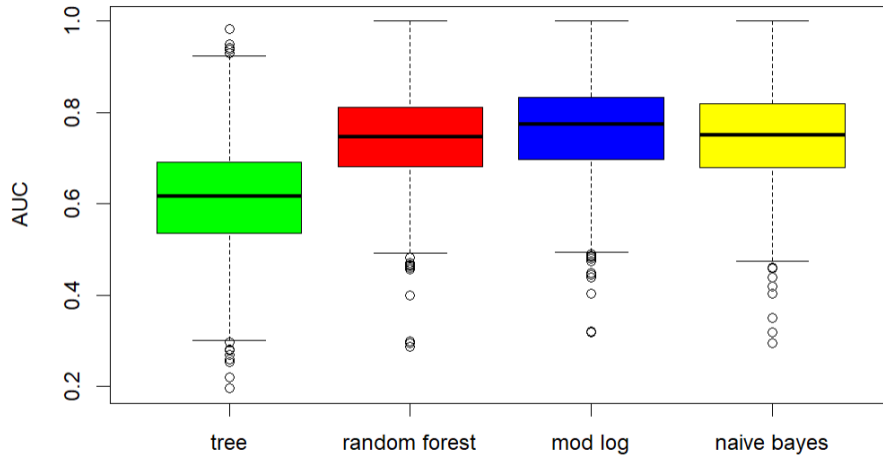


Figure 6: AUC boxplots of regression tree, random forest, logistic regression and naive Bayes.

We note that, in term of AUC, the LASSO logistic regression seems to be the best model. We also print the AUC boxplot of:

- the logistic regression associated to the Clustering of Variables presented in **Section 4**;
- the LASSO logistic regression associated to the same Clustering of Variables.

On Figure 7 too, the LASSO logistic regression is the best model in term of AUC, but random

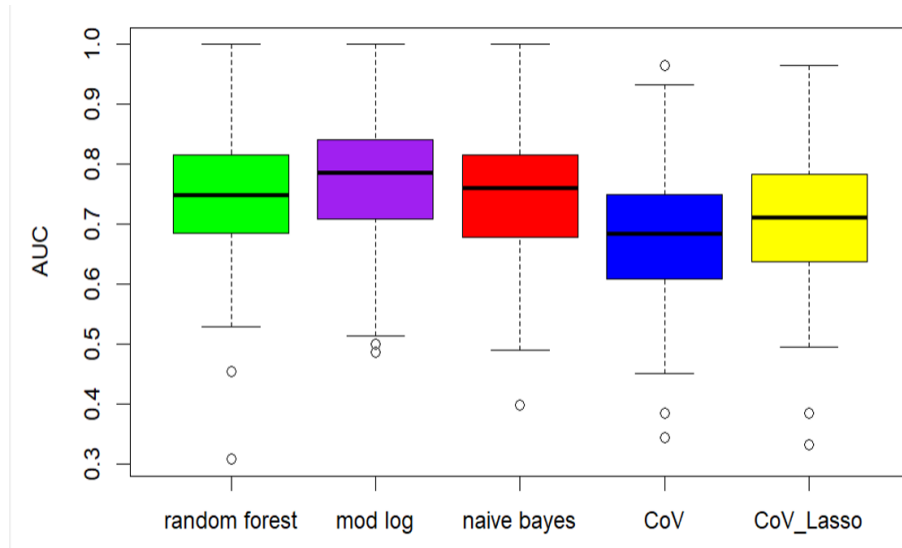


Figure 7: AUC boxplots of previous methods, ClustOfVar logistic regression and LASSO ClustOfVar logistic regression.

forest and naive Bayes are good too.

However, we decide to not only compare our models thanks to AUC, but also with indicators more precise. We choose the following rule:

- If  $\hat{P}(Y_i = 1|X = x_i) > 0.7$ , we predict  $\hat{Y}_i$  by "Injury";
- If  $\hat{P}(Y_i = 1|X = x_i) \leq 0.3$ , we predict  $\hat{Y}_i$  by "No injury";
- If  $0.3 < \hat{P}(Y_i = 1|X = x_i) \leq 0.7$ , we predict  $\hat{Y}_i$  by "Uncertain".

Then, we print the boxplots of good predictions for *LL Injury* and *No LL injury* (Figure 8). We choose to print it for this category of injury instead of the general one "Injury" because the prediction of "No LL injury" is good. This is not the case with the general category because only a few athletes does not have any injury during the year, so it is hard to predict.

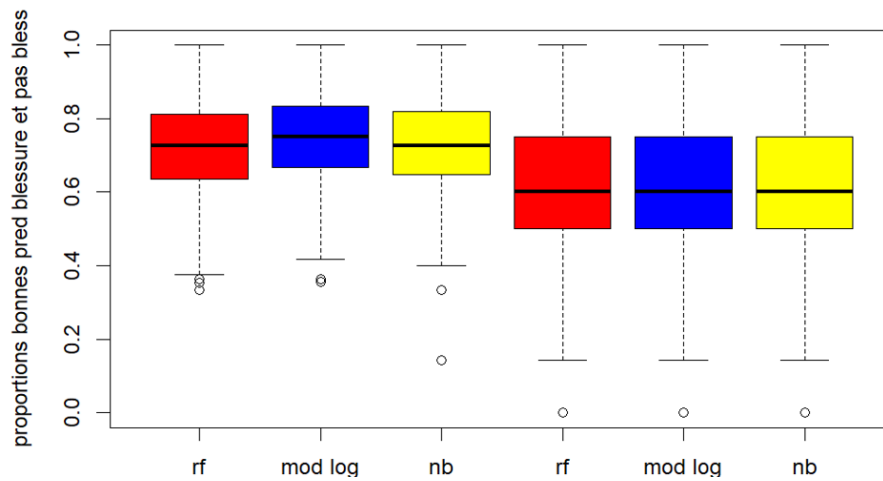


Figure 8: Boxplots of correct predictions for *LL Injury* and *No LL injury*.

The first three boxplots represents the correct predictions for *LL Injury* with the best models



(ie rf = random forest, mod log = logistic regression, nb = naïve Bayes). The last three boxplots represents the correct predictions for *No LL injury* with the same models as previously. This time again, the LASSO logistic regression seems to be the best model to predict both *LL Injury* and *No LL Injury*. However, the three models are quite equivalent for this indicator: they almost have the same median and standard deviation.

We finally print the number of "Uncertains" for each method. It is also a good way to compare models: if a method has a low number of "Uncertains", it has a better chance to predict the presence / absence of an injury correctly (as we have seen on Figure 8).

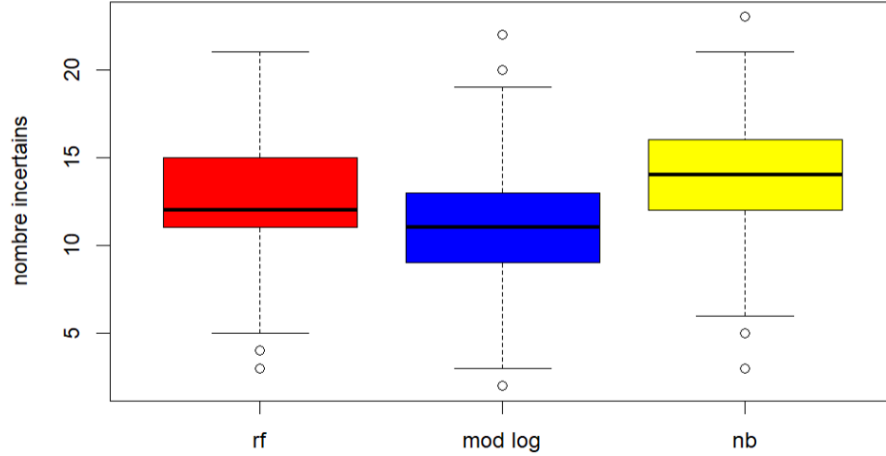


Figure 9: Boxplots of number of uncertain for *LL Injury* and *No LL injury*.

Figure 9 clearly shows that the number of "Uncertains" is the lower with the LASSO logistic regression (with a median around 11 against 12 and 14 for random forest and naïve Bayes respectively). Once again, the LASSO regression appears to be the best model among all.

We use the three indicators previously presented (AUC, good predictions for a type of injury, number of uncertain) for all our categories of injury. In Table 5, for each category on injury, we indicate the best for each indicators (LASSO LR = LASSO logistic regression):

	AUC	Good predictions	Nb of uncertain
Injury	LASSO LR	LASSO LR	LASSO LR
LL Injury	Naive Bayes	LASSO LR	LASSO LR
UL Injury	Naive Bayes	Random forest	Random forest
TS Injury	Naive Bayes	LASSO LR	Random forest

Table 5: Best model dependent on the type of injury and the estimator used.

Finally, we choose to keep only one model to predict each kind of injury. For all injuries except TS, we use a majority vote: if a model is considered as the most efficient for at least 2 indicators, we use it to predict the injury. For *TS Injury*, as 3 different models are the best for the 3 indicators, we keep the one which make a better compromise (ie the model is not too bad on the indicators for which it is not the best). Then, we get the following results:

- LASSO LR will predict *Injury* and *LL Injury*;
- Random forest will predict *UL Injury*;
- Naive Bayes will predict *TS Injury*.

We also tried to perform a Multiple Correspondance Analysis (MCA), in order to do a clustering afterward. Unfortunately, as our dataset is hard to operate, we could not find 2 axis (which are a combination of our variables) that can explain a significant part of the variance. Thus, it is hard to perform a clustering with the aim to separate injured athletes from others. We choose to not print our results here because they cannot be interpreted, and we do not use them to select the best model for each injury.

We also wanted to use methods of boosting such as AdaBoost and XGboost. However, they were highly inefficient in terms of AUC: only 0,43 for AdaBoost (which is worse than heads or tails), and 0,54 for XGboost. Moreover, these boosting methods are black boxes, so they are not very useful for our supervisors as they cannot really interpret the impact of each variable. Thus, we choose to not present them neither.

## 6 Presentation of the R interface

One of the requests from our CREPS supervisors was to provide them with a tool that they could use as a decision support. Indeed, the coaches have implemented corrective routines to prevent their athletes from injuries. These routines are based on the TM2S score of the athletes. For example, if the coaches notice that an athlete has a poor score on Lower Mobility, he would provide him with a corrective routine focused around Lower Mobility improvement exercises. However, these routines are only based on the TM2S score and the empirical observations of the coaches. They do not take into account variables which turned to be essential, such as Sport and New / Old. Then, one part of our job was to find a tool that can be used by our supervisors even if they do not have mathematical skills and cannot use the R software. We first think to implement our own website; unfortunately, we do not have the time and the computer skills to do it properly. After some research, we finally choose to use the R Shiny interface, which is quite easy to use and convenient for our supervisors. Indeed, they just have to install R Studio on their computer and the libraries needed for the implementation of the models. Once we have given them the code to print the interface, they just have to click on the "Run" button. The names of variables, the recap of the athlete and the pie charts are written in French at the request of our supervisors.

On the left part of the window, the coaches can select manually the value of each variable for an athlete in particular (Figure 12 - Appendix). On the right part of the window, at the top, the recap of the athlete is available: we provide the selected value for each variable (Figure 10).

Bellow the recap, we print several things:

- For each type of injury, we print the probability of an athlete to be injured during the season. Each probability is computed from the best model we selected in **Section 5**. The probabilities evolve in real time; if the coaches change the value of a variable, some probabilities will certainly change too (it depends if the variable is relevant in the different models). We choose to print the real probability instead of  $\hat{Y}_i = 0$  or 1 because the coaches could evaluate precisely the risk of injury of an athlete;
- Underneath the probability, we plot the pie part of *Unavailability*. Thanks to the CREPS definition of unavailability, we can split this variable into 3 categories:
  1. *First type*: if an athlete cannot train for one week or less;
  2. *Second type*: if an athlete cannot train for a period between one and three weeks;
  3. *Third type*: if an athlete cannot train for more than three weeks.

The result is available in Figures 11. For each pie chart, we use the entire dataset to compute the different proportions (*ie* we do not make the unavailability depend on other variables, such as the sport or the fact that an athlete is new / old in the structure).

## Récap du sportif

Name	Value
Sport	Aviron
Somme mob inf	4
Somme mob sup	2
Somme stab pelv	3
Stab core + pelv	1
Entrant	oui
Anciennes blessures aux membres inf	oui
Anciennes blessures aux membres sup	oui

Figure 10: Right part of the Shiny interface: athlete’s recap.

## 7 Conclusion and discussion

Our main goal was to predict athlete’s injuries thanks to the TM2S score and other significant variables (the sport practiced, the past of the athlete or the fact that he is new / old in the structure). With machine learning algorithms and statistical models, we achieve to correctly predict between 60 and 80% of each type on injury (Global, Lower Limb, Upper Limb, Torso and Spine). These results are satisfying, most of all because our predictions are not an end in themselves. Indeed, they will be used as a decision support tools for our supervisors. Thanks to the probabilities computed and the empirical observations of the coaches, they will be able to take a decision to decrease the risk of injury of an athlete (as corrective routines, a diminution of training volume or an adaptation of some motions). The pie charts also give them an idea of the athlete’s unavailability if, unfortunately, he is injured during the season.

The second objective of this project was to provide coaches with an easily understandable tool that can be used without particular mathematical or computational skills. When we presented the Shiny interface to our supervisors, we taught them how to fill the different cursors and to interpret the printed probabilities. Therefore, they should be able to use the Shiny interface on new athletes.

Finally, we want to discuss about the difficulties we had to overcome. First of all, the size of the dataset is very poor (only 119 athletes) compared to the number of variables ( $p = 20$ ) so it was hard to find efficient models to predict injuries. Moreover, most of the variables we studied are qualitative, and are harder to exploit than the quantitative ones. For example, we could not use a corrplot to test linear dependencies between these variables (indeed, corrplots have to be used on quantitative variables only). Unfortunately, some efficient models in the machine learning toolbox (such as neuronal networks) are not adapted for our dataset because of these qualitative variables.

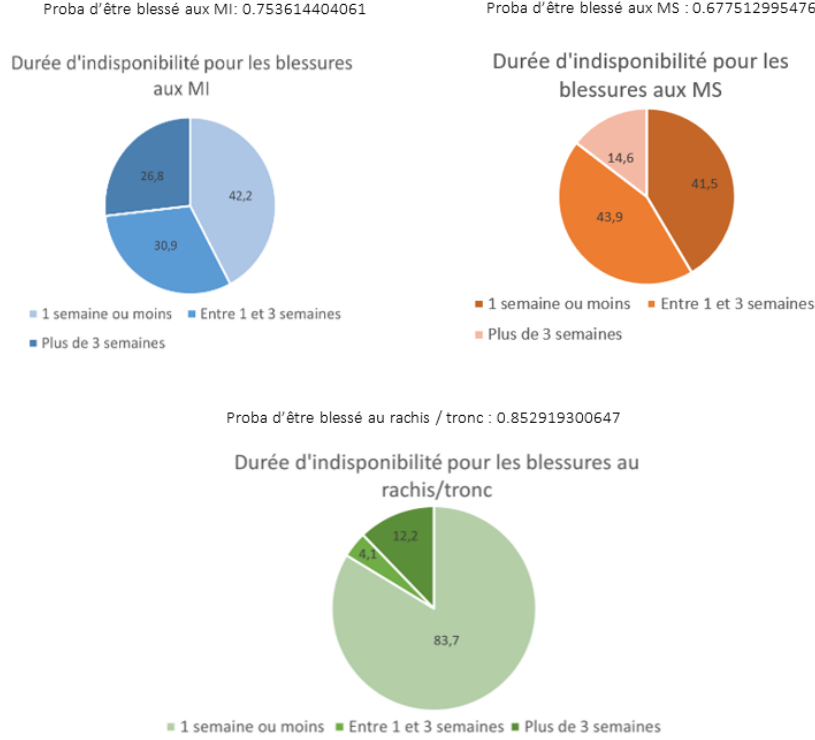


Figure 11: Pie charts of the unavailability for LL, UL and TS injuries.

Furthermore, the values of these qualitative variables are measured by humans, not machines, and different coaches evaluated the TM2S on athletes. Therefore, bias is originally present in the dataset, and we cannot do anything to avoid it.

Another problem we faced concerns the duration of downtime. Once we succeeded to have a good prediction rate for each type of injury, our supervisors asked us to also predict the duration of downtime. Nevertheless, a basic descriptive analysis shew that this prediction was not possible for our dataset. Indeed, some athletes had several injuries on the same part of the body, but their unavailability was not the same for each injury. It is problematic, because in this particular case, we cannot predict the value of the duration of downtime. This example (which is actually recurrent in our dataset) explains why we cannot fit prediction models for downtime. It is a pity; having an idea of the unavailability of an athlete would have been a significant clue for coaches.

The last point we want to highlight concerns the TM2S exercises. During all this study, we do not take into account the fact that TM2S exercises may not be independent two at a time. However, the dependency of exercises could affect the variables we use, because they are obtained from the TM2S form. As we have explained it previously, we cannot just print a corrplot of the TM2S exercises to analyse their redundancy:

- The TM2S exercises can be regarded as qualitative variables, so a corrplot is not suitable;
- Furthermore, even if we consider them as quantitative variable, the corrplot only underlines linear correlation, which is very restrictive.

Moreover, we based our study on the TM2S scores obtained from the testing performed at the beginning of the season. However, corrective routines are given to athletes to improve their mobility an stability. Therefore, the link between TM2S results and injuries is biased. To have a real idea of this relation, coaches should not give corrective routines to the athletes during an entire season. Nevertheless, it seems to be unachievable and contrary to the first objective (*ie* injuries prevention).

Finally, the binary notation of the TM2S exercises may not be the best. Even if each exercise differentiates athletes, the variable associated to one exercise is not quantitative. Therefore, the LMS, UMS, PSS and SSCS can be regarded as quantitative or qualitative, which is not optimal for prediction models. If the notations were not binary, the previous variables would be only considered as quantitative one, which could improve the quality of prediction.

Due to the points highlighted above, our results could be improved on several levels, but they are a good start as a decision support tool.

We want to acknowledge our supervisor, Marine GARGAGLI (Department of Sport Performance), for her help and her questions, which allowed us to improve our vulgarisation skills. She also accepted us as a second working group, which was not the case when we first received the project attribution. We also thank Thomas BAUDRY (coach) for his presentation of the TM2S and the main goals of this project.

## 8 Referencies

- *GMC méthode de prévention au service de tous*(<https://www.global-mobility-condition.fr/>) (accessed 2022/02/14);
- *FMS an introduction to the functional movement screen*, 2021 (<https://www.functionalmovement.com/files/Articles>)(accessed 2022/02/15);
- Cathy Maugis Rabusseau’s hands-out (accessed in 2022/02/30);
- Juliette Chevalier’s hands-out (accessed in 2022/03/02);
- Beatrice Laurent Bonneau’s hands-out (accessed in 2022/03/10);
- *ClustOfVar: An R Package for the Clustering of Variables*, Marie Chavent, V. Kuentz Simonet, Benoit Liquet, Jérôme Saracco ([https //hal.archives-ouvertes.fr/hal-00742795/document](https://hal.archives-ouvertes.fr/hal-00742795/document)) (accessed in 2022/03/18);
- *Naive Bayesian Classifier*, K. Ming Leung, Polytechnic University (<https://cse.engineering.nyu.edu/~mleung/FRE7851/f07/naiveBayesianClassifier.pdf>) (accessed in 2022/03/20)
- *ROC Curves in Clinical Chemistry: Uses, Misuses, and Possible Solutions*: N. Obuchowski, M. Lieber, F. Wians, Jr.(<https://academic.oup.com/clinchem/article/50/7/1118/5640054?login=true>) (accessed in 2022/03/05);
- *CHI-Squared Test of Independence*, Minhaz Fahim Zibran, Department of Computer Science (2007) (accessed in 2022/02/29)

## 9 Appendix

**Donnees du sportif**

**Choisir le sport :**

Aviron ▼

**Somme mob inf :**

0 4 8

0 1 2 3 4 5 6 7 8

**Somme mob sup :**

0 2 4

0 1 2 3 4

**Somme stab pelv :**

0 3 6

0 1 2 3 4 5 6

**Stab core + Scap :**

0 1 2

0 1 2

**Entrant dans la structure :**

oui ▼














**Anciennes blessures au membres inf :**

oui ▼

**Anciennes blessures au membres sup :**

oui ▼

Figure 12: Left part of the Shiny interface: data entry.

Critères de réalisation		Tests	Scores	
MOBILITÉ MEMBRES INFÉRIEURS	<b>1-ADDUCTORS STRETCH</b> Fesses et dos collés au mur / Jambes tendues / Mâtéoles qui touchent le mur		1	
	<b>2-HAMSTRING STRETCH</b> Les deux jambes tendues et collées au mur ou au sol / Pointes de pieds fléchies / Pas de rotation externe de la jambe		G 1	D 1
	<b>3-QUAD STRETCH</b> Bassin en avant de la ligne épaule genou (fléchisseurs de hanche) / Talon touche la fesse (extenseurs du genou)		G 1	D 1
	<b>4-PIGEON POSE</b> Extérieur jambe au sol jusqu'à la fesse / Jambe arrière tendue dans l'axe / Buste sur le tibia / Bassin plus bas que les épaules / Avants bras sur le sol		G 1	D 1
MOBILITÉ MEMBRES SUPÉRIEURS	<b>5-CHANDELIER</b> Fesses, dos (en entier) et épaules collés au mur / Bras plaqués à l'horizontale / Avant-bras, poignets, mains et doigts au mur / Maintien 15s		1	
	<b>6-ARM CLOCK</b> Jambe tendue au sol / Pointes de pieds relevées / La main et le genou (fléchi) gardent le contact avec le sol durant tout le mouvement		G 1	D 1
	<b>7-REVERSE TABLE</b> Alignement tête épaules bassin genoux / Jambes écartées de la largeur du bassin / Genou dans l'axe du tronc / Poignets sous les épaules / Doigts vers les pieds / Stomum plus haut que les épaules / Maintien 15s		1	
MOBILITÉ GLOBALE	<b>8-OVER HEAD SQUAT</b> Pieds largeur de bassin / Bâton au-dessus des pieds / Dos plus redressé que tibias / Fesses sous les genoux / Pas de valgus genoux et chevilles		1	
STABILITÉ CEINTURE PELVIENNE	<b>9-HURDLE STEP</b> Équilibre sur une jambe, les yeux fermés, pendant 15s / Bâton parallèle à la halle / Buste droit (pas d'inclinaison ou torsion) / Jambes parallèles / Bassin horizontal / Ne pas toucher la halle lors du franchissement		G 1	D 1
	<b>10-HAMSTRING CURL</b> Avants bras décollés / Angles 90° chevilles et genoux / Alignement genou hanche épaules / Pas de valgus genou / Maintien 15s		G 1	D 1
	<b>11-COPENHAGEN PLANK</b> Main au sol, bras tendu / Ligne des épaules verticale / Alignements frontal et sagittal cuisses torse tête / Pieds fléchis en contact avec le banc / Maintien 15s		G 1	D 1
STABILITÉ C. SCAP.	<b>12-PUSH UP</b> Mains écartées de la largeur des épaules, au niveau du menton (garçons) ou des épaules (filles) / Jambes collées et genoux et coudes décollés du sol / Le corps se déplace en un seul bloc		1	
STABILITÉ CORE	<b>13-CORE ACTIVATION</b> Maintenir un élastique plaqué entre le sol et les lombaires / Fléchir les chevilles et lever les pieds de 30cm / Garder la tête et les épaules au sol / Maintien 15s		1	

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

MOBILITÉ M. INF.	7	MOBILITÉ M. SUP.	4	MOBILITÉ GLOBALE	1	STABILITÉ C. PELV.	6	STABILITÉ C. SCAP.	1	STABILITÉ CORE	1
------------------	---	------------------	---	------------------	---	--------------------	---	--------------------	---	----------------	---

Figure 13: Exercises sheet used by physical coaches.