UNIVERSIDAD TORCUATO DI TELLA

Master in Management + Analytics.
Escuela de Negocios

Making the cut: forecasting non impact injuries in professional soccer.

Student: Agustin Cicognini

Advisor: Pablo Roccatagliata

December 2021

Making the cut: forecasting non impact injuries in professional soccer.

ABSTRACT

This paper proposes a methodology to predict work in non-traumatic injuries in professional soccer players. The task to be solved is a classification problem of the player's status with a window of 72 hours. The data set used corresponds to records of complete training by the players of Belgrano de Córdoba professional soccer team of the first division of Argentina. The chosen model is GBM with an AUC of 0.7. Interpretation exercises based on SHAP are performed on the chosen model to analyze the characteristics that determine the model's predictions. In addition, possible extensions are proposed such as the use of the results of the model at the time of contractual negotiation given the estimated proportion of time that the player will spend outside due to injury and the economic cost of those absences given, at least, by the direct salary cost of that player. Another approach to the injury forecasting problem based on survival time models is also discussed.

Keywords: non-traumatic injury, professional football, machine learning, survival analysis, SHAP.

Index

1. Introduction

Injury incidence has been a constant concern in modern sport science both in the scientific environment and in the applied professional field. Availability of full teams is associated with teams success. Negative correlation between injury incidence and likelihood of classification to finals has been demonstrated in European professional teams (Ekstrand, 2013). Also, economical losses associated with time loss of injured players should be taken into account. It has been estimated that the mean annual costs of injuries for European teams is €500,000 (Hägglund et al., 2013). Therefore, injury prevention is an essential topic in professional teams and in international sports federations (McCall, Dupont, & Ekstrand, 2016).

Modern football has been characterized for its high intensity and high physical demand, more elevated than in past decades (Bush, Barnes, Archer, Hogg, & Bradley, 2015). In a typical match the mean distance covered by each player is 11km, distributed in a wide range of intensities, where 12% of the distance correspond to short max effort sprints and high intensity accelerations and decelerations (Chmura et al., 2018). All this represents high demand in musculoskeletal structures.

The work of injury prevention has been mostly relegated to team trainers and medical staff using multidimensional approaches based on risk factors and preventive strategies (Mylonas, Angelopoulos, Tsepis, Billis, & Fousekis, 2021). One dimension of this approach includes managing workloads (WL) (McLaren et al., 2018). External workloads make reference to factors such as totals sprints and total covered distance, all of which are measured with GPS devices (Oliveira et al., 2021). On the other hand, internal workloads (IL) make reference to factors such as heart rate and perceived exertion (sRPE) (Fernandes et al., 2021) . Internal and external workloads have shown to be highly correlated and the use of IL have been validated as a tool for quantifying intensity of a stimulus (Seshadri et al., 2020).

The sRPE can be used to quantify different types of training such as cardiorespiratory or resistance training (Impellizzeri, Rampinini, Coutts, Sassi, & Marcora, 2004). This feature is particularly useful considering that most schedules of professional players include many types of different trainings. In 2001 Foster et. al. (Haddad, Stylianides, Djaoui, Dellal, & Chamari, 2017) introduced an extension of the sRPE in order to improve the estimation of IL including not only the perceived exertion, but also the duration of the stimulus. In this sense, the arbitrary units were created to measure the total WL of a session of training.

$$UA = sRPE\ x\ Stimulis_{duration}$$

Furthermore, in the same work the author proposes another metric named Monotony in order to quantify the variability of the training sessions during a particular week. In later work many other metrics derived from the UA were introduced such as acute:chronic workload ratio (ACWR) and workload strain index (Maupin, Schram, Canetti, & Orr, 2020).

$$Monotony = \frac{WeeklyMean_{UA}}{WeeklySD_{UA}}$$

Since its introduction the increasing interest in the aforementioned metrics was mainly for two reasons: first they represent an easy and low cost heuristic to manage and plan intensity of training and secondly they were found to be associated with risk of suffering injuries (Maupin et al., 2020).

$$ACWR = \frac{Daily_{UA}}{Rollmean_{N_{days}}}$$

Currently this last topic has been in the spotlight because of conflicting results. Many authors such as (Enright, Green, Hay, & Malone, 2020) have stated that metrics quantifying IL like the ACWR have potential as predictors of injuries. On the other hand, the author (Impellizzeri et al., 2021) has been the principal retractor of this hypothesis in his work this author stated that the effects of the ACWR on the risk of suffering an injury are a product of an artifact of the statistical models. Beyond contradictory opinions from different authors most of the work done by now where association between IL and injury risk was found have been done in a descriptive framework, predictive work has been mostly done using external loads data probably because low availability of big enough dataset of IL sessions. In the present work we propose to develop a predictive framework based in IL using machine learning techniques.

1.2 Motivation

The present work was carried out as part of the development of a system to facilitate the storage, processing and use of data associated with the health, well-being and physical performance of professional soccer players.

The request for this development was carried out by the management and medical team of the Belgrano de Córdoba professional soccer team. Until the time of development, the only data that was stored was that associated with the training duration and Rating of Perceived Exertion of each training. After each training session, each player was entrusted with completing a form where said data was recorded. This spreadsheet was then loaded into a spreadsheet, which was used to plan the following week's workouts.

The planning of the training sessions followed a very simple heuristic which is based on preventing the average number of arbitrary units of the team from exceeding a certain threshold. This threshold is usually defined at the discretion, depending on the objective of the medical staff and the physical preparation staff of the club. This method has major disadvantages:

- It only allows decisions to be made based on a temporary threshold reduced to a window of one week. Due to the simplicity of the heuristics, the data that is used to make decisions about managing the future training load is restricted to 7 days.
- It does not allow the individual evaluation of the players. This heuristic is based on aggregate values at the team level, which makes it impossible to study the training load that each player receives and does not allow evaluating the adaptive response of each player to the training load. Nor does it allow studying how each player is positioned with the rest of the team.

In addition to these disadvantages, during preliminary meetings the concern of the medical and physical training staff about the incidence of injuries suffered by the players

was highlighted. For this reason, the need to evaluate and investigate the possibility of developing a model that allows quantifying a player's injury risk based on workload data.

As a general objective, the model should exploit the historical records of each player, as well as the relative position of a player with respect to the rest of the team, to create an output that quantifies the risk of injury within a reasonable time threshold, which allows carry out an intervention to avoid the possible injury of said player. Due to data limitations we decided to use only the historical workload data, and not data associated with physical and/or medical evaluations.

2. Methods

2.1 Dataset

The dataset is composed of ten years of sRPE and exposure time records of each training and match session of a professional football team from Argentina (Belgrano de Córdoba) this accounts for 80000 observations. In addition, epidemiological data of injuries and its aetiology of the corresponding ten years is available. Both sRPE and exposure time are self-reported, each player is responsible for uploading his score after every training session. Then, the medical staff checks the answers in order to find discrepancies, missing values or any kind of error that could arise. After the sanity check, data is uploaded to the database.

2.2 Features and feature engineering

Using sRPE exposure time and injury records of each player, 32 new features were created. Commonly used IL metrics were replicated from the literature including UA, monotony and strain. Rolling means with 4 different time windows were calculated for each metric (2-, 4-, 7- and 15-days windows). Session sRPE Score was created to compare the relative position of each player with the rest of the team in each particular training session, also as previously described rolling windows were calculated for this feature.

A player's RPE score for a given date is calculated as the difference between that player's RPE and the minimum RPE reported by the entire team divided by the difference between the team's maximum RPE for that date, minus the minimum for that same date.

Information of previous injuries was also included as a cumulative sum of previous lesions (PI). Descriptive statistics of all features for both groups (train and test) can be seen in the appendix section. A brief description of the after mentioned variables can be found in Table 1.

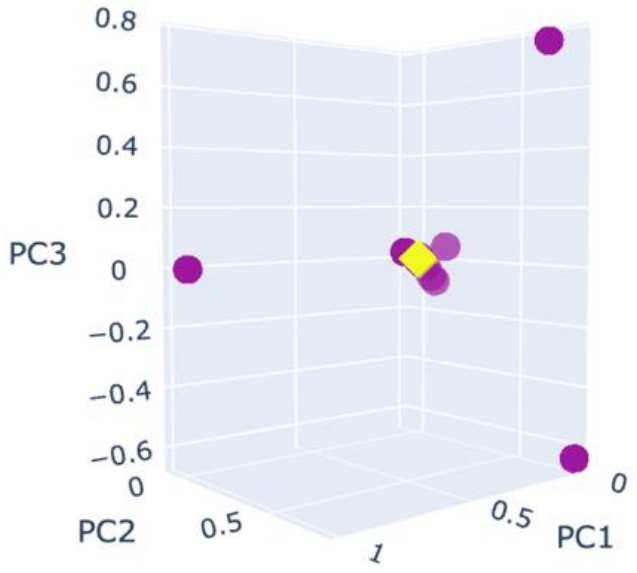
Table 1. Descriptive variables.

Var.	Description	Var.	Description
UA	Arbitrary Units	UA_cum2	Accumulated sum of last 2 days of UA
T	Exposure Time	UA_cum4	Accumulated sum of last 4 days of UA
sRPE	Rating of Perceived Exertion	UA_cum7	Accumulated sum of last 7 days of UA
RPE_r2	2 days rolling mean of RPE	UA_cum15	Accumulated sum of last 15 days of UA
RPE_r4	4 days rolling mean of RPE	mon_2	Monotony of last 2 days
RPE_r7	7 days rolling mean of RPE	mon_4	Monotony of last 4 days
RPE_r15	15 days rolling mean of RPE	mon_7	Monotony of last 7 days
T_r2	2 days rolling mean of T	mon_15	Monotony of last 15 days
T_r4	4 days rolling mean of T	str_2	Strain of the last 2 days
T_r7	7 days rolling mean of T	str_4	Strain of the last 4 days
T_r15	15 days rolling mean of T	str_7	Strain of the last 7 days
UA_r2	2 days rolling mean of UA	str_15	Strain of the last 15 days
UA_r4	4 days rolling mean of UA	PI	Number of previous Injs.
UA_r7	7 days rolling mean of UA	RPE_score	RPE score
UA_r15	15 days rolling mean of UA	RPE_score_r2	2 days rolling mean of RPE Score
UA_sdr2	Standard deviation of last 2 days of UA	RPE_score_r4	4 days rolling mean of RPE Score
UA_sdr4	Standard deviation of last 4 days of UA	RPE_score_r7	7 days rolling mean of RPE Score
UA_sdr7	Standard deviation of last 7 days of UA	RPE_score_r15	15 days rolling mean of RPE Score

Before creating all the new features from the two original ones, an analysis of correlation between them was made. As it would be expected for features created from the same source, high correlation was found. For this reason, we explored the use of principal component analysis (PCA) for pre-processing of the data before training the models.

PCA is a method used to reduce the number of variables in your data by extracting important ones from a large pool (Yata & Aoshima, 2010). It reduces the dimension of your data with the aim of retaining as much information as possible. In other words, this method combines highly correlated variables together to form a smaller number of an artificial set of variables which is called “principal components” that account for most variance in the data. For exploratory reasons, we reduce the number of features that describe our data through PCA, the result of the first 3 principal components (accounting for the 43% of the variance in the data) is shown below.

Figure 1. 3D Representation of PCA decomposition.

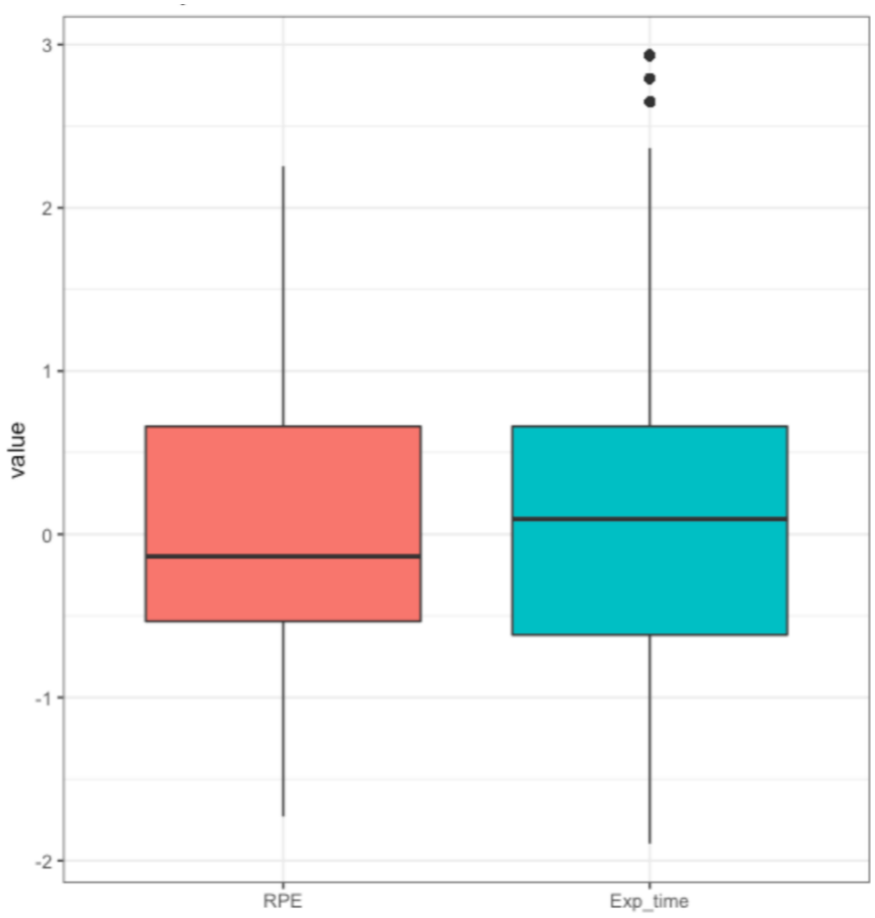


The observations in violet correspond to the negative class. Yellow rectangles are observations that correspond to positive classes. High overlapping between classes observed in the figure can be interpreted as the impossibility of linearly separating them through linear transformations, probably due to complex non-linear relations between features and target classes.

2.2.1 Missing values and outliers

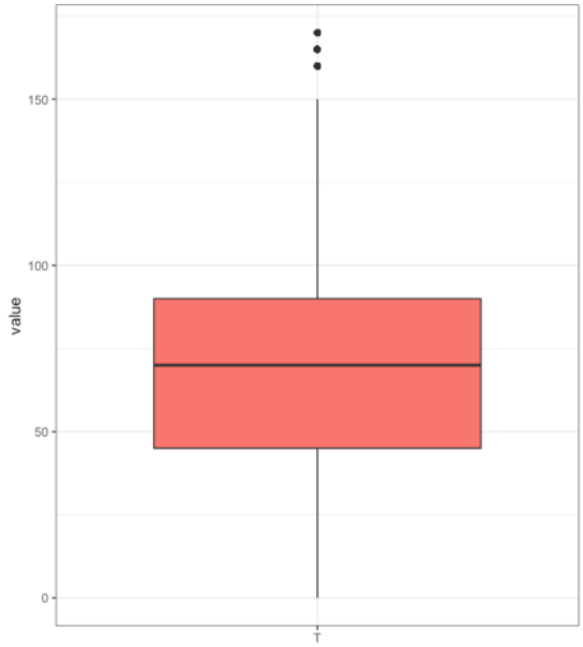
Due to the nature of the dataset, prior to its delivery for analysis, missing values and outliers were removed by the medical staff of the team. However, the distribution of the original variables was checked to guarantee the sanity of the information.

Figure 2. Box Plot of the scaled and centred RPE and exposure time



As we can see, there appears to be three extreme values in the variable exposure time. However as it can be seen below, they don't appear to be outliers, just particularly long sessions (time is expressed in minutes).

Figure 3. Box Plot of exposure time.



2.2.2 Exploratory Data Analysis.

An exploratory analysis of the data will be carried out in order to better understand the data set. One of the objectives will be to evaluate if there are differences in the distributions of the explanatory variables in relation to the target using different time windows until the event of interest, which is the injury of a player.

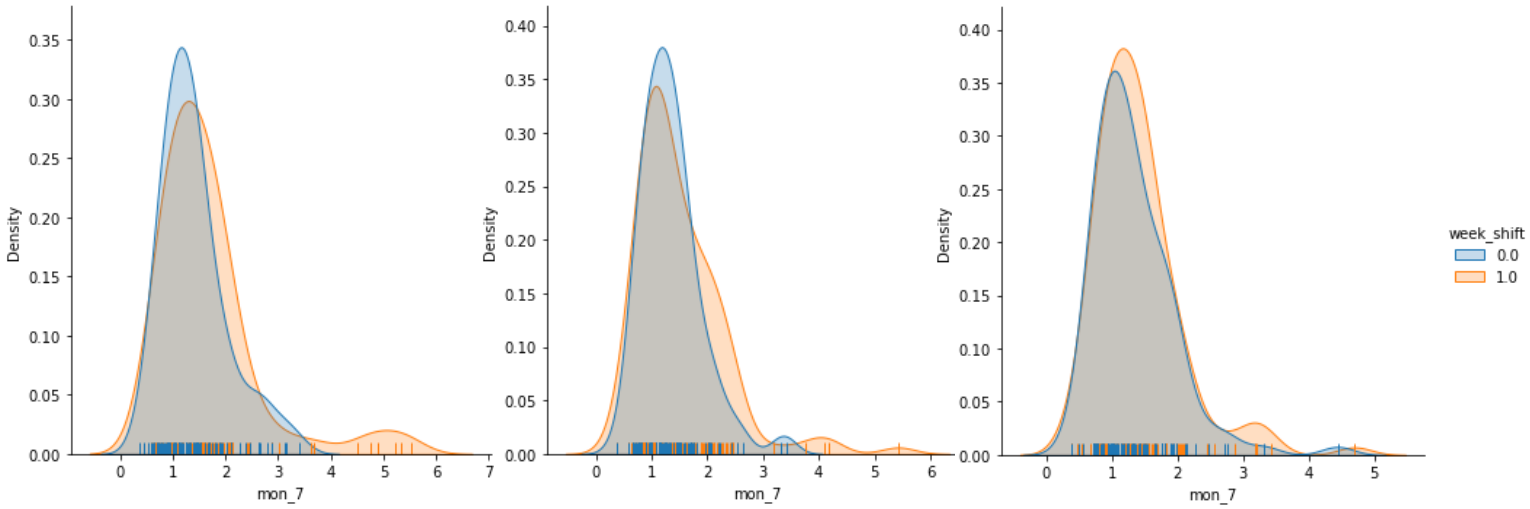
In order to evaluate how the distributions of the variables conditioned by the player's condition (injury or non-injury) behave, 3 different cut-off points were used, one at 24 hours, another at 72 hours and another at 120 hours (that is, if the player suffered an injury within the following 24/72/120 hours is assigned a value of 1). For that kernel density estimation plots were used.

During the analysis some interesting patterns emerged. In most cases, the distributions conditioned by the state of the player showed a large overlap, except for 5 variables whose distributions differed considerably when conditioned by the outcome of interest. For these, in addition to the kernel density graphs, ECDF graphs were included to appreciate in more detail how the distributions diverge when conditioned. Another interesting pattern found was that this difference between distributions is only present for the cut-off points of 24 and 72 hours, being lost for the cut-off point of 120 hours. The variables in which the aforementioned patterns were found are shown in detail below.

2.2.2.1 Monotony

Recall that monotony is defined as the mean daily workload divided by the standard deviation of the daily workload. In our case, we calculate this variable from the exposure time and daily RPE data of the players. We then calculated rolling means of this variable using different time periods. During our analysis, we found that the distributions that differed the most when conditioned were those with a 7- rolling mean.

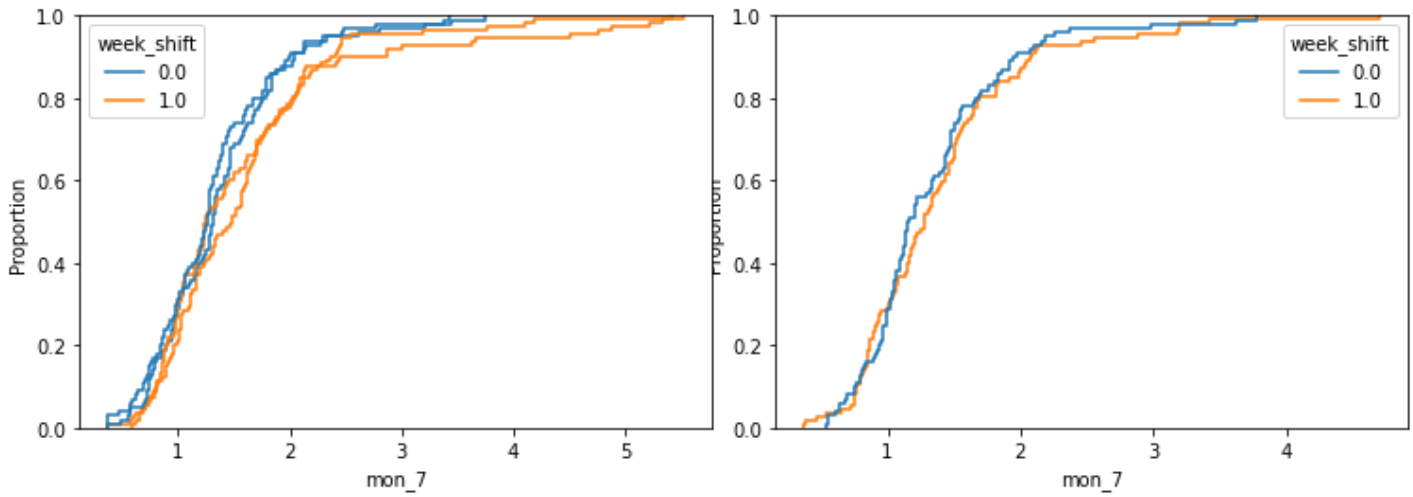
Figure 4. KDE for monotony Rolling mean of 7 days. From Left to Right, 24/72/120 hs thresholds are shown.



In the previous graphs, the aforementioned pattern can be observed, and it can be seen how the distributions become more overlapping as we move away from the event of interest.

To better appreciate how the distributions differ, ECDF plots were used, which are shown below. The 24/72h cut-off points are shown together and the 120h cut-off point is shown separately.

Figure 5. ECDF for monotony Rolling mean of 7 days.



2.2.2.2 Strain

Recall that the strain is defined as the monotony multiplied by the workload. As in the case of monotony, this variable was calculated from the exposure time and RPE data of the players, and moving averages with different periods were also calculated, to later be incorporated into the models.
In this case, the distributions of the rolling means of 2, 4 and 7 days showed differences when conditioned by the outcome.

Figure 6. KDE for Strain Rolling mean of 4 days. From Left to Right, 24/72/120 hs thresholds are shown.

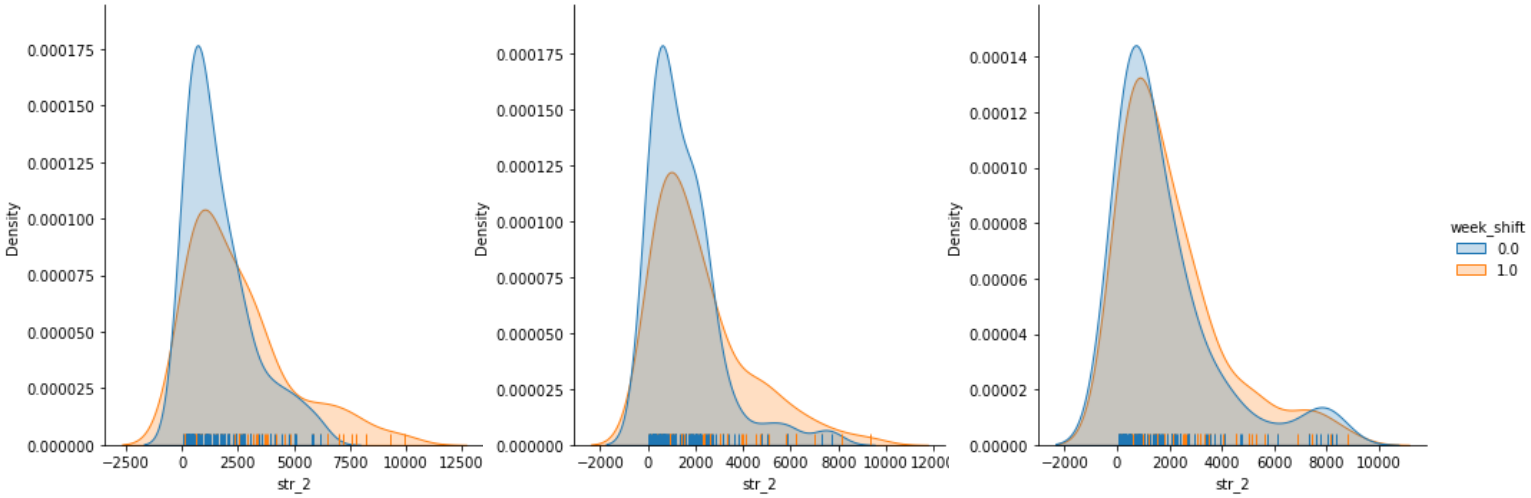


Figure 7. KDE for Strain Rolling mean of 4 days. From Left to Right, 24/72/120 hs thresholds are shown.

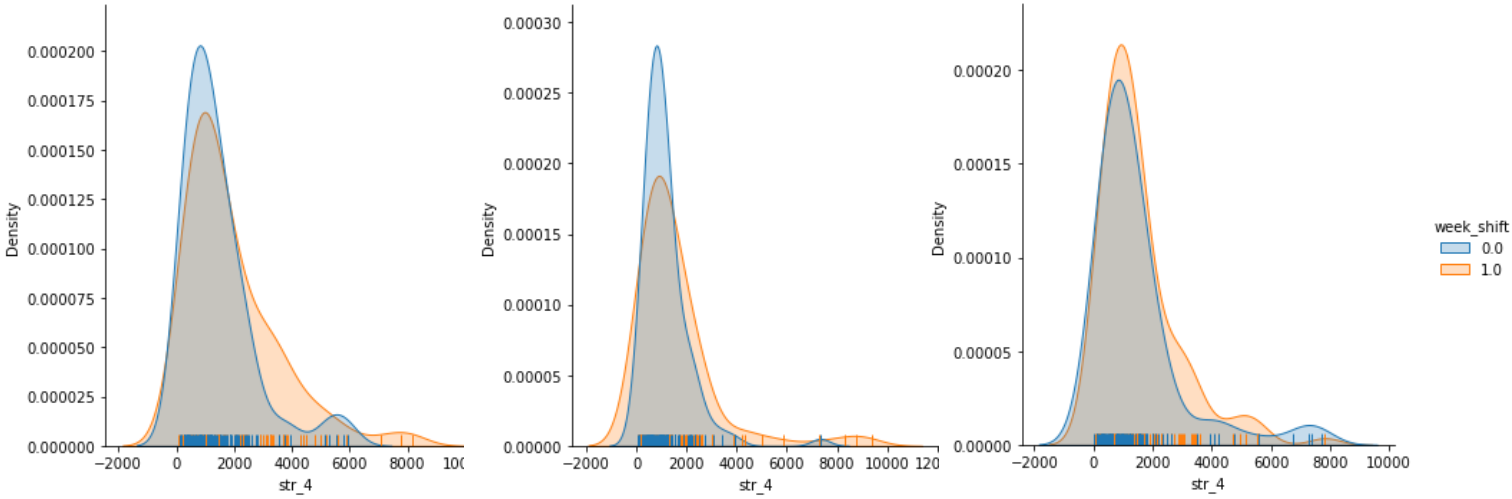
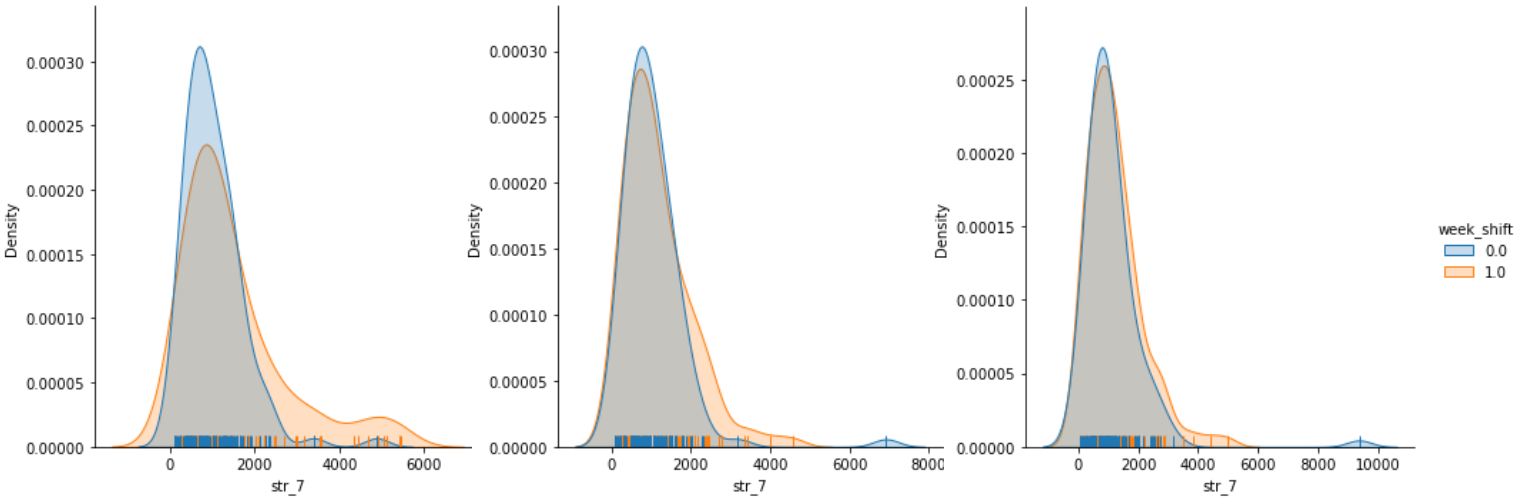


Figure 7. KDE for Strain Rolling mean of 7 days. From Left to Right, 24/72/120 hs thresholds are shown.



As in the case of monotony, we can see that the distribution of the variables differs when conditioned and this difference disappears as we move away from the time of injury.

ECDF graphs were incorporated to better appreciate how the distributions of the variables differ. The 24/72h cut-off points are shown together and the 120h cut-off point is shown separately.

Figure 8. ECDF for Strain Rolling mean of 2 days.

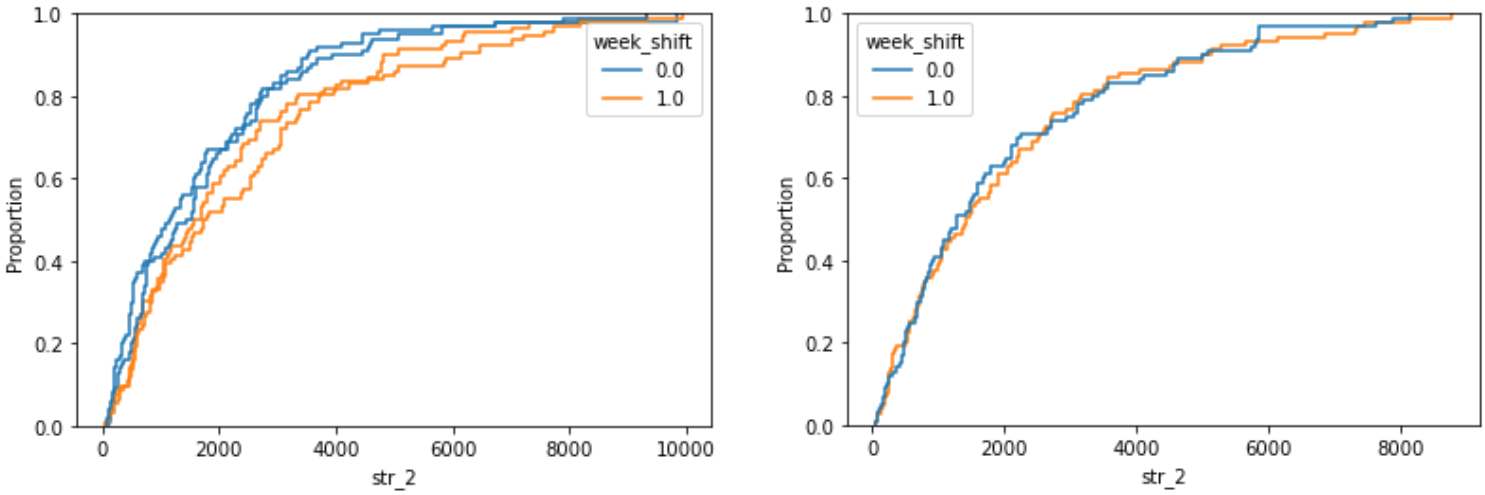


Figure 9. ECDF for Strain Rolling mean of 4 days.

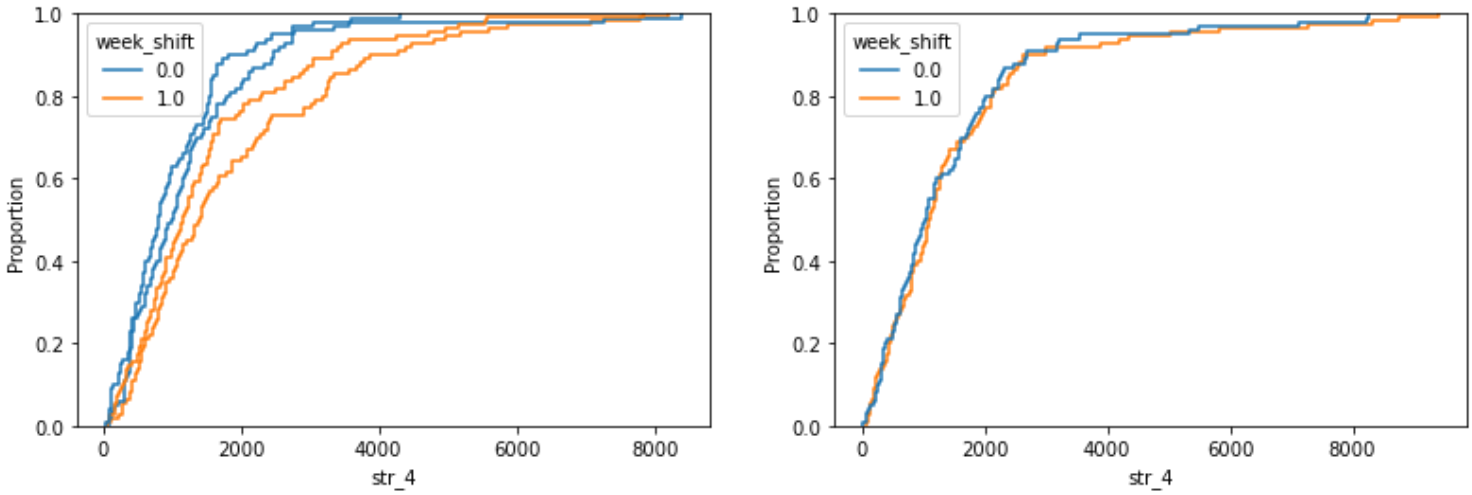
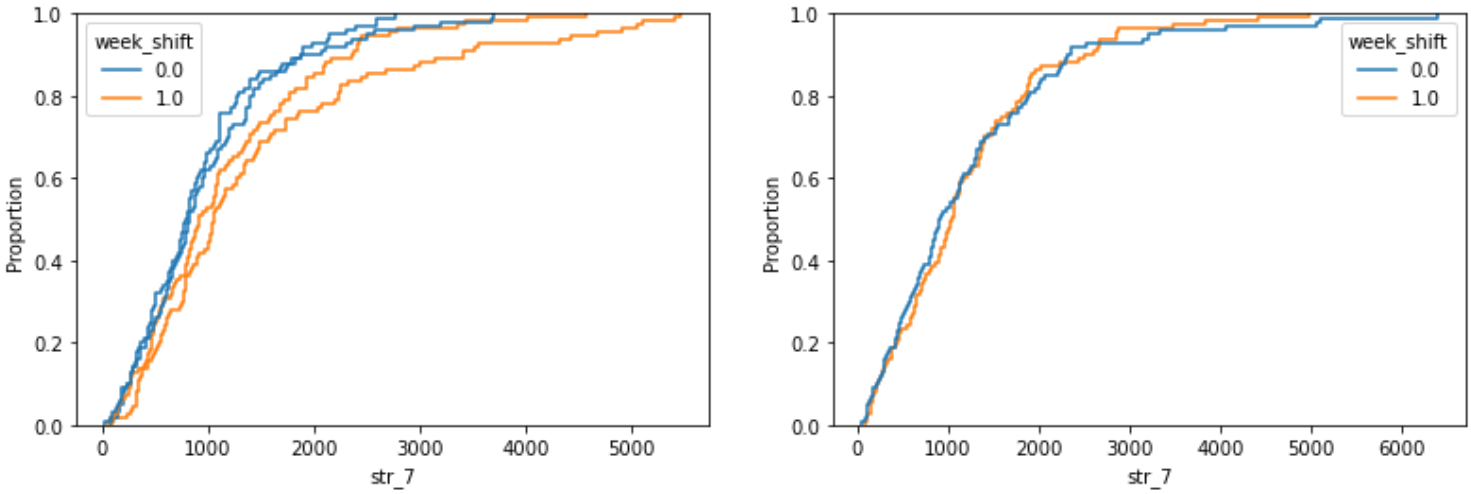


Figure 10. ECDF for Strain Rolling mean of 7 days.



Observing the ECDF graphs, it can be seen how the conditioned distributions for the 24 and 72h thresholds differ according to the group to which they belong, while in the case of the 120h threshold, an almost perfect overlap between the distributions is shown.

2.2.2.3 RPE score

The RPE score variable is a feature derived in this project of which no reference was found in any previous academic work. The main objective of this variable is to incorporate information about the relative position of a player with respect to the rest of the team for a given date. Like the rest of the variables, rolling means were calculated to incorporate historical information on it. The distribution of this variable was the one that differed most strongly when it was conditioned by the player's status (injured / not injured).

Figure 11. KDE for RPE score. From Left to Right, 24/72/120 hs thresholds are shown.

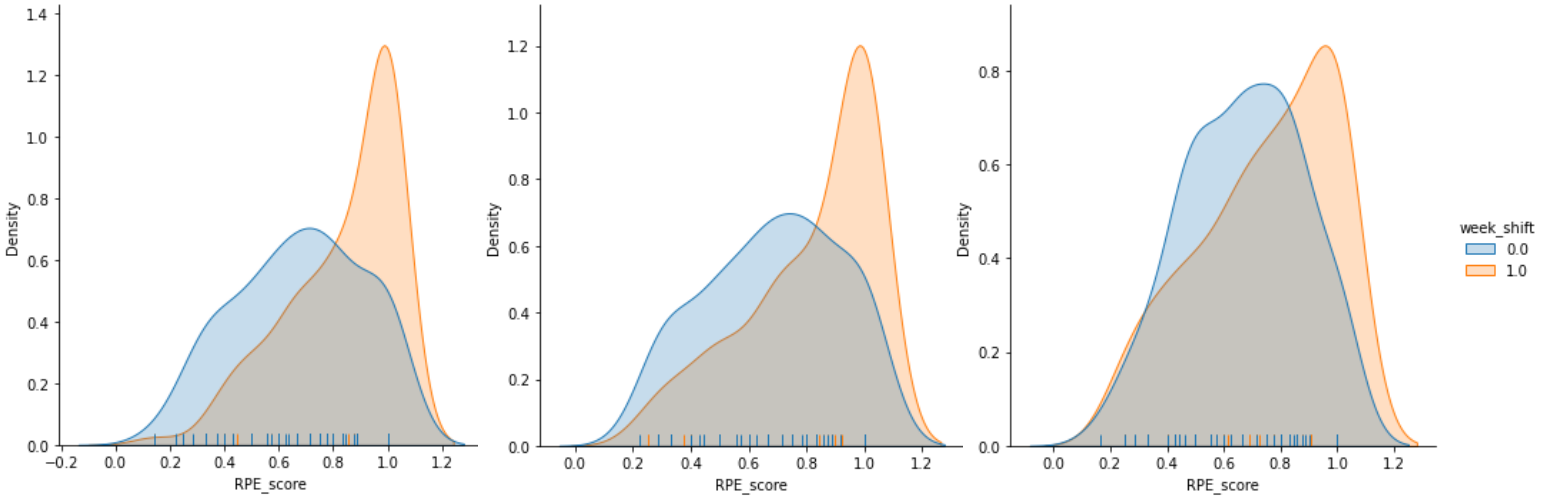


Figure 12. KDE for RPE score Rolling Mean of 2 days . From Left to Right, 24/72/120 hs thresholds are shown.

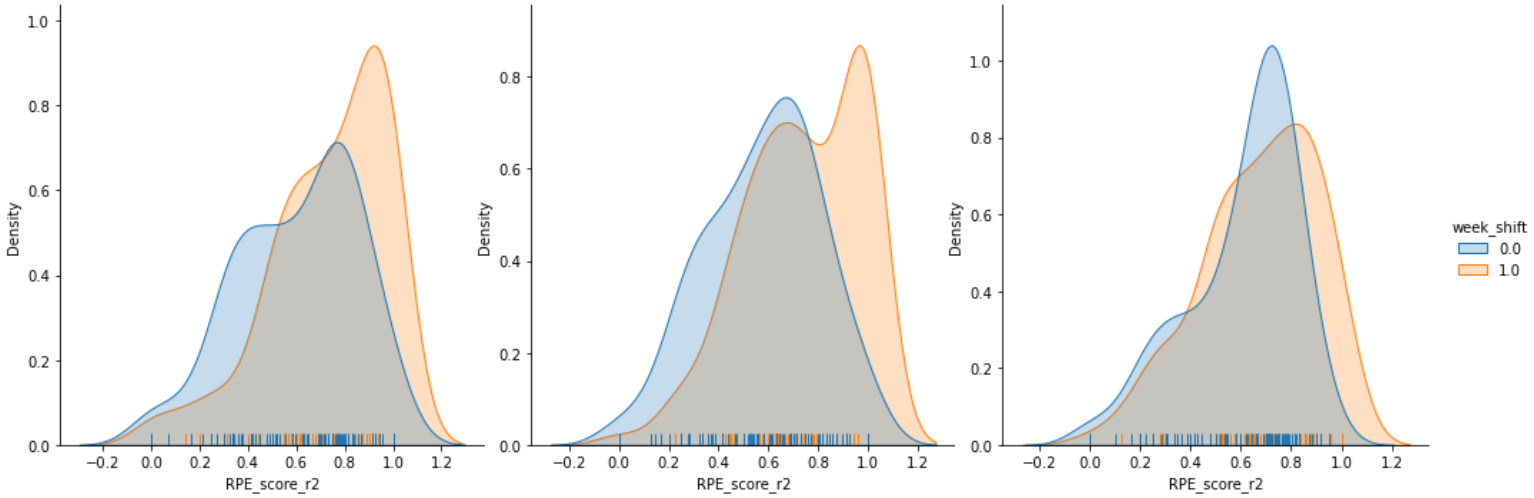
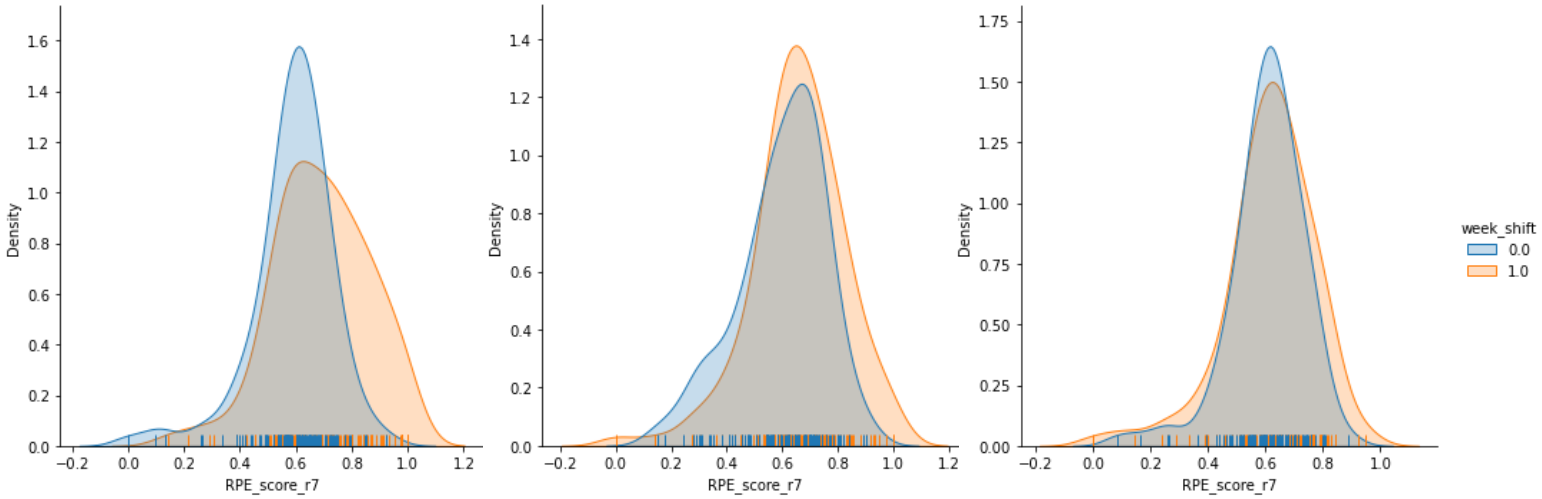


Figure 13. KDE for RPE score Rolling Mean of 7 days . From Left to Right, 24/72/120 hs thresholds are shown.



When looking at the ECDF it is remarkable how different the distributions are between them. As in the previous cases, the 24/72h cut-off points are shown together and the 120h cut-off point is shown separately.

Figure 14. ECDF for RPE score.

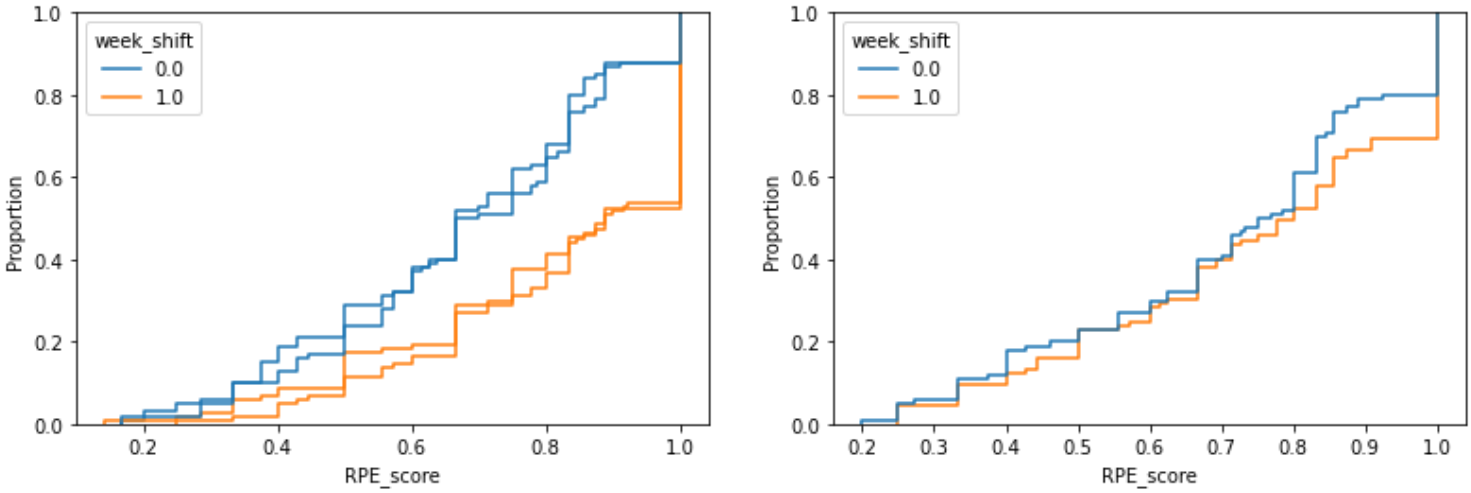
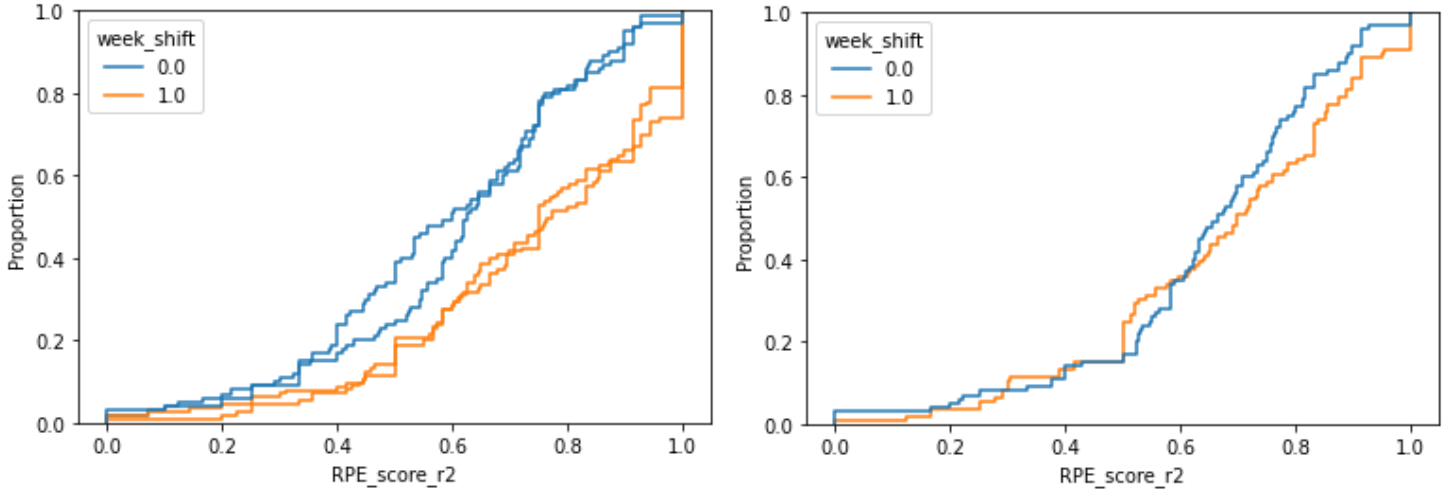


Figure 15. ECDF for RPE score Rolling mean of 2 days.



During the exploratory analysis, no outliers, null values or atypical data were found, which is a good sign, since it indicates that the functions that were built to create the variables work correctly and that the previous curation of the database, by the medical staff, was correct

2.2.3 Target Imbalance

Generally speaking, it is usual and expected that a database exhibits a different distribution between classes, but when we talk about a database being unbalanced, we refer to the fact that there is a significant, and often extreme, difference between classes

(He & Shen, n.d.). This type of imbalance is often referred to as between-class imbalance. Although in our case we refer to an imbalance between two classes, it can easily be extended to multiclass classification problems.

This problem of class imbalance usually arises frequently in the field of biomedicine (Woods et al., 1994). It is usual for a collection of data, with certain characteristics, to have the objective of classifying them into two groups, "healthy" and "pathological". This is usually due to the origin of the data, these data normally come from preventive studies, which are indicated to individuals whose probability of suffering from a certain pathology is low. This happens, for example, in the cases of breast screening, where it is sought to detect different types of breast pathologies early. This makes it expected that negative cases (healthy individuals) exceed positive cases (pathological individuals).
Of course, our objective is to train a model that provides a balanced degree of precision for both classes, both the majority (negative in this case) and the minority (positive in this case). In practice we usually find that the models tend to provide highly unbalanced predictions, in favour of the majority class, this is a consequence of the loss functions that the models use internally to optimize their parameters. This usually represents a serious problem in these cases, since the relative cost of false negatives is usually high, and many times much more expensive than obtaining a false positive (Z. H. Zhou & Liu, 2005).

Furthermore, all this suggests that the model performance evaluation practices using general metrics such as overall accuracy and error rate do not provide adequate information (Gary M. Weiss, 2004), because, in general, a model that has a bias towards the majority class (Joshi, Kumar, & Agarwal, 2001), will tend to have high values of overall accuracy and low of error rate, even if it is not able to identify observations of the minority class. Due to all this, it is advisable in these cases to use metrics that provide more information, such as the receiver operating characteristics curve, precision-recall curves and cost curves (Japkowicz, 2013). The type of imbalance that we have discussed so far is often called intrinsic, since it depends on the nature of the data. There is another type of imbalance, called extrinsic, which is usually the product of factors such as lack of data collection or lack of a type of class due to externalities associated with data storage (Y. Liu, Wang, Ren, Zhou, & Diao, 2019).

As we have discussed so far, class imbalance is usually a frequent pattern found in different types of applications in practice, and because of this it has been in the focus of interest of multiple researchers (Haixiang et al., 2017). Some studies have shown that for some types of class imbalance, the minority concept is accurately learned with little disturbances from the imbalance (Batista, Prati, & Monard, 2004). These results are particularly interesting, since they suggest that the simple fact of class imbalance is not the only reason why the models are an impediment to learning the rules that allow correct classifying the observations. From this arises the hypothesis that the ability of the models to generate correct predictions depends on the complexity of the data, and the relationships that determine an outcome (Japkowicz & Stephen, 2002). In this way, understanding the nature of the phenomenon that generated the data would allow us to understand why in some cases the performance of the models is severely limited. In our case, when studying a phenomenon that arises from the interaction of a complex system with the environment, understanding this last premise is essential to be able to correctly address the problem and understand possible limitations of the data modelling results (Bittencourt et al., 2016).

2.2.2.1 Consequences of target imbalance.

Taking into account everything mentioned so far, and not minimizing the fact that the simple imbalance is not the only factor that inclines the models towards a particular class, it is generally accepted that when standard models are used on unbalanced data, the amount of inductive rules that describe the minority class are usually less and more diffusely learned, compared to the rules that describe the majority class (Chawla, 2003).

This can be clearly seen in decision tree models. In these, the imbalances exploit the inadequacies in the splitting criterion in each node of the decision tree (G M Weiss & Provost, 2003). Decision trees use a recursive, top-down greedy search algorithm that uses a feature selection scheme to select the best features as the split criterion at each node of the tree; a successor (the leaf) is then created for each of the possible values corresponding to the split feature. As a result, the training set is successively partitioned into smaller subsets that are ultimately used to form disjoint rules related with the class concepts. These rules are finally combined so that the final hypothesis minimises the total error rate across each class. The problem of this procedure in the presence of imbalanced datasets has two parts. First, successive partitioning of the dataspace results in fewer observations of the minority class examples, resulting in fewer leaves describing minority concepts and with weaker confidence estimates. Second, concepts that have dependencies on different features space conjunctions can go unlearned by the spareness introduced through partitioning.

2.2.2.2 Solutions for target imbalance

Typically, the use of sampling methods in imbalanced learning applications consist of the modification of an imbalanced dataset by some mechanism in order to provide a balanced distribution (Laurikkala, 2001).

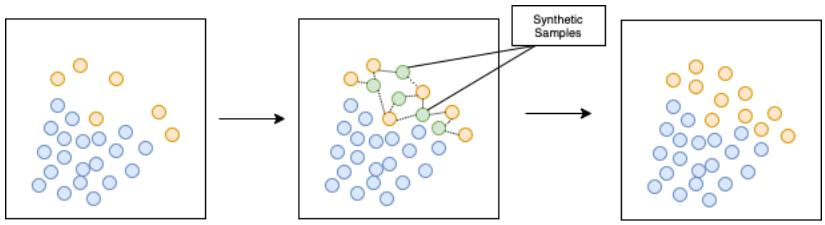
Random oversampling is a mechanism by which we add a set of observation sampled from the minority class (Moreo, Esuli, & Sebastiani, 2016), for a set of randomly selected minority examples, we augment the original dataset by replicating the selected examples and adding them to the original dataset. In this way, the number of total minority examples randomly chosen can be learned and adjusted. This provides a mechanism for varying the degree of class distribution balance to any desired level.

While oversampling appends data to the original dataset, random under sampling (X.-Y. Liu, Wu, & Zhou, 2009) removes data from the original dataset. In this case we randomly select a set of majority class examples and remove these from the original dataset. The total number of removed samples has to be chosen as in the previous case we can vary the number in order to get a desired balance in our dataset.

Moving forward from resample techniques, synthetic data generation is another way to get a desired balance between classes. In particular synthetic minority oversample technique (SMOTE) is a powerful method that has shown great deal of success in various applications (Raghuwanshi & Shukla, 2020). The algorithm works as follow: for the subset of minority class examples, consider the K-nearest neighbours for each example included in the subset of the minority class, for some specified integer K; the K-nearest

neighbours are defined as the K elements of the minority subset whose Euclidean distance between itself and the examples in consideration exhibit the smallest magnitude along the n-dimensions of the feature space. To create a synthetic sample, randomly select one of the K-nearest neighbours, then multiply the corresponding feature vector difference with a random number between [0,1], and finally add this vector to the chosen neighbours. As in the resample techniques, the total number of synthetic examples can be chosen.

Figure 16. Graphic representation of SMOTE



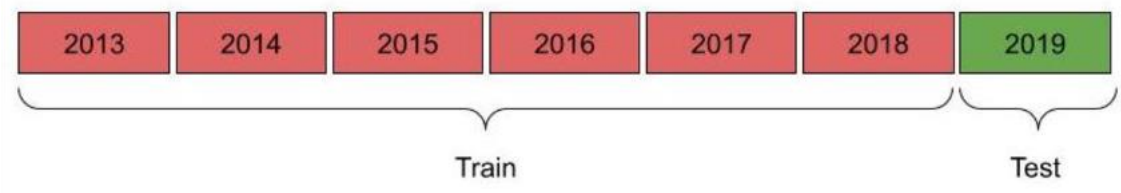
While sampling methods attempt to balance distributions by considering the representative proportions of class examples in the distribution, cost-sensitive learning methods consider the costs associated with misclassifying examples (Thai-Nghe, Gantner, & Schmidt-Thieme, 2010). Instead of creating balanced data distributions through different sampling strategies, cost sensitive learning targets the imbalanced learning problem by using different cost matrices that describe the costs for misclassifying any particular data example. Fundamental to the cost sensitive learning methodology is the concept of the cost matrix. The cost matrix can be considered as a numerical representation of the penalty of classifying examples from one class to another. The objective of cost sensitive learning then is to develop a hypothesis that minimizes the overall cost on the training data set. We were not able to test this method because there was no clear way to determine the cost matrix. The main issues will be described in section 5.

In our case, we are dealing with an extremely imbalanced dataset, only ~ 0.5% of the observations correspond to positive class. Oversample and SMOTE were tested in order to improve the performance of the models. Oversample showed to be the best approach in our case. Model performance using SMOTE in the testing set is shown in the appendix.

2.2.3 Modelling approach

The models were trained using 6 consecutive years (2013, 2014, 2015, 2016, 2017, 2018) and were tested in one year (2019). Due to the temporal distance between the end of one season and the next, it was not necessary to take precautions in the possible overlap between the last year of training (2018) and the year used as test (2019).

Figure 17. Data split for train and test



In order to optimize the hyperparameters of the models, a random search and a cross-validation scheme were used on the training data as indicated in (Bergmeir, Hyndman, & Koo, 2018). Then, the best model with its respective hyperparameters was trained on the entire training set, and then validated with the test data. In order to assess the uncertainty of evaluation metrics, bootstrapping was used. Bootstrapping is a resampling technique by which many new data sets are created by resampling the original dataset (Adler & Lausen, 2009). Then, every new dataset is pre-processed, divided in train and test, model parameters are estimated and then tested using different evaluation metrics, generating a distribution of parameter and evaluation metrics. Using the resulting empirical distribution of evaluation metrics, we can assess the uncertainty of them. Also, these distributions are used to compare different models' performances. An analysis of the robustness of the estimation of the hyperparameters is included in section 6.6 of the appendix.

The models were built with the objective of predicting injuries with a time window of 72 hours in the future from the observation under consideration, using the historical data of a player's IL at a given moment in time. Only non-contact injuries that generated training or match time loss were taken into consideration. The 72-hour threshold was chosen for two reasons, the first is that during the period of experimentation and evaluation of better models, we evaluated how the predictability of the lesions changed at different time thresholds, where it remained constant for the thresholds of 24, 48 and 72 hours respectively and then it falls precipitously. The second reason is related to the time it takes to take a course of action in the application of the model, the further away the event is from occurring, the more time is given to the medical staff to take action with a certain player. The results of these experiments can be seen in the appendix.

2.2.4 Models

Multiple models were used to fit the training data and were then tested against the validation data. All of them were trained with all the available features. Due to the complex nature of the interactions that determine the predisposition to injury, models with different flexibility to adjust to the data were evaluated, which included:

- Decision Tree Classifier (DT): tree-based methods partition the feature space into a set of rectangles, and then fit a simple model like a constant in each one. Roughly speaking, there are two steps to build a DT (Trevor Hastie, Robert Tibshirani, & Jerome Friedman, 2016). First, we divide the predictor space into K distinct and non-overlapping regions. Second, for every observation that falls into a given region, we make the same prediction, which is simply the most commonly occurring class for the training observations in that region. The first question that arises from this idea, is how do we construct the regions? The answer is quite simple, we have to test every single possible partitioning, and keep the one that minimises a given loss function. However, this is computationally impossible. For this reason, we take a top-down, greedy approach that is known as recursive binary splitting. The approach is top-down because it begins at the top of the tree (at which point all observations belong to a single region) and then successively splits the predictor space; each split is indicated via two new branches further down on the tree. It is greedy because at each step of the tree-building process, the best split is made at that particular step, rather than looking ahead and picking

a split that will lead to a better tree in some future step. In order to perform recursive binary splitting, we first select a predictor and a cut point such that splitting the predictor space into the regions leads to the greatest possible reduction in the loss function. Next, we repeat the process, looking for the best predictor and best cut point in order to split the data further so as to minimize the loss within each of the resulting regions. However, this time, instead of splitting the entire predictor space, we split one of the two previously identified regions. This process continues until a stopping criterion is reached. There are two classical loss functions that are used in classification problems. The first one is the Gini index, a measure of total variance across the all-response classes, commonly referred to as a measure of node purity; a small value indicates that a node contains predominantly observations from a single class. And the second one is the entropy. When building a classification tree, either the Gini index or the entropy are typically used to evaluate the quality of a particular split, since these two approaches are very sensitive to node purity.

- Logistic Regression (LR): Is a transformation of a linear regression using a sigmoid function. Despite its name, it is a classification model rather than a regression model (Trevor Hastie et al., 2016). Logistic regression is a simple and more efficient method for binary and linear classification problems. It is a classification model, which is very easy to realize and achieves very good performance with linearly separable classes. We are going to estimate model parameters through a general method called maximum likelihood. The basic intuition behind using maximum likelihood to fit a logistic regression model is as follows: we seek estimates for the parameters such that the predicted probability of injury for each individual, corresponds as closely as possible to the individual's observed injury status. Regularization is a technique used to prevent overfitting problems (Ng, 2004). It adds a regularization term to the maximum likelihood equation. Regularization encompasses techniques that are used to avoid overfitting. In our case due to the high dimensionality of the input and high multicollinearity between variables, we use L2 regularization.

- Random Forest (RF): Bootstrap idea was already described previously, we will see here that the bootstrap can be used in a completely different context, in order to improve statistical learning (Trevor Hastie et al., 2016). Bootstrap aggregation, is a general-purpose procedure for reducing the variance of a statistical learning method. Recall that given a set of independent $\boldsymbol{n}$ observations, each with variance $\boldsymbol{V}$, the variance of the mean of the observations is given by $\boldsymbol{V/n}$. In other words, averaging a set of observations reduces variance. Hence a natural way to reduce the variance and hence increase the prediction accuracy of a statistical learning method is to create many training sets through bootstrapping, build a separate prediction model using each bootstrapped training set, and average the resulting predictions. RF models use this powerful idea, and introduce another tweak in order to decorrelate the created trees. We build a number of decision trees on bootstrapped training samples, but when building these decision trees, each time a split in a tree is considered, a random sample of predictors is chosen as split candidates from the full set of $\boldsymbol{p}$ predictors. The split is allowed to use only one of those predictors. A fresh sample of predictors is taken at each split. In other words, when building a random forest, at each split in the tree, the algorithm is not even allowed to consider a majority of the available predictors.

- Gradient Boosting Machine (GBM): boosting is a general approach that can be applied to many statistical learning methods for regression or classification (Garreth Jamws, Daniela Witten, Trevor Hastie, & Robert Tibshirani, 2021). Here we restrict our discussion of boosting to the context of decision trees. As in RF, in boosting we will build many trees, but in boosting, trees are grown sequentially, each tree is grown using information from previously grown trees. Boosting involves combining a large number of decision trees, but unlike fitting a single large decision tree to the data, which amounts to fitting the data hard and potentially overfitting, the boosting approach instead learns slowly. Given the current model, we fit a decision tree to the residuals from the model, that is, we fit a tree using the current residuals, rather than the outcome. We then add this new decision tree into the fitted function in order to update the residuals. Each of these trees can be rather small, with just a few terminal nodes. By fitting small trees to the residuals, we slowly improve predictions in areas where the model does not perform well. The GBM model uses this idea of boosting, but it identifies the shortcomings of weak learners by using gradients in the loss function.

Due to the high correlation between features created in previous steps, two experimental settings were created. In the first one the raw features were used as input for the model training and testing. In the second approach, the PCA method was applied over predictive features. PCA allows a huge amount of information enclosed in initially correlated data to be transformed into a set of new orthogonal components, thereby making it possible to discover concealed relationships, enhance data visualization, detection of outliers, and classification within the newly defined dimensions (Khalid, Khalil, & Nasreen, 2014). In some cases, the applications of PCA on a dataset as a pre-processing can improve the performance of learning methods (Moghaddasi, Jalab, Md Noor, & Aghabozorgi, 2014). However, in our case best performance was achieved without using PCA (all principal components were used for this experiment). In this way, five different experiments were tested; in the first one, random oversampling without PCA was tested; in the second one random oversampling with PCA was tested; in the third one SMOTE without PCA was tested; in the fourth SMOTE with PCA was tested; in the fifth one undersampling was applied. The first setting was the best one, and its results are shown in detail in section four. The other results are shown in the appendix.

2.2.5 Model Evaluation

Traditionally, the most frequently used metrics are accuracy and error rate. Considering a basic two class classification problem, then a representation of classification performance can be formulated by a confusion matrix. In our case the minority class is represented as the positive class. Following this idea, accuracy and error rate are defined as:

$$Accuracy = \frac{TP + TN}{N}$$

$$ErrorRate = 1 - Accuracy$$

Where ***TP*** are true positives, ***TN*** are true negatives, and ***N*** are the total predictions made. These metrics provide a simple way of describing a classifier's performance on a given

data set. However, they can be deceiving in certain situations and are highly sensitive to changes in data. In the simplest situation, if a given data set includes 5 percent of minority class examples and 95 percent of majority examples, a naive approach of classifying every example to be a majority class example would provide an accuracy of 95 percent. A classifier that has an accuracy of 95 in the entire dataset sounds superb; however, this same classifier has a 0 percent accuracy in the minority class. In lieu of accuracy, other evaluation metrics are frequently adopted in the research community to provide comprehensive assessments of imbalanced learning problems, namely, precision, recall and F1 Score, defined as:

$$Precision = \frac{TP}{(TP + FP)}$$

$$Recall = \frac{TP}{(TP + FN)}$$

$$F_1 = 2 * \left[\frac{(recall * precision)}{(recall + precision)}\right]$$

Where ***FP*** are false positive predictions and ***FN*** are false negative predictions. Intuitively, precision is a measure of exactness (of the examples labelled as positive, how many are actually labelled correctly), whereas recall is a measure of completeness (how many examples of the positive class were labelled correctly). These two metrics, much like accuracy and error, share an inverse relationship between each other. However, unlike accuracy and error, precision and recall are not both sensitive to changes in data distributions. A quick inspection on the precision and recall formulas readily yields that precision is sensitive to data distributions, while recall is not. On the other hand, that recall is not distribution dependent is almost superfluous because an assertion based solely on recall is equivocal, since recall provides no insight to how many examples are incorrectly labelled as positive. Similarly, precision cannot assert how many positive examples are labelled incorrectly. Specifically, the F1 Score combines precision and recall as a measure of the effectiveness of classification in terms of a ratio of the recall and precision. As a result, F1 Score provides more insight into the functionality of a classifier than the accuracy metric, however remaining sensitive to data distributions.

In order to overcome the sensibility of an imbalanced dataset of the aforementioned metrics, we will introduce the receiver operating characteristics (ROC) curves. The ROC assessment technique makes use of the proportion of two evaluation metrics, namely, true positives rate (TP rate) and false positives rate (FP rate), which are defined as:

$$TP_{rate} = \frac{TP}{Pc}$$

$$FP_{rate} = \frac{FP}{N_c}$$

Where ***Pc*** are observed positive class examples, and ***Nc*** are observed negative class examples. The ROC graph is formed by plotting ***TP*** rate over ***FP*** rate, and any point in ROC space corresponds to the performance of a single classifier on a given distribution. The ROC curve is useful because it provides a visual representation of the relative trade-offs between the benefits (reflected by true positives) and costs (reflected by false

positives) of classification in regards to data distributions. Generally speaking, for the case of soft-type classifiers (classifiers that output a continuous numeric value to represent the confidence of an instance belonging to the predicted class) a threshold can be used to produce a series of points in ROC space. This technique can generate an ROC curve. In order to assess different classifiers performance, one generally uses the area under the curve (AUC) as an evaluation criterion. For instance, a classifier that has a higher AUC is better than a classifier with a lower AUC. Of course, one should also note that it is possible for a high AUC classifier to perform worse in a specific region in ROC space than a low AUC classifier.

2.3 Interpretable Machine Learning

Interpretability is the degree to which a human can understand the cause of a decision. The higher the interpretability of a machine learning model, the easier it is for someone to comprehend why certain decisions or predictions have been made (Christoph Molnar, 2020). A model is better interpretable than another model if its decisions are easier for a human to comprehend than decisions from the other model. We will differentiate explicability from interpretability in the sense that explicability will refer as the ability to explain a particular prediction.

In our case model interpretability will be quite important. As it will be seen in the next sections, we are not only interested in a powerful model to predict injuries, we also want to use the output of the model for many other applications (such as renegotiation of contracts), and because of this, we need understand the "Why" of the predictions. When we are in this kind of setting, social acceptance of the output model and transparency are important milestones. When we talk about interpretability, we can use different criteria whether interpretability is achieved by restricting the complexity of the machine learning model (intrinsic) or by applying methods that analyse the model after training (post hoc) (Serg Masís, 2021). Intrinsic interpretability refers to machine learning models that are considered interpretable due to their simple structure, such as short decision trees or sparse linear models. Post hoc interpretability refers to the application of interpretation methods after model training. Then it is also important to differentiate if methods used to interpret the model are model specific or model agnostic (Du, Liu, & Hu, 2019). Model specific interpretation tools are limited to specific model classes. The interpretation of regression weights in a linear model is a model specific interpretation. Model agnostic tools can be used on any machine learning model and are applied after the model has been trained (post hoc). These agnostic methods usually work by analysing feature input and output pairs. By definition, these methods cannot have access to model internals such as weights or structural information.

Another issue to address when we talk about interpretability, is whether the interpretation method explains an individual prediction or the entire model behaviour. You could describe a model as interpretable if you can comprehend the entire model at once (Christoph Molnar, 2020). To explain the global model output, you need the trained model, knowledge of the algorithm and the data. This level of interpretability is about understanding how the model makes decisions, based on a holistic view of its features and each of the learned components such as weights, other parameters, and structures (Doshi-Velez & Kim, 2017), which features are important and what kind of interactions between them take place. Global model interpretability helps to understand the

distribution of your target outcome based on the features. However, global model interpretability is very difficult to achieve in practice. Any model that exceeds a handful of parameters or weights is unlikely to fit into the short-term memory of the average human. On the other hand, you can change this approach and make a kind of zoom in on a single instance and examine what the model predicts for this input, and explain why (Murdoch, Singh, Kumbier, Abbasi-Asl, & Yu, 2019). If you look at an individual prediction, the behaviour of the otherwise complex model might behave more pleasantly. Locally, the prediction might only depend linearly or monotonically on some features, rather than having a complex dependence on them.

2.3.1 Model Agnostic Methods

Separating the explanations from the machine learning model has some advantages. The great advantage of model-agnostic interpretation methods over model-specific ones is their flexibility (Christoph Molnar, 2020). Machine learning developers are free to use any machine learning model they like when the interpretation methods can be applied to any model. Anything that builds on an interpretation of a machine learning model, such as a graphic or user interface, also becomes independent of the underlying machine learning model. Typically, not just one, but many types of machine learning models are evaluated to solve a task, and when comparing models in terms of interpretability, it is easier to work with model-agnostic explanations, because the same method can be used for any type of model. Desirable aspects of a model-agnostic explanation system are model flexibility (methods that work with any kind of model), explanation flexibility, representation flexibility.

2.3.1.1 Permutation Feature Importance

Permutation Feature Importance measures the increase in model prediction error after permuting the feature values, which breaks the relationship between the feature and the actual output.
The concept is relatively simple: the importance of a feature should be measured by calculating the increase in the prediction error of the model after exchanging said feature. A feature is important in terms that if we mix its values, it increases the error of the model, because in this case the model was based on the feature for the prediction.
A feature is not important if, by mixing its values, the model error remains invariant, so in this case the model ignored the feature for the prediction.

2.3.1.2 Shapley Values

By definition, the shapley value is the average marginal contribution of a feature value across all possible coalitions. The Shapley value, comes from an idea of coalitional game theory, is a method for assigning payouts to players depending on their contribution to the total payout (Merrick & Taly, 2020). Players cooperate in a coalition and receive a certain profit from this cooperation. The "game" is the prediction task for a single instance of the dataset. The "gain" is the actual prediction for this instance minus the average

prediction for all instances. The "players" are the feature values of the instance that collaborate to receive the gain.

The interpretation of the Shapley value for feature value j is: The value of the j-th feature contributed to the prediction of this particular instance compared to the average prediction for the dataset (Rodríguez-Pérez & Bajorath, 2020). The difference between the prediction and the average prediction is fairly distributed among the feature values of the instance, the Efficiency property of Shapley values. This property distinguishes the Shapley value from other methods. The Shapley value might be the only method to deliver a full explanation. In situations where the law requires explicability, like EU's "right to explanations", the Shapley value might be the only legally compliant method, because it is based on a solid theory and distributes the effects fairly (Rodríguez-Pérez & Bajorath, 2020).

Obviously there exist some disadvantages associated with shapley values. The first one is that it requires a lot of computing time. In most real-world problems, only the approximate solution is feasible. An exact computation of the Shapley value is computationally expensive because there are ***2^k*** possible coalitions of the feature values and the "absence" of a feature has to be simulated by drawing random instances, which increases the variance for the estimate of the Shapley values estimation.
Shapley values are also a drawback if we are seeking sparse explanations (explanations that contain few features). Explanations created with the Shapley value method always use all the features. A solution for this is SHAP, which is based on the Shapley value, but can also provide explanations with few features.

2.3.2.1 SHAP

The goal of SHAP is to explain the prediction of an instance ***x*** by computing the contribution of each feature to the prediction. The SHAP explanation method estimates Shapley values from coalitional game theory. The feature values of a data instance act as players in a coalition. One innovation that SHAP brings to the table is that the Shapley value explanation is represented as an additive feature attribution method, a linear model (Christoph Molnar, 2020). SHAP specifies the explanation as:

$$g(z') = \phi_0 + \sum_{j=1}^{M} \quad \phi_j z_j'$$

where ***g*** is the explanation model, ***z {0,1}*** is the coalition vector, ***M*** is the maximum coalition size and ***phi*** is the feature attribution for a feature ***j***, the Shapley values. In the coalition vector, an entry of 1 means that the corresponding feature value is "present" and 0 that it is "absent".

TreeSHAP uses the conditional expectation to estimate effects (Yang, 2021). If we conditioned on all features, if ***S*** was the set of all features, then the prediction from the node in which the instance ***x*** falls would be the expected prediction. If we did not condition on any feature, if ***S*** was empty, we would use the weighted average of predictions of all terminal nodes. If ***S*** contains some, but not all, features, we ignore predictions of unreachable nodes. Unreachable means that the decision path that leads to this node contradicts values in. From the remaining terminal nodes, we average the predictions weighted by node sizes. The mean of the remaining terminal nodes, weighted

by the number of instances per node, is the expected prediction for $\boldsymbol{x}$ given $\boldsymbol{S}$. You can visualize feature attributions such as Shapley values as "forces". Each feature value is a force that either increases or decreases the prediction. The prediction starts from the baseline. The baseline for Shapley values is the average of all predictions. In the plot, each Shapley value is an arrow that pushes to increase (positive value) or decrease (negative value) the prediction. These forces balance each other out at the actual prediction of the data instance. Shapley values can be combined into global explanations. If we run SHAP for every instance, we get a matrix of Shapley values. This matrix has one row per data instance and one column per feature. We can interpret the entire model by analysing the Shapley values in this matrix. This idea is implemented to capture SHAP feature importance.

The idea behind SHAP feature importance is simple: Features with large absolute Shapley values are important (Bowen & Ungar, 2020). Since we want the global importance, we sum the absolute Shapley values per feature across the data. Next, we sort the features by decreasing importance and plot them. SHAP feature importance is an alternative to permutation feature importance. There is a big difference between both importance measures: Permutation feature importance is based on the decrease in model performance. SHAP is based on magnitude of feature attributions.

The summary plot combines feature importance with feature effects. Each point on the summary plot is a Shapley value for a feature and an instance. The position on the y-axis is determined by the feature and on the x-axis by the Shapley value. The colour represents the value of the feature from low to high. Overlapping points are jittered in the y-axis direction, so we get a sense of the distribution of the Shapley values per feature. The features are ordered according to their importance. In the summary plot, we see first indications of the relationship between the value of a feature and the impact on the prediction.

2.3.1.2.1 Key Advantages

Since SHAP computes Shapley values, all the advantages of Shapley values apply: SHAP has a solid theoretical foundation in game theory. The prediction is fairly distributed among the feature values. We get contrastive explanations that compare the prediction with the average prediction. SHAP has a fast implementation for tree-based models. The fast computation makes it possible to compute the many Shapley values needed for the global model interpretations. The global interpretation methods include feature importance, feature dependence, interactions, clustering and summary plots. For all this, and because our best performance model was a gradient boosting machine based on tree models, SHAP was used to make the interpretation of model results.

3. Results

A total of 57,476 observations were used to train the model and 7,408 were used to test it. The total injuries and the time loss expressed in days associated with them is shown in Table 1. During all the years that were used to train the models, we can see that 300 non-contact injuries were witnessed, accounting for 0.52% of the observations of training. During the period used to validate the performance of the models, 39 non-contact injuries

were observed, accounting for 0.53% of the observations in validation. The descriptive data of the variables that were created to train the models can be observed in the supplementary material.

Table 2. Shows total of injured and mean time loss in train and test sets

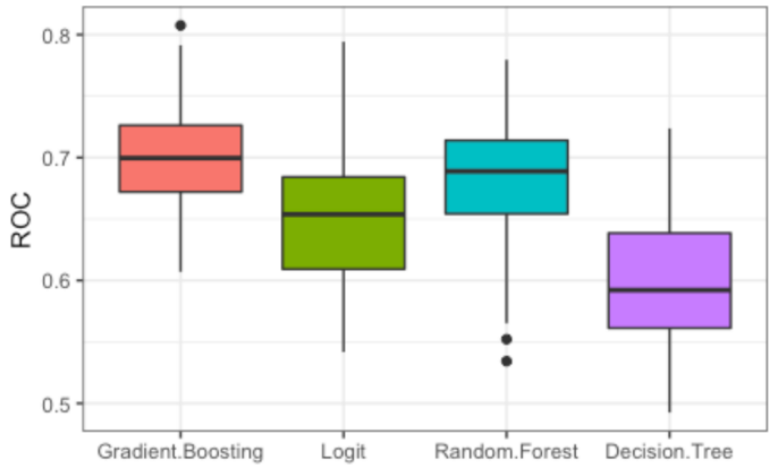
Variable	Test, N = 39[1]	Train, N = 300[1]
Time Loss	18 (5, 22)	14 (4, 16)
Inj.	39	300

[1] Mean (IQR); n

3.1 Model performance

As can be seen in the literature, the ability to predict an injury, using only variables associated with IL, was limited. In our case, the best validation results were found using the oversampling of the minority class balancing method. These results can be seen in Figure 3. The models that performed the best were the assembly models. In first place was GBM with a mean AUC of 0.7, followed by RF with a mean AUC of 0.69, in third place by the LR model with a mean AUC of 0.65 and in last position, showing to be slightly better than chance, the DT model with a mean AUC of 0.59.

Figure 18. AUC results in Test set for each model

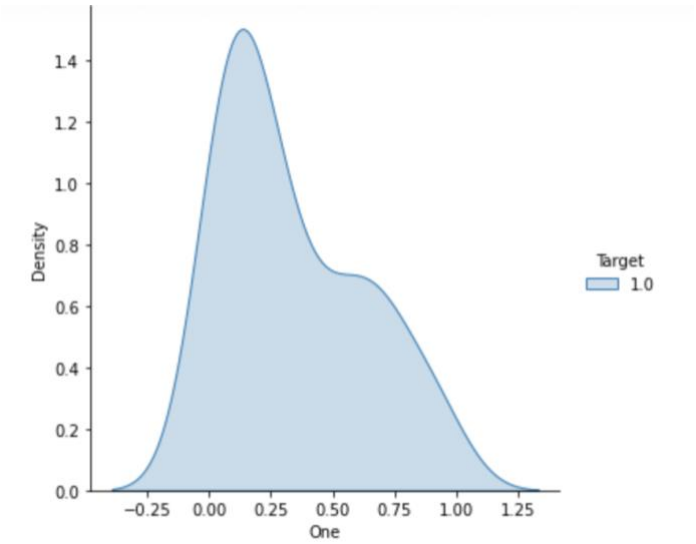


Although the AUC values in the best performing models are above what would be expected from a random classifier, the precision and recall values were particularly low. In the case of our GBM, using a decision threshold of 50%, a recall of 30% was obtained in validation and a precision of 3%. Showing a strong tendency to classify false positives.

By modifying the decision threshold, making it lower, it is possible to increase recall at the expense of precision. This trade-off will depend on the relative costs of obtaining false positives and false negatives.

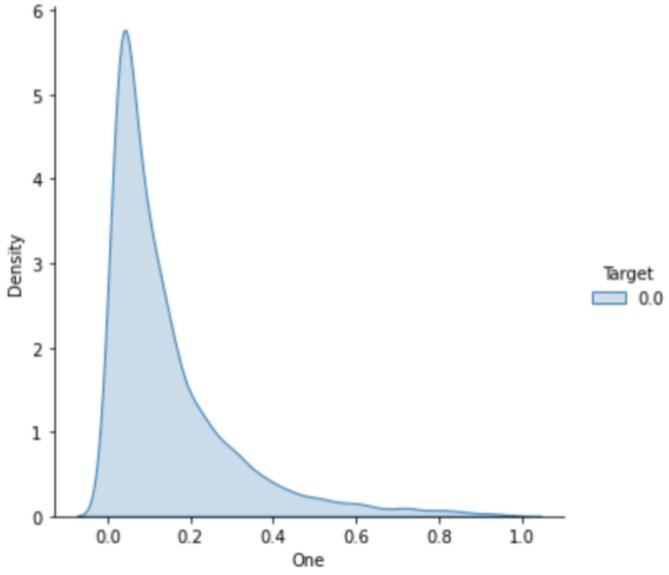
In order to graphically represent the predictions of the best model (GB), a kernel density graph was made with the probabilistic predictions for the test, of the observations whose ground truth was positive.

Figure 19. Distribution of predicted probabilities (ground truth equals 1 observations)



The results that are observed are logical for a model with the observed AUC values; the confidence when predicting that an observation belongs to the positive group given that it is positive is low. But this distribution is different with respect to the predictions made for the group of observations whose ground truth is negative. This shows that the model, although poorly, is capturing information from the data.

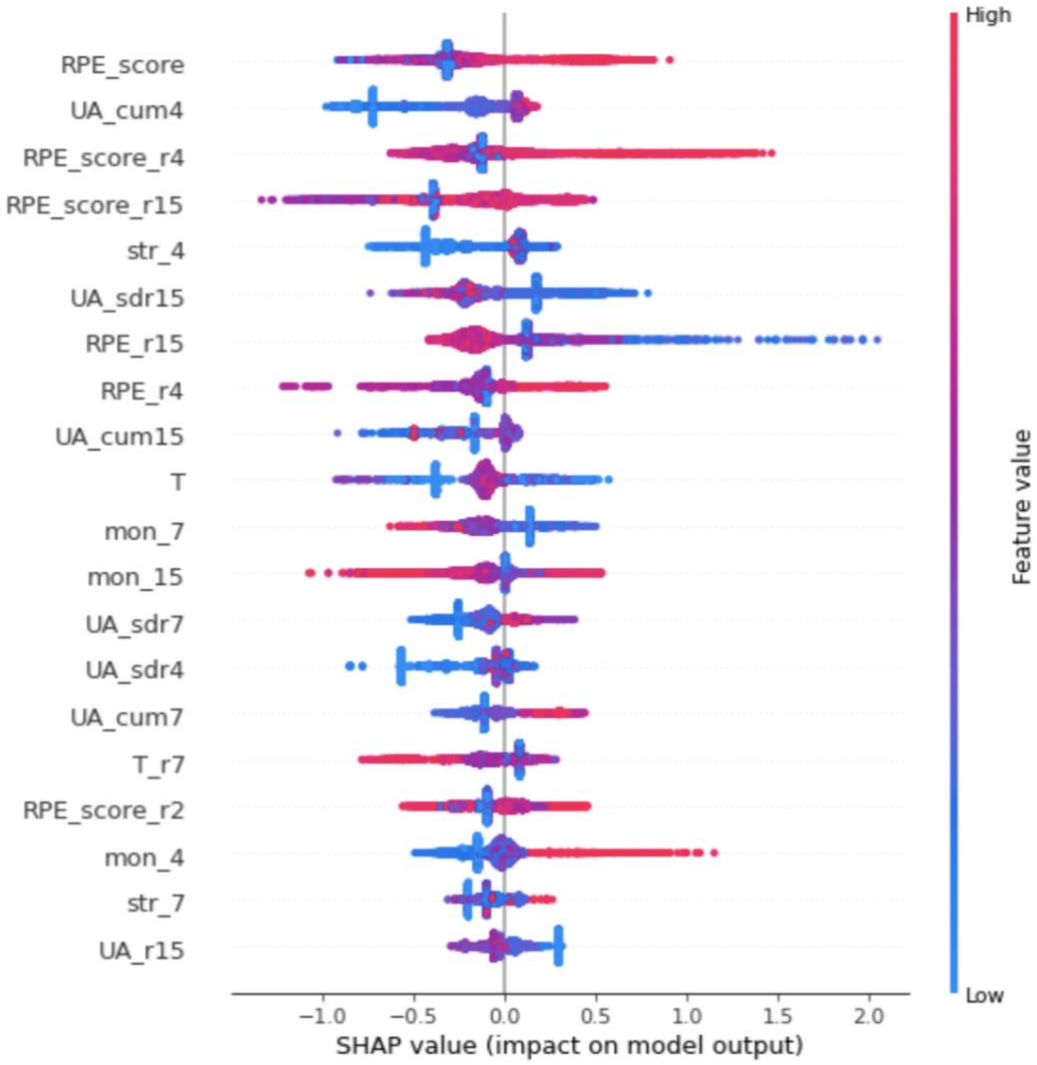
Figure 20. Distribution of predicted probabilities (ground truth equals 0 observations)



3.2 Interpretability exercises

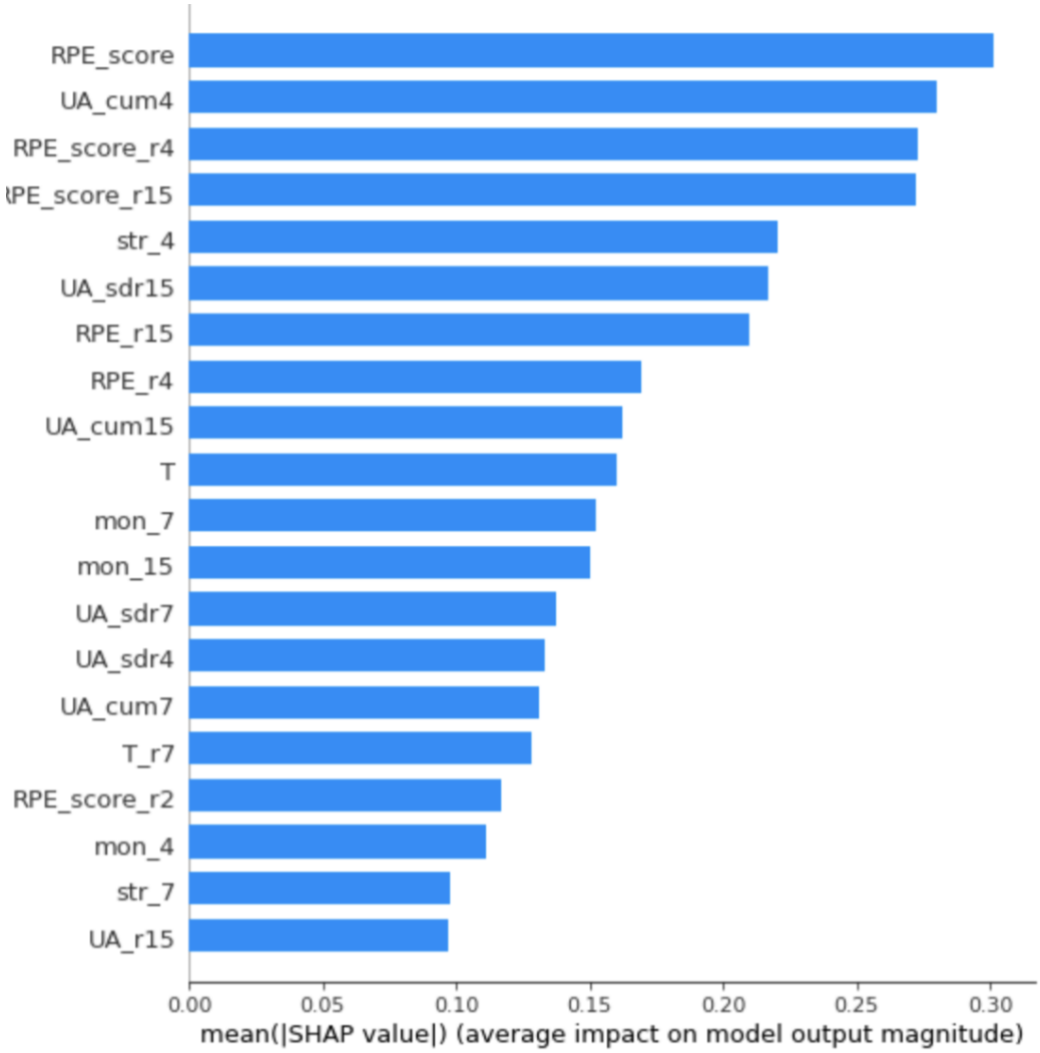
After having evaluated the performance of our models with the validation set (year 2019), we choose the best of the 4 models to try to understand which are the variables that most influence the output of the model and how they impact the probability estimation of suffering an injury. For this as mentioned above, SHAP-values were used. The SHAP-values represent the impact of a variable in the decision process of the model. Positive SHAP-values represent a higher probability of a positive prediction (i.e get injured). In order to visualize these results a SHAP summary plot was created. Dots representing the SHAP-vlaues for each feature value of a player in the dataset are plotted horizontally next to the feature (Figure 4).

Figure 21. Shap-values Summary Plot.



Using SHAP-values it is also possible to estimate the importance of the variables in the results of the model. These are represented in Figure 22.

Figure 22. Feature Importance.



In this way we can observe that among the variables that best explain the probability of suffering an injury are those created from the relative position that a player occupies with respect to the rest of the team at a given moment of time, as well as the historical data for this variable. Positioning yourself negatively with respect to the team, that is, obtaining high sRPE Scores, increases the probability estimated towards the positive class by the model. Low accumulated UA values of the last 4 days decrease the probability of being assigned to the positive class. High 15-day historical sRPE values decrease the probability of being assigned to the positive class. Similarly high historical variability, represented as the average standard deviation of the last 15 days (UA_sdr15) tends to decrease the probability of being assigned to the positive class by the model. Low recent strain index values (str_4) show to have a protective effect in some players.

3.3 Impact of reducing the number of explanatory variables

Because the explanatory variables that were incorporated into the model come from only two variables, which are exposure time and sRPE, there is a strong correlation between them. This could indicate that there is a certain degree of redundancy in the

information they provide to the model about the target variable that is trying to be predicted.

To evaluate this redundancy of information, we perform an experiment that is based on reducing the number of variables used to train the model and evaluating its performance, while comparing it with the performance of the model that was trained with all the variables.

In the present experiment, the performance of the model was evaluated being trained with 2, 4, 6, 8, 10, 12, 14 and 16 variables. The model used was GBM and the class balancing method used was over-sampling.

The steps of the experiment were as follows:

1. We choose N variables randomly
2. we optimize the model parameters by random search CV using N variables.
3. We evaluate the model with the test data and calculate AUC ROC.
4. we repeat steps 1, 2, 3, 30 times with each subset of N variables.

Results are shown below.

Figure 23. Results are shown for experiments with 2 and 4 variables (left), and experiments with 6 and 8 variables (right).

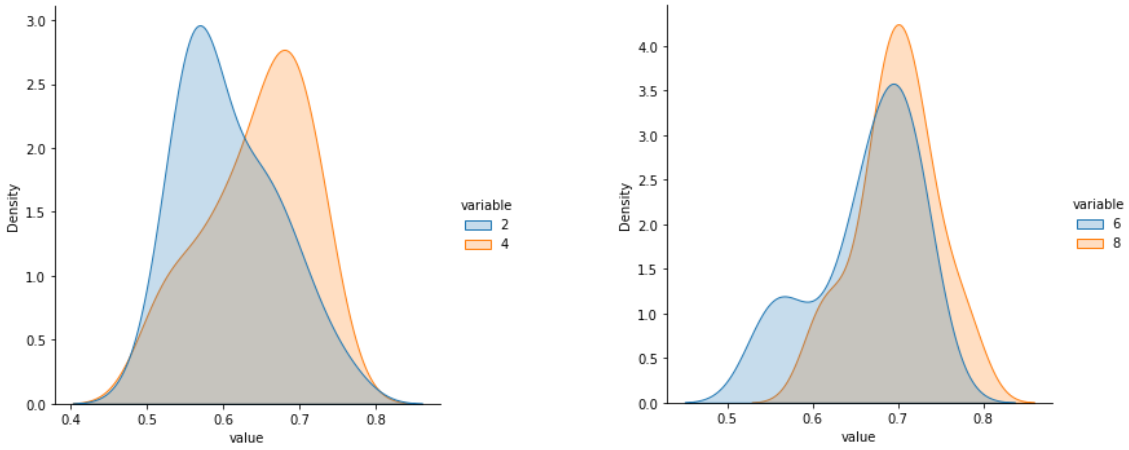
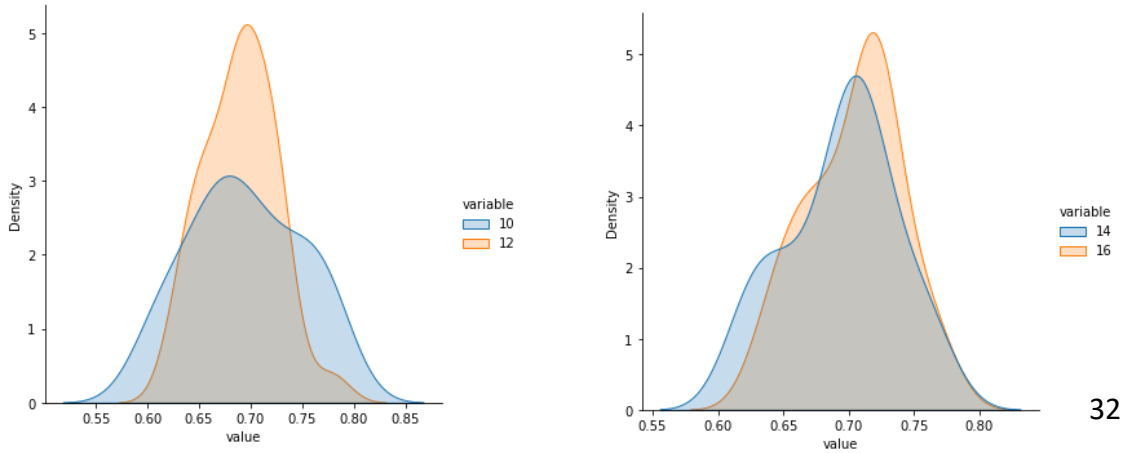


Figure 24. Results are shown for experiments with 10 and 12 variables (left), and experiments with 14 and 16 variables (right).



As would be expected, the models that were trained with fewer variables performed worse than those that were trained with more variables. As we add explanatory variables, we can see that the AUC ROC distribution begins to converge to the results found when training the model with all the variables.

Some interesting results emerged during the experiment.
When analysing the results of the models that were trained with two variables, we can see cases that stand out from the mean, reaching values higher than AUC ROC >= 0.65. By only using two variables, it is easy to inspect which variables are the ones that were used to train the model. In all cases, variables associated with the RPE score were included, a variable that stood out both in the exploratory analysis and during the analysis of the interpretability of the models.

Another interesting feature that emerges when analysing the experiment in dynamics is that not only does the mean AUC ROC increase as the number of predictors increases, but the variance of these results decreases.

The results of these experiments may be of particular importance for the implementation of this model, since they indicate that we could work on the explanatory variables to reduce their redundancy, and propose new variables that better capture the information contained in the data.

4.Discussion

The models used that exploit the relationship between IL and the risk of suffering an injury were shown to have limited predictive capacity. The more complex models (GBM and RF) managed to outperform the simpler models (LR and DT), possibly thanks to their ability to detect complex interactions between explanatory variables and the target variable. The general low precision of the models to distinguish between classes could be attributed to the fact that by themselves, the IL measures are unable to perfectly explain the risk of injury. This could become apparent when analysing the results of the SHAP-values. Of the 5 most important variables, 3 were derivations of the sRPE score feature, which compares a player with the rest of the team at a given time. That a certain player perceives a training as more strenuous than the rest of the team, provides information on internal variables of said player, partially explaining his physical capacity to resist and adapt to training. This shows that the interpretation of the models used could provide clinically relevant information on the interactions that exist between a player's physical state and workloads. This could suggest that incorporating variables that provide information on the physical capacity of the players could improve the performance of the models.

Other authors have suggested that models based on machine learning have different sensitivity for the prediction of certain injuries. In our case, not having differentiated between different types of non-contact injuries could have been a limitation. This is because the pathophysiological underpinnings of different types of lesions differ significantly.

As previously mentioned, other authors have suggested that there is no predictive power in IL-derived metrics and the risk of injury. These conclusions were the result of analysing previous works which used descriptive statistical methods to study the

relationship between IL and the incidence of injuries. Our results suggest that IL metrics provide partial information on a player's injury risk.

4.1 Relative costs analysis

Although our GBM model was shown to have low precision and recall values, its usefulness should be considered taking into account the relative cost of obtaining false positives and true positives (Bahnsen et al., 2015). The translation of obtaining a false positive is not necessarily synonymous with generating a time loss (in training sessions and games) in the player; multiple strategies could be chosen to keep the player active and at the same time prevent him from incurring an injury. In turn, a coach's appetite for risk is not constant and depends on the time of year in which it is considered. For example, during the off season the risk tolerance of a coach could be considered high, since an injury to a certain player only implies that he suffers a time loss in training sessions, generating economic losses only associated with the absence of said player in the training sessions. On the other hand, during the on season, the risk tolerance of a coach decreases significantly, since an injury of a key player could indicate a significant loss of competitiveness of the team, generating significant economic losses that simply exceed the player's salary time loss. in question.

Taking all this into account, one option that could be presented to us is to modify the threshold of our model according to the preferences of the coach at a certain moment. In this way we would be working on the balance between precision and recall.

To understand how this equilibrium behaves, the performance of the model was evaluated using different thresholds, from 0.05 to 0.95.

Table 3. Recall and Precision trade off

Precision	Recall	Tresh
0.684530	89.743590	0.05
0.799087	71.794872	0.10
1.032631	64.102564	0.15
1.236650	56.410256	0.20
1.444867	48.717949	0.25
1.736466	43.589744	0.30
2.060440	38.461538	0.35
2.545455	35.897436	0.40
3.110048	33.333333	0.45
3.529412	30.769231	0.50
4.230769	28.205128	0.55
4.455446	23.076923	0.60
4.666667	17.948718	0.65

5.737705	17.948718	0.70
4.878049	10.256410	0.75
5.454545	7.692308	0.80
9.677419	7.692308	0.85
17.647059	7.692308	0.90

A natural way to approach this problem would be defining a cost matrix, assigning a cost to each type of error (false positives and false negatives) and modifying the decision threshold of the model in such a way as to minimize the sum of the costs for each error. In order to achieve this, it is necessary to re-train the model and evaluate it with the new threshold (Z.-H. Zhou, 2011)(Fernández et al., 2018). Determining the expected losses from a player injury is relatively straightforward. A valid way could be to calculate the average daily salary of the players and multiply it by the average lost time (expressed in days) per injury, thus obtaining the average loss per injury. This could be defined as the cost of a false negative. Our problem arises when trying to quantify the costs of false positives since they depend on the type of intervention that the club's medical staff chooses as appropriate for a player whose estimated probability of suffering an injury for the next 72 hours is high. After consulting with multiple experts in the area, all concluded that the intervention would not necessarily generate time loss in the player. For example, for a certain player whose probability of injury has been estimated high, the intensity and load of training could be modified, extra sessions of physical therapy could be indicated, a series of physical tests could be indicated to evaluate his condition generally and thus avoid injury. All this card of possibilities does not represent a cost for the club since it is prepared to offer such services to the players. In this way our cost for a false positive could be zero. Of course, it should also be taken into account how effective these measures are, but in our case, we do not have data in this regard, so it is not possible to estimate such effectiveness.

Assuming that the effectiveness of the measurements is 100% and that the cost of performing them is zero (since in our particular case all preventive work is carried out in house by the club's medical staff), we determine that our cost for a false positive is zero. This is a problem, since it is easy to assess that the threshold that minimizes our costs in this case is zero, that is, we classify all players as high risk and indicate the treatment, avoiding incurring costs for false negatives. In appendix, 6.5 a mathematical formalization of this approach is introduced.

4.2 Contractual negotiations

In the world of professional sports, drafting contracts can be a challenging task. This is because many characteristics of the players are often overlooked, and estimating the returns generated by incorporating or keeping an individual within a team is often a difficult task. In many cases, there have been cases where closures have been incorporated that penalize the player's salary if he does not show an expected physical performance (Michael A Leeds, Peter von Allmen, & Victor A. Matheson, 2018). In literature, the physical performance of an athlete is usually encompassed in what is called general skills of the same, and the better these general skills, the greater the value of the athlete. By this we mean that if we maintain constant the talent that an athlete may have, that he can, for

example, run faster and for longer without fatigue, they make the value that he can bring to his team / employer greater. At the same time, it has been shown that the better the physical performance of an athlete, the lower the risk of injury, reducing the time loss due to injuries (Charest & Grandner, 2020).

The relevant unit for labour supply is usually a season as players are usually hired for the length. In the case of salary renegotiations, one could take the historical data of the last year of a given player, generate results for each interval of 72 hours, and with these results accumulated values could be calculated which could be used to evaluate and compare this player's risk of injury to other players. For example, when computing the probability of injury in each window of 72 hours and multiplying it by the average time loss of a player, we would have an expected value of days which the player would lose due to being injured, this translates directly as a loss for the team, since they will be days where the player enjoys his salary but does not contribute his services to the team

To generate a probabilistic output, the model uses as input historical data from the training records of each player, as well as information on their relative position with the team. In this way, for players who stay for more than one season in a certain team, the output of the model could work to adjust salaries in the contracts or even by incorporating clauses in which an expected performance is stipulated, functioning as an incentive for the participation in training and improvement of general skills of the players, thus increasing the intrinsic value of the player, subsequently reflected in the profits of the employer.

5. Conclusions and recommendations for future research

5.1 The nature of injuries

Previous research has demonstrated that training and competition stress result in temporary decrements in physical performance and significant levels of fatigue post-competition (Cunniffe et al., 2010; McLellan, Lovell, & Gass, 2011). These decrements are typically derived from increased muscle damage, impairment of the immune system, imbalances in anabolic – catabolic homeostasis, alteration in mood (Cunniffe, Proctor, Baker, & Davies, 2009) and reduction in neuromuscular function (NMF) (McLellan, Lovell, & Gass, 2010). The resultant fatigue from these variables can take up to 5 days to return to baseline values post-competition, with sports that have frequent competition (i.e. often weekly in team sports) also inducing accumulative fatigue over time. In addition to the significant amounts of fatigue induced by competition, many athletes experience fatigue as a result of the work required to develop the wide variety of physical qualities that contribute significantly to performance. For example, in both team and individual sports, speed, strength, power and endurance are required in addition to technical and tactical skills.

An accumulation of fatigue can result in overtraining, which has a significant negative impact on performance (Meeusen et al., 2013). For example, the investigation by (Johnston, Gabbett, & Jenkins, 2013) regarding the physiological responses to an intensified period of rugby league competition over a 5-day period found that cumulative fatigue appeared to compromise high-intensity running, maximal accelerations and defensive performance in the final game. This means that when athletes do not receive adequate time to recover between training and competition, fatigue will accumulate,

compromise key aspects of performance and result in an increased risk of injury and illness to the athlete.

The majority of training load / fatigue monitoring research has focused on acute responses to measure recovery of performance variables and the acceleration of this process through the implementation of recovery modalities (Taylor et al., 2015). In contrast, fewer attempts have been made to monitor acute and / or cumulative load and fatigue variables longitudinally to determine the association with injury / illness. Longitudinal monitoring refers to the investigation of how change or accumulation in training load / fatigue is associated with injury / illness over time. The use of long-term monitoring allows for the measurement of training load and fatigue variables to identify any injury / illness trends in order to provide practitioners with objective data for planning training over multiple blocks, rather than relying solely on anecdotal evidence, with the aim of reducing overtraining and injury / illness (Gabbett, 2010). Any subsequent reduction in injury and illness is likely to have a significant impact on team performance due to the large percentages of athletes from training squads in team sports injured at any one time (Brooks, Fuller, Kemp, & Reddin, 2005), and the association between the number of injuries and matches won

In this sense, the one-dimensional probabilistic output of our model could be used as a historical monitoring metric since it incorporates not only information about individual IL but also captures information that positions each individual with respect to the rest of the team. The interactions captured by our model may be useful for evaluating the cumulative impact of training and competition on athletes. This could be validated using the model trained with the present data, and the validations carried out with future data that are incorporated into the database.

5.2 Causal interpretation

In the present work we have made considerable contributions on the path of elucidating the relationship between IL and harmful risk. However, as previously mentioned, more work must be done to be able to quantify the causal relationship between IL and injurious risk. Although there is a theoretical framework that supports this relationship, there is still no work that has tried to quantify (if such a relationship existed). In our case, we set ourselves the objective of predicting the response of Y (harmful risk) when certain $\boldsymbol{X}$ characteristics (IL metrics) were present (Zhao & Hastie, 2019). In a causal paradigm, we would like to evaluate how much it affects $\boldsymbol{Y}$ that generates interventions in $\boldsymbol{X}$. In this case, it would be of great importance to determine said effect, mainly because of the possibilities it would open for making decisions based on preventing injuries.

5.3 Survival Analysis

Survival analysis is the phrase used to describe data analysis in the form of times from a well-defined origin to the occurrence of some particular event or end point. In medical research, the origin of time will often correspond to the recruitment of an individual into an experimental study, such as a clinical trial to compare two or more treatments. This, in turn, may coincide with the diagnosis of a particular condition, the initiation of a

treatment regimen, or the appearance of an adverse event. If the end point is the death of a patient, the resulting data is literally survival times. However, data can be obtained in a similar way when the end point is not fatal, such as pain relief or symptom recurrence. In this case, the observations are often referred to as time-to-event data. Another characteristic of these types of data is censorship. The survival time of an individual is said to be censored when the end point of interest for that individual has not been observed. This may be because data from a study needs to be analyzed at a time when some people are still alive (or the event of interest in question has not happened). Alternatively, an individual's survival status may not be known at the time of analysis because that individual was lost to follow-up.

A review of this topic was carried out in detail by (Nielsen et al., 2018). In this article, the methodological challenges that must be taken into account when using time-to-event models in the context of injury risk analysis in sports sciences are raised. Which include the difficulty of dealing with injuries that go through multiple states (tendinopathy is usually classified in different ways from diagnosis to full recovery), recurrent events (in most survival analyses, after the onset of the event individuals withdraw from the study, in our case these are sustained over time and many times suffer more than one injury in a certain period), the difficulty of defining follow-up and cut-off periods for observational studies and the amount of observations necessary to achieve robust analysis.

In our case, in turn, due to the temporal dependence of the explanatory variables, precautions had to be taken to adapt the model used. The indications given in chapter 8 of (David Collett, 2015) were taken as a reference to solve this.

The creation of the follow up and censorship data was not trivial. In this case, we consider that the censorship was reached at the end of the season of the corresponding year, since between training seasons the tracking of the players was lost due to the ~ 2 months of vacation.

Taking all this into consideration, we propose to use a Cox proportional hazard model to evaluate how the risk profile of the players is modified as a function of the internal load variables created. The data used for the present analysis corresponds to the train data. We use the model interpretation section as a reference to choose the explanatory variables in this section. In this way, we chose the RPE score, the 4-week accumulated UA, and the 4-week moving average of the strain index as descriptive variables.

Table 4. Results of Proportional Cox Model.

Variable[1]	Coef[1]	Exp.coef[1]	Error.standar[1]	p.val[1]
RPE Score	0.31	1.36	0.13	<0.01
UA Cum 4	0.23	1.25	0.12	<0.01
STRr 4	0.18	1.19	0.14	0.18
[1] R2 = 0.52				

It can be observed in the results of the model that only the variable RPE score and UA cum 4 were significant. In the case of RPE Score, each time a unit increases, the instantaneous risk of injury increases 1.36 times; For UA Cum 4, each time you increase a unit, your instant risk of injury increases 1.25 times. These results were consistent with the findings made in previous studies.

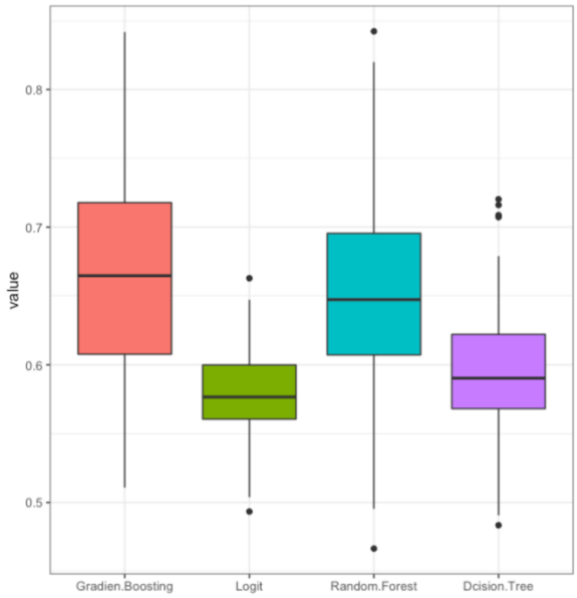
In this regard, future work could be directed to implement models that predict the time until a future injury and that fixed time windows are not restricted as was the case in our case.

6. Appendix

6.1 Second Experiment Results

In this case random oversampling was used in order to overcome the imbalance problem and in order to reduce high correlation between predictors, PCA was applied.

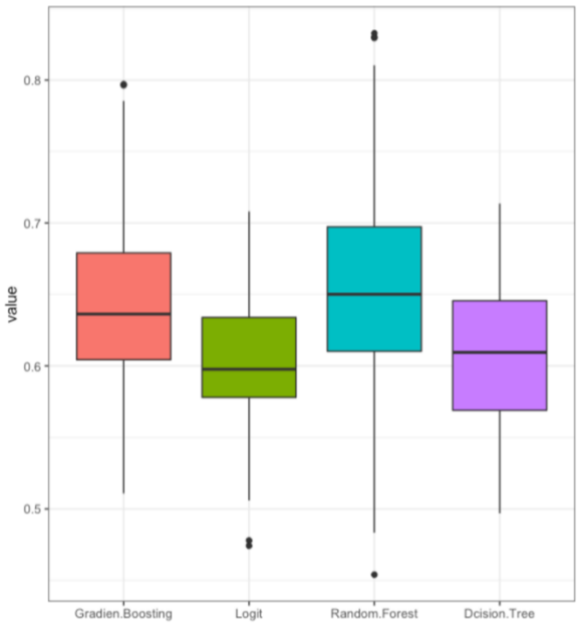
Figure 25. AUC results over cross validation folds for the second experiment.



6.2 Third Experiment Results

In this case SMOTE was used to overcome imbalance problems. No preprocessing was applied in this case.
Results are shown in figure 26.

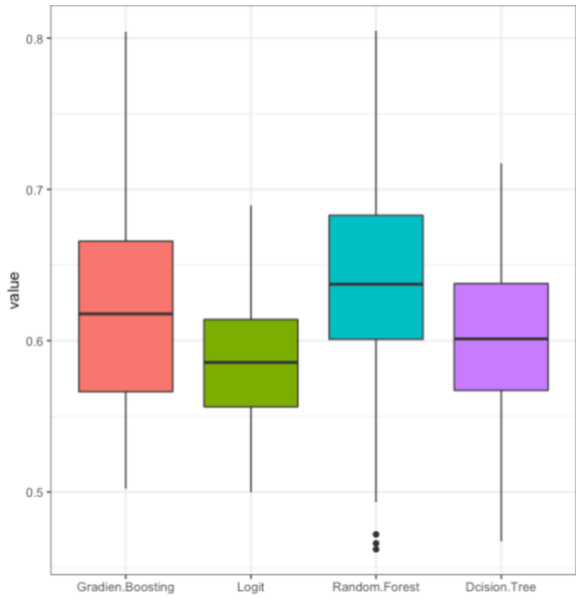
Figure 26. AUC results over cross validation folds for the third experiment.



6.3 Fourth Experiment Results

In this case SMOTE and PCA were applied.

Figure 27. AUC results over cross validation folds for the fourth experiment.



6.4 Fifth Experiment

As mentioned in previous chapters, random undersampling involves randomly selecting examples from the majority class to delete from the training dataset.

When using random undersampling, different sampling strategies can be proposed, for example, the majority class can be reduced until the number of classes is equal to the minority class, or value can be chosen which will be a percentage relative to the minority class, specifically the number of examples in the minority class divided by the number of examples in the majority class. For example, if we set sampling strategy to 0.5 in an imbalanced data dataset with 1,000 examples in the majority class and 100 examples in the minority class, then there would be 200 examples for the majority class in the transformed dataset (or 100/200 = 0.5).

In our case, we use the second strategy and try multiple values for it. The best results were found for intermediate values. This may be due to extreme class imbalance, that is, the final balanced dataset with equality of classes has too few total observations, which generates a loss of model performance.

AUC ROC values are reported in the following table:

Table 5. Mean AUC results for the fifth experiment.

Sampling Strategy	GBM	RF	Logit
0.01	0.61	0.59	0.51
0.13375	0.60	0.58	0.56
0.2575	0.62	0.61	0.59
0.38125	0.64	0.61	0.61
0.505	0.64	0.63	0.61
0.62875	0.65	0.65	0.62
0.7525	0.68	0.64	0.63
0.87625	0.65	0.62	0.61
1.0	0.63	0.62	0.60

6.5 Descriptive Statistics of Features

Variable	**Test**, N = 7,408[1]	**Train**, N = 57,476[1]
RPE	4.0 (1.0, 5.0)	4.0 (3.0, 6.0)
T	60 (40, 80)	70 (60, 90)
UA	270 (45, 400)	300 (180, 450)
RPE_r2	4.00 (2.00, 5.00)	4.50 (3.50, 6.00)
RPE_r4	4.00 (2.75, 4.75)	4.25 (3.50, 5.50)
RPE_r7	3.86 (2.86, 4.71)	4.29 (3.71, 5.29)
RPE_r15	3.93 (3.13, 4.60)	4.20 (3.60, 4.87)
T_r2	65 (35, 75)	70 (60, 82)
T_r4	62 (45, 75)	71 (62, 82)
T_r7	62 (48, 74)	73 (61, 84)
T_r15	63 (51, 72)	71 (63, 80)
UA_r2	270 (130, 390)	320 (225, 460)

UA_r4	278 (172, 368)	342 (250, 480)
UA_r7	280 (196, 363)	347 (271, 521)
UA_r15	289 (215, 368)	351 (273, 463)
UA_sdr2	99 (21, 223)	127 (57, 283)
UA_sdr4	162 (90, 260)	199 (132, 337)
UA_sdr7	189 (130, 259)	234 (165, 494)
UA_sdr15	204 (159, 325)	298 (183, 495)
UA_cum2	540 (260, 780)	640 (450, 920)
UA_cum4	1,100 (680, 1,460)	1,370 (1,000, 1,920)
UA_cum7	1,950 (1,340, 2,520)	2,410 (1,890, 3,640)
UA_cum15	4,280 (3,110, 5,450)	5,160 (4,075, 6,890)
mon_2	1.4 (0.7, 4.1)	2.1 (1.0, 4.9)
mon_4	1.29 (0.86, 2.06)	1.38 (1.06, 2.09)
mon_7	1.27 (0.88, 1.70)	1.29 (0.96, 1.70)
mon_15	1.20 (0.89, 1.52)	1.13 (0.92, 1.51)
str_2	885 (225, 3,190)	1,704 (650, 3,930)
str_4	783 (259, 1,603)	1,080 (604, 1,998)
str_7	725 (272, 1,301)	925 (516, 1,491)
str_15	682 (273, 1,134)	821 (472, 1,249)
PI		
0	4,291 (58%)	36,552 (64%)
1	1,714 (23%)	14,778 (26%)
2	854 (12%)	4,538 (7.9%)
3	419 (5.7%)	1,024 (1.8%)
4	10 (0.1%)	494 (0.9%)
5	120 (1.6%)	90 (0.2%)
RPE_score	0.67 (0.22, 0.83)	0.75 (0.50, 0.92)
RPE_score_r2	0.62 (0.38, 0.77)	0.65 (0.50, 0.80)
RPE_score_r4	0.61 (0.43, 0.74)	0.65 (0.53, 0.77)
RPE_score_r7	0.61 (0.46, 0.71)	0.63 (0.54, 0.73)
RPE_score_r15	0.61 (0.49, 0.69)	0.62 (0.55, 0.67)

[1] Median (IQR); n (%)

6.6 Cost Sensitive classification

Given a cost matrix ***c = (c(i,j)(x))*** where ***c(i,j)(x)*** represents the cost (perhaps negative or zero) of classifying ***x*** (which is really a member of class ***j***) as being in class ***i***. The first reduction one might make is in assuming that these functions ***c(i,j)*** are, in fact, constant.

The two-class case of class dependent cost sensitive learning has been adequately solved (at least in some sense) by reducing to the two-class cost-insensitive learning case and then choosing an optimal decision threshold determined by the cost matrix (assuming some very mild reasonableness conditions on the cost matrix). The classical solution, which does not reduce to the two-class case involves weighting class I by:

$$w(i) = \frac{nc(i)}{\sum_{k=1}^{k} n(k)c(k)}$$

where ***c(i)*** is the sum of the ***ith*** column of the cost matrix:

$$c(i) = \sum_{j=1}^{k} c(j,i)$$

and ***n(k)*** is the number of examples in class ***k,*** ***κ*** is the number of classes, and ***n*** is the total number of examples in the training set.

Suppose you have a learning task with training data ***X = {x(1), x(2), . . . , x(n)} → {0,1}*** that you would like to model. Let's say that there are ***n(0)*** negative cases and ***n(1)*** positive cases. Suppose we choose to rebalance that data (say up or down sampling the negative class) so that there are ***n(1)*** negative cases and ***n(1)*** positive cases. That is, we are multiplying the number of negative cases by ***n(1)/n(0).***

Consider now the following theorem of Elkan, to make a target probability threshold ***p**** correspond to a given probability threshold ***p(0),*** the number of negative examples in the training set should be multiplied by:

$$\frac{p^*}{1-p*}\frac{1-p(0)}{p(0)}$$

Suppose further that we train a model ***f : X → [0, 1]*** which models the probability **P(y = 1|x),** and predict using the threshold *0.5*. That is:

$$Pred(x) = \begin{cases} 0 & f(x) \leq 0.5 \\ 1 & f(x) > 0.5 \end{cases}$$

Using the Elkan theorem, we can calculate the decision threshold of the original data (before balancing) to which this corresponds. This calculation is below:

$$\frac{n(1)}{n(0)} = \left(\frac{\frac{n(1)}{n}}{\frac{1-n(1)}{n}}\right)\left(\frac{1-0.5}{0.5}\right)$$

So we see that modelling on the balanced data with a threshold of 0.5 corresponds to modelling on the raw (or original) data with a threshold of ***n(1)/n***. What is left to observe now is that the optimal decision threshold is determined precisely by the existence of a cost matrix.

Let ***c*** be a real valued matrix:

$$c = [c(0,0)\ c(0,1)\ c(1,0)\ c(1,1)\ ]$$

where ***c(i,j)*** is the cost of predicting class ***i*** when the truth is class ***j*** (e.g. ***c(0,1)*** is the cost associated to a false negative).
We will further require that the following two conditions be satisfied (known as the "reasonableness" conditions):

$$(1)c(0,1) > c(0,0)$$

$$(2)\ c(1,0) > c(1,1)$$

This means that correctly labelling a sample is preferable than incorrectly labelling a sample. There is always a cost (perhaps negative or zero) associated to classifying an example as class ***i*** that is in actuality a member of class j.
A natural question is then, given a cost matrix *c,* and a function $f : X \rightarrow [0, 1]$, which models the probability $P(y = 1|x)$, what criteria should be used to determine if we should classify ***x*** as 0 or 1? The natural criteria is to classify *x* as a 1 exactly when the expected cost of classifying as a 1 is less than classifying it as a 0. (If the two expected values are equal, it doesn't matter what we choose).

More precisely, we should label ***x*** as a 1 if and only if:

$$\big(1 - f(x)\big)c(1,0) + f(x)c(1,1) < \big(1 - f(x)\big)c(0,0) + f(x)c(0,1)$$

The optimal threshold is then the value ***p**** satisfying:

$$(1 - p^*)c(1,0) + p * c(1,1) < (1 - p^*)c(0,0) + p^*c(0,1)$$

Then, solving for ***p****:

$$p^* = \frac{c(1,0) - c(0,0)}{c(1,0) - c(0,0) + c(0,1) - c(1,1)}$$

To determine the value of the false positives, meetings were held with the club's medical staff. During these meetings, the trade-off that exists between false positives and false negatives was explained to the members of the staff, in order to have an estimate of the expected cost of generating an intervention. Because all the preventive work is carried out by the club's staff, for example, performing kinesiology evaluations, does not present an extra cost, since the club has physical therapists working on a permanent basis. This means that a moderate increase in individuals receiving preventive treatment would not generate higher costs for the club. This assumption could be violated if the number of individuals receiving preventive treatment exceeds the work capacity of the medical staff. Although the latter could be the case, the club does not have specific data to determine said maximum capacity and thus we cannot, with current data, determine the cost of false positives exceeding said capacity threshold.

taking into account the previous discussion, we are violating the assumption of reasonableness, since the cost of a false positive (***c (1,0)***) is not greater than the cost of a correct assignment. By replacing our false positive cost in the previous formula, we can see that the optimal threshold value that we obtain is zero, that is, our optimum would be to classify all observations as positive cases. In this way, we ruled out the possibility of using cost-sensitive classification techniques in our study.

6.7 Hyperparameter sensibility analysis

In order to study the robustness with which the hyperparameters were optimized in the model with the best performance, an experiment was designed to evaluate how the performance of the model is modified by disturbing the optimal values of the parameters.

The data used for this purpose was balanced using oversampling technique.
The parameters that were evaluated in this experiment were:

- Max depth: The maximum depth of a tree.
- Learning rate: This determines the impact of each tree on the final outcome. GBM works by starting with an initial estimate which is updated using the output of each tree. The learning parameter controls the magnitude of this change in the estimates.
- Min sample Split: Defines the minimum number of samples (or observations) which are required in a node to be considered for splitting.
- Min sample leaf: Defines the minimum samples (or observations) required in a terminal node or leaf.
- Max features: The number of features to consider while searching for a best split. These will be randomly selected.
- N estimators: The number of sequential trees to be modelled

This experiment consisted of the following steps:

1. For each model parameter, a set of values was defined around the optimal value found by random search CV.
2. All the hyperparameters were positioned at the optimal value, except one.
3. The model is trained with each of the values chosen around the optimal parameter.
4. Performance was tested on the test data set.
5. Steps 2, 3, 4 were repeated until all the hyperparameters of the model had been iterated.

The results for each parameter can be seen in the following figures.

Figure 28. Results for perturbations in learning rate (left) and N estimators (right)

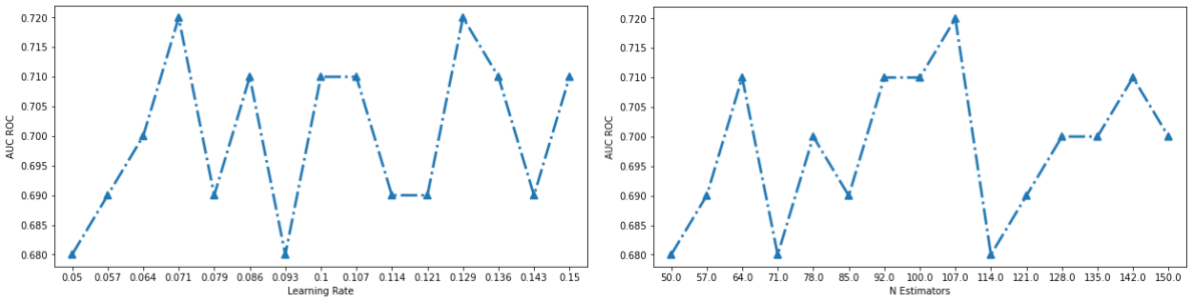


Figure 29. Results for perturbations in max depth (left) and min sample Split (right)

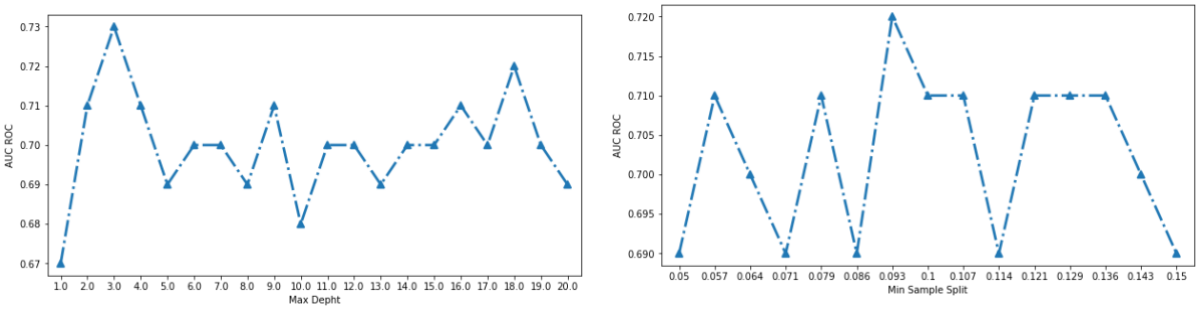
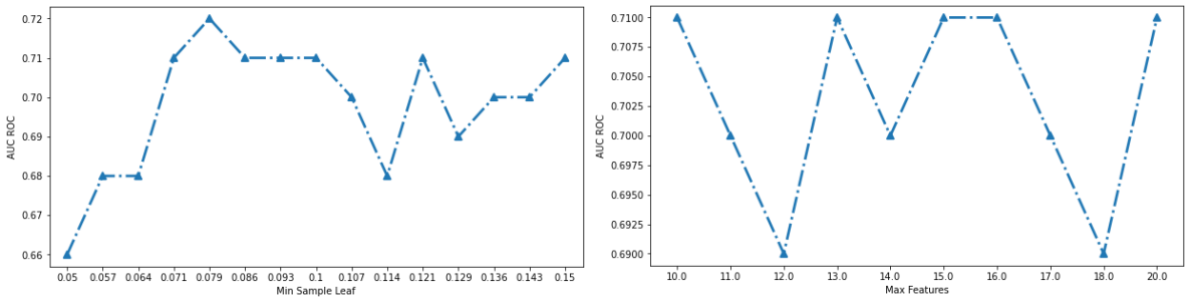


Figure 30. Results for perturbations in min sample leaf (left) and max features (right)



Although the variation rate of the AUC ROC does not exceed ≈ 6% within the space of parameters tested for all hyperparameters evaluated. Some volatility of the performance after changing hyper parameter values can be noticed. In many cases we can see that the performance of the model changes abruptly with small changes. This is the case, for example, of the max depth hyperparameter, where we can see that the optimal performance is reached at the value of 3 and quickly drops when reaching the value of 5.

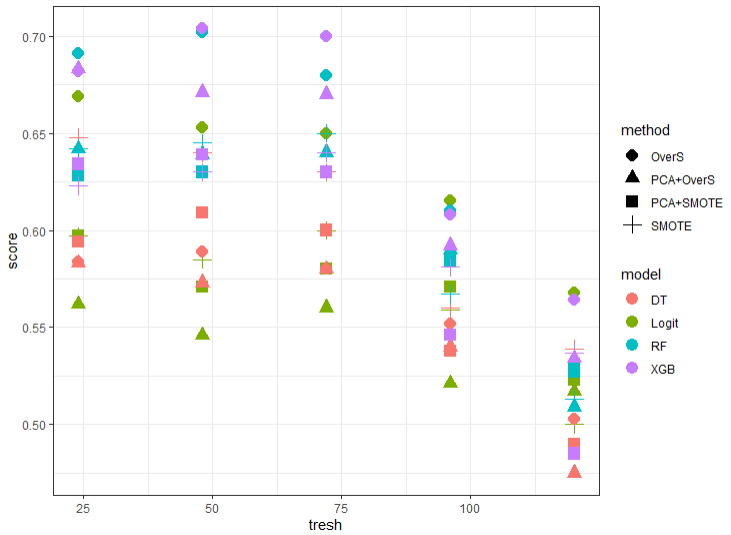
To deal with this problem, other hyperparameter optimization methods could be evaluated or a larger space of hyperparameters could be evaluated and the process iterated until we find a combination which guarantees greater stability.

6.8 Choice of predictability threshold

One of the objectives of this study was to determine how many days in advance it would be possible to predict an injury of a player. The choice of this time threshold is influenced by two factors, the first is that the further the studied event is from the moment of prediction, the more difficult it becomes to predict it with certainty, and the second, the time threshold must be sufficiently long enough to allow the medical staff to make decisions regarding the player at risk.
Taking all this into account, we decided to test the performance of the models at different thresholds up to the time of injury. The predictive performance was evaluated at 24, 48, 72, 96 and 120-hours intervals. The class balancing methods and pre-processing methods were implemented as explained in the previous sections. Combinations of PCA and SMOTE, PCA and oversampling, SMOTE and oversampling were tested with each of the models.

During this experiment, a very important pattern emerged. The performance of the models (measured in AUC ROC) is relatively constant when using thresholds of up to 72h, after this cut-off point, the performance of all the models falls sharply, until it is slightly better than chance at 120h.
In figure 31, the aforementioned pattern can be seen.

Figure 31. AUC ROC results for predictability threshold



After discussing this fact with the club's technical staff, the decision was made to implement the model predicting injuries with a time threshold of 72 hours until the time of injury.

6.9 Code

In the following link you may find all the necessary code to reproduce our results. https://drive.google.com/drive/folders/13KIP7IoET6K-lyXFwWMd3aCdKlUypa3t?usp=sharing

7. References

Adler, W., & Lausen, B. (2009). Bootstrap estimated true and false positive rates and ROC curve. *Computational statistics & data analysis*, *53*(3), 718–729.

Bahnsen, C., Alejandro, Aouada, Djamila, Ottersten, & Bjorn. (2015). Ensemble of Example-Dependent Cost-Sensitive Decision Trees. *arXiv e-prints*.

Batista, G. E. A. P. A., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, *6*(1), 20.

Bergmeir, C., Hyndman, R. J., & Koo, B. (2018). A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational statistics & data analysis*, *120*, 70–83.

Bittencourt, N. F. N., Meeuwisse, W. H., Mendonça, L. D., Nettel-Aguirre, A., Ocarino, J. M., & Fonseca, S. T. (2016). Complex systems approach for sports injuries: moving from risk factor identification to injury pattern recognition-narrative review and new concept. *British Journal of Sports Medicine*, *50*(21), 1309–1314.

Bowen, D., & Ungar, L. (2020). Generalized SHAP: Generating multiple types of explanations in machine learning. *arXiv*.

Brooks, J. H. M., Fuller, C. W., Kemp, S. P. T., & Reddin, D. B. (2005). Epidemiology of injuries in English professional rugby union: part 1 match injuries. *British Journal of Sports Medicine*, *39*(10), 757–766.

Bush, M., Barnes, C., Archer, D. T., Hogg, B., & Bradley, P. S. (2015). Evolution of match performance parameters for various playing positions in the English Premier League. *Human Movement Science*, *39*, 1–11.

Charest, J., & Grandner, M. A. (2020). Sleep and athletic performance: impacts on physical performance, mental performance, injury risk and recovery, and mental health. *Sleep medicine clinics*, *15*(1), 41–57.

Chawla, N. V. (2003). C4. 5 and imbalanced data sets: investigating the effect of sampling method, probabilistic estimate, and decision tree structure. *Proceedings of the ICML*.

Chmura, P., Konefał, M., Chmura, J., Kowalczuk, E., Zając, T., Rokita, A., & Andrzejewski, M. (2018). Match outcome and running performance in different intensity ranges among elite soccer players. *Biology of sport / Institute of Sport*, *35*(2), 197–203.

Christoph Molnar. (2020). *Interpretable Machine Learning*. (---, Ed.).

Cunniffe, B., Hore, A. J., Whitcombe, D. M., Jones, K. P., Baker, J. S., & Davies, B. (2010). Time course of changes in immuneoendocrine markers following an international rugby game. *European Journal of Applied Physiology*, *108*(1), 113–122.

Cunniffe, B., Proctor, W., Baker, J. S., & Davies, B. (2009). An evaluation of the physiological demands of elite rugby union using global positioning system tracking software. *Journal of Strength and Conditioning Research*, *23*(4), 1195–1203.

David Collett. (2015). *Modeling survival data in medical research*. (---, Ed.).

Doshi-Velez, F., & Kim, B. (2017). Towards A Rigorous Science of Interpretable Machine Learning. *arXiv*.

Du, M., Liu, N., & Hu, X. (2019). Techniques for interpretable machine learning. *Communications of the ACM*, *63*(1), 68–77.

Ekstrand, J. (2013). Keeping your top players on the pitch: the key to football medicine at a professional level. *British Journal of Sports Medicine*, *47*(12), 723–724.

Enright, K., Green, M., Hay, G., & Malone, J. J. (2020). Workload and injury in professional soccer players: role of injury tissue type and injury severity. *International journal of sports medicine*, *41*(2), 89–97.

Fernandes, R., Brito, J. P., Vieira, L. H. P., Martins, A. D., Clemente, F. M., Nobari, H., Reis, V. M., et al. (2021). In-Season Internal Load and Wellness Variations in Professional Women Soccer Players: Comparisons between Playing Positions and Status. *International Journal of Environmental Research and Public Health*, *18*(23).

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., & Herrera, F. (2018). Cost-Sensitive Learning. *Learning from Imbalanced Data Sets* (pp. 63–78). Cham: Springer International Publishing.

Gabbett, T. J. (2010). The Development and Application of an Injury Prediction Model for Noncontact, Soft-Tissue Injuries in Elite Collision Sport Athletes. *Journal of Strength and Conditioning Research*, *24*(10), 2593–2603.

Garreth Jamws, Daniela Witten, Trevor Hastie, & Robert Tibshirani. (2021). *An Introduction to Statistical Learning: with Applications in R* . (---, Ed.).

Haddad, M., Stylianides, G., Djaoui, L., Dellal, A., & Chamari, K. (2017). Session-RPE Method for Training Load Monitoring: Validity, Ecological Usefulness, and Influencing Factors. *Frontiers in Neuroscience*, *11*, 612.

Hägglund, M., Waldén, M., Magnusson, H., Kristenson, K., Bengtsson, H., & Ekstrand, J. (2013). Injuries affect team performance negatively in professional football: an 11-year follow-up of the UEFA Champions League injury study. *British Journal of Sports Medicine*, *47*(12), 738–742.

Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., & Bing, G. (2017). Learning from class-imbalanced data: Review of methods and applications. *Expert systems with applications*, *73*, 220–239.

He, H., & Shen, X. (n.d.). A Ranked Subspace Learning Method for Gene Expression Data Classification.

Impellizzeri, F. M., Rampinini, E., Coutts, A. J., Sassi, A., & Marcora, S. M. (2004). Use of RPE-based training load in soccer. *Medicine and Science in Sports and Exercise*, *36*(6), 1042–1047.

Impellizzeri, F. M., Woodcock, S., Coutts, A. J., Fanchini, M., McCall, A., & Vigotsky, A. D. (2021). What role do chronic workloads play in the acute to chronic workload ratio? time to dismiss ACWR and its underlying theory. *Sports medicine (Auckland, N.Z.)*, *51*(3), 581–592.

Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study1. *Intelligent Data Analysis*, *6*(5), 429–449.

Japkowicz, N. (2013). Assessment metrics for imbalanced learning. In H. He & Y. Ma (Eds.), *Imbalanced learning: foundations, algorithms, and applications* (pp. 187–206). Hoboken, NJ, USA: John Wiley & Sons, Inc.

Johnston, R. D., Gabbett, T. J., & Jenkins, D. G. (2013). Influence of an intensified competition on fatigue and match performance in junior rugby league players. *Journal of science and medicine in sport / Sports Medicine Australia*, *16*(5), 460–465.

Joshi, M. V., Kumar, V., & Agarwal, R. C. (2001). Evaluating boosting algorithms to classify rare classes: comparison and improvements. *Proceedings 2001 IEEE International Conference on Data Mining* (pp. 257–264). Presented at the 2001 IEEE International Conference on Data Mining, IEEE Comput. Soc.

Khalid, S., Khalil, T., & Nasreen, S. (2014). A survey of feature selection and feature extraction techniques in machine learning. *2014 Science and Information Conference* (pp. 372–378). Presented at the 2014 Science and Information Conference (SAI), IEEE.

Laurikkala, J. (2001). Improving identification of difficult small classes by balancing class distribution. In S. Quaglini, P. Barahona, & S. Andreassen (Eds.), *Artificial intelligence in medicine*, Lecture notes in computer science (Vol. 2101, pp. 63–66). Berlin, Heidelberg: Springer Berlin Heidelberg.

Liu, X.-Y., Wu, J., & Zhou, Z.-H. (2009). Exploratory undersampling for class-imbalance learning. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, *39*(2), 539–550.

Liu, Y., Wang, Y., Ren, X., Zhou, H., & Diao, X. (2019). A classification method based on feature selection for imbalanced data. *IEEE access : practical innovations, open solutions*, *7*, 81794–81807.

Maupin, D., Schram, B., Canetti, E., & Orr, R. (2020). The relationship between acute: chronic workload ratios and injury risk in sports: A systematic review. *Open access journal of sports medicine*, *11*, 51–75.

McCall, A., Dupont, G., & Ekstrand, J. (2016). Injury prevention strategies, coach compliance and player adherence of 33 of the UEFA Elite Club Injury Study teams: a survey of teams' head medical officers. *British Journal of Sports Medicine*, *50*(12), 725–730.

McLaren, S. J., Macpherson, T. W., Coutts, A. J., Hurst, C., Spears, I. R., & Weston, M. (2018). The Relationships Between Internal and External Measures of Training Load and Intensity in Team Sports: A Meta-Analysis. *Sports medicine (Auckland, N.Z.)*, *48*(3), 641–658.

McLellan, C. P., Lovell, D. I., & Gass, G. C. (2010). Creatine kinase and endocrine responses of elite players pre, during, and post rugby league match play. *Journal of Strength and Conditioning Research*, *24*(11), 2908–2919.

McLellan, C. P., Lovell, D. I., & Gass, G. C. (2011). Markers of postmatch fatigue in professional Rugby League players. *Journal of Strength and Conditioning Research*, *25*(4), 1030–1039.

Meeusen, R., Duclos, M., Foster, C., Fry, A., Gleeson, M., Nieman, D., Raglin, J., et al. (2013). Prevention, diagnosis and treatment of the overtraining syndrome: Joint consensus statement of the European College of Sport Science (ECSS) and the American College of Sports Medicine (ACSM). *European journal of sport science*, *13*(1), 1–24.

Merrick, L., & Taly, A. (2020). The explanation game: explaining machine learning models using shapley values. In A. Holzinger, P. Kieseberg, A. M. Tjoa, & E. Weippl (Eds.), *Machine Learning and Knowledge Extraction: 4th IFIP TC 5, TC 12, WG 8.4, WG 8.9, WG 12.9 International Cross-Domain Conference, CD-MAKE 2020, Dublin,*

Ireland, August 25–28, 2020, Proceedings, Lecture notes in computer science (Vol. 12279, pp. 17–38). Cham: Springer International Publishing.

Michael A Leeds, Peter von Allmen, & Victor A. Matheson. (2018). *The Economics of Sports*. (----, Ed.).

Moghaddasi, Z., Jalab, H. A., Md Noor, R., & Aghabozorgi, S. (2014). Improving RLRN image splicing detection with the Use of PCA and kernel PCA. *Thescientificworldjournal*, *2014*, 606570.

Moreo, A., Esuli, A., & Sebastiani, F. (2016). Distributional random oversampling for imbalanced text classification. *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval - SIGIR '16* (pp. 805–808). Presented at the the 39th International ACM SIGIR conference, New York, New York, USA: ACM Press.

Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R., & Yu, B. (2019). Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences of the United States of America*, *116*(44), 22071–22080.

Mylonas, K., Angelopoulos, P., Tsepis, E., Billis, E., & Fousekis, K. (2021). Soft-Tissue Techniques in Sports Injuries Prevention and Rehabilitation. In R. Taiar (Ed.), *Contemporary advances in sports science*. IntechOpen.

Ng, A. Y. (2004). Feature selection,L_1 vs.L_2 regularization, and rotational invariance. *Twenty-first international conference on Machine learning - ICML '04* (p. 78).

Presented at the Twenty-first international conference, New York, New York, USA: ACM Press.

Nielsen, R. O., Bertelsen, M. L., Ramskov, D., Møller, M., Hulme, A., Theisen, D., Finch, C. F., et al. (2018). Time-to-event analysis for sports injury research part 2: time-varying outcomes. *British Journal of Sports Medicine*, *53*(1), 70–78.

Oliveira, R., Brito, J. P., Moreno-Villanueva, A., Nalha, M., Rico-González, M., & Clemente, F. M. (2021). Reference values for external and internal training intensity monitoring in young male soccer players: A systematic review. *Healthcare (Basel)*, *9*(11).

Raghuwanshi, B. S., & Shukla, S. (2020). SMOTE based class-specific extreme learning machine for imbalanced learning. *Knowledge-Based Systems*, *187*, 104814.

Rodríguez-Pérez, R., & Bajorath, J. (2020). Interpretation of Compound Activity Predictions from Complex Machine Learning Models Using Local Approximations and Shapley Values. *Journal of Medicinal Chemistry*, *63*(16), 8761–8777.

Serg Masís. (2021). *Interpretable Machine Learning with Python*. (---, Ed.).

Seshadri, D. R., Thom, M. L., Harlow, E. R., Gabbett, T. J., Geletka, B. J., Hsu, J. J., Drummond, C. K., et al. (2020). Wearable technology and analytics as a complementary toolkit to optimize workload and to reduce injury burden. *Frontiers in Sports and Active Living*, *2*, 630576.

Taylor, T., West, D. J., Howatson, G., Jones, C., Bracken, R. M., Love, T. D., Cook, C. J., et al. (2015). The impact of neuromuscular electrical stimulation on recovery after intensive, muscle damaging, maximal speed training in professional team sports

players. *Journal of science and medicine in sport / Sports Medicine Australia*, *18*(3), 328–332.

Thai-Nghe, N., Gantner, Z., & Schmidt-Thieme, L. (2010). Cost-sensitive learning methods for imbalanced data. *The 2010 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8). Presented at the 2010 International Joint Conference on Neural Networks (IJCNN), IEEE.

Trevor Hastie, Robert Tibshirani, & Jerome Friedman. (2016). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. (---, Ed.).

Weiss, Gary M. (2004). Mining with rarity. *ACM SIGKDD Explorations Newsletter*, *6*(1), 7.

Weiss, G M, & Provost, F. (2003). Learning when training data are costly: the effect of class distribution on tree induction. *Journal of Artificial Intelligence Research*, *19*, 315–354.

Woods, K. S., Solka, J. L., Priebe, C. E., Kegelmeyer, W. P., Doss, C. C., & Bowyer, K. W. (1994). Comparative evaluation of pattern recognition techniques for detection of microcalcifications in mammography. *State of the art in digital mammographic image analysis*, Series in machine perception and artificial intelligence (Vol. 9, pp. 213–231). WORLD SCIENTIFIC.

Yang, J. (2021). Fast TreeSHAP: Accelerating SHAP Value Computation for Trees. *arXiv*.

Yata, K., & Aoshima, M. (2010). Effective PCA for high-dimension, low-sample-size data with singular value decomposition of cross data matrix. *Journal of multivariate analysis*, *101*(9), 2060–2077.

Zhao, Q., & Hastie, T. (2019). Causal interpretations of black-box models. *Journal of business & economic statistics : a publication of the American Statistical Association*, *2019*.

Zhou, Z.-H. (2011). Cost-Sensitive Learning. In V. Torra, Y. Narakawa, J. Yin, & J. Long (Eds.), *Modeling decision for artificial intelligence*, Lecture notes in computer science (Vol. 6820, pp. 17–18). Berlin, Heidelberg: Springer Berlin Heidelberg.

Zhou, Z. H., & Liu, X. Y. (2005). Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on knowledge and data …*.