Applying ML to assess fatigue and prevent injury in high performance swimming athletes

Hugo Veríssimo
Foundations of Machine Learning 24/25
University of Aveiro
Aveiro, Portugal
hugoverissimo@ua.pt

João Cardoso
Foundations of Machine Learning 24/25
University of Aveiro
Aveiro, Portugal
joaopcardoso@ua.pt

***Abstract*—temos que arranjar refs para ir metendo por ai**

Não esquecer de definir abreviaturas na primeira ocorrência e usá-las daí em diante.

I. Introduction

The evolution in performance of high level athletes is highly dependent on their skill, motivation, and discipline. With the support of a knowledgeable coach, the evolution can be substantially improved, through careful tailoring of the training regimen. One of the most relevant metrics since the dawn of structure training in sports is the feedback from the athlete, commonly described as the rate of perceived exertion (RPE). This single metric comprises the athletes analysis and intuition of the effort that was carried and how ready they feel for another bout of training. In recent years, more and more sports coaches have relied heavily on collected data to better assess, plan and adjust the training plans of their athletes in a systematic way. This allows for a fine balance between intense workouts, that generate stronger stimuli for muscle development and sport specific skills, taking the balance between effort and fatigue as the crucial ratio to respect. Too high effort, may lead to injury, too low and some gains may be left on the table.

In the scope of the first project for Foundations of Machine Learning, we decided to partner with the local sports club CAPGE (Clube Associação de Pais da Gafanha da Encarnação) to treat the data (kindly shared by Sr Coach Daniel Tavares) for estimating the fatigue of several athletes. The data was curated and prepared to implement and fit several machine learning algorithms to estimate fatigue after workout.

With this approach, we aim to generalize the models for different athletes/sports, and make it available to the local club for implementation and further testing.

II. State of the Art

Over the past decade there have been significant improvements in the field of ML applied algorithms for sports' related applications. Here we present the most relevant work in fatigue and injury prediction, which is not specifically on the sport we're assessing in our dataset, which lead to additional interpretations from the works analyzed to our own case of study. In general, the problem of class imbalance is seen throughout the literature, and different solutions are proposed, such as data gathering and preprocessing, over sampling and under sampling, with SMOTE (synthetic minority over-sampling technique) being the most commonly used approach for over sampling. As early as 2010, Gabbet and colleagues modeled the risk of injury with a monodimensional approach using logistic regression, based on athletes rate of perceived exertion, showing that even with a monovariate approach to injury prediction useful results could be attained. In recent years, several authors have focused in alternative techniques such as Logistic Regression, Random Forest, Support Vector Machine, or Convolutional Neural Network on Multivariate Timeseries (the references for these papers can be found in the paper "A Narrative Review").

Besides model selection, feature engineering and selection is among the most debated topics. Several authors opt to include GPS data, metabolic consumption, mechanical load, RPE, detailed quantification of workloads, ratio between acute:chronice loads. Despite the multivariate imputation, data analysis often shows strong correlations between them, leading to over fitting problems (usually model independent).

In the work by Carey et al. (2018) different algorithms have been implemented to predict the risk of injury in an Australian football club. The data collection lasted for three seasons, consisting of absolute and relative training load metrics, derived from GPS, accelerometer, and RPE data. The prediction models used were regularized logistic regression, generalized estimating equations, random forests, and support vector machines, with periods of 3, 6, and 21 days (these periods have been studied and verified as adequate for the case of Australian football). The periods served to calculate moving averages and exponentially weighted moving averages (EWMA). The latter allowed to account for the decay in significance of the training load the further it happened from a given day, in accordance with the work from Williams et al. (2016). From the results it was possible to verify that over fitting was very likely due to the multicollinearity between variables, which was confirmed by principal component analysis (PCA). The use of PCA with regularized logistic regression slightly improved the results.

More recent studies have employed ensemble algorithms, in order to take most of the different learning models selected,

taking into account the need to balance the classes as is common practice for this type of problems.

In summary, the integration of machine learning techniques in sports fatigue and injury prediction has evolved from simple monovariate models to complex multivariate and ensemble approaches. Addressing challenges such as class imbalance, feature selection, and multicollinearity remains crucial for developing robust predictive models applicable across different sports contexts.

III. Methodology

The methodology defined for this project consisted of three major steps: assessment and curation of the dataset, feature engineering and selection of features, followed by data normalization; training of the selected machine learning models; finalizing with model evaluation and subsequent training until optimum results were achieved. Figure 1 illustrates the methodology used in this project.

After data normalization for all features (`StandardScaler`), the data was separated between training and testing data (80/20, respectively), using the (`train_test_split`) function from (`sklearn`). The seeds for randomization were kept consistent across models, to ensure reproducibility, and avoid biases towards any model. The ML models selected for the project were (`LogisticRegression`), (`SupportVectorMachine`), and (`DecisionTreeClassifier`), where the modeling approach and hyperparameters are detailed below. For the given hyperparameters available in each model the function (`RandomizedSearchCV`) was used with 8-fold cross-validation to minimize the risk of over fitting (the selected 8-fold CV was consistent throughout all the relevant stages, class weight estimation, training).

The metrics used to assess the different models are presented in Table I, and are consistent with those used in the literature.

TABLE I: Metrics for multiclass classification model evaluation.

Measure	**Formula**
Precision (per class i)	$\frac{TP_i}{TP_i + FP_i}$
Recall (per class i)	$\frac{TP_i}{TP_i + FN_i}$
F1-score (per class i)	$2 \cdot \frac{\text{Precision}_i \cdot \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i}$
Accuracy	$\frac{\sum_{i=1}^{C} TP_i}{\sum_{i=1}^{C} (TP_i + TN_i + FP_i + FN_i)}$

Given the class imbalance associated with the problem at hand, the importance of precision, recall and F1-score per class are especially relevant, along with the confusion matrix, to understand how the model is failing to correctly classify the various observations. The precision is the ratio of true positives for all the positives attributed (the higher, the better). Recall (or sensitivity) gives the ratio of true positives among true positives and false negatives, where lower values indicate a higher number of misidentified true positives. The F1-score provides a balanced assessment of the model, taking a harmonic mean of precision a recall, providing a particularly good way to assess datasets with imbalanced classes, while accuracy provides an overall assessment with the ratio of true positives and true negatives over the total predictions.

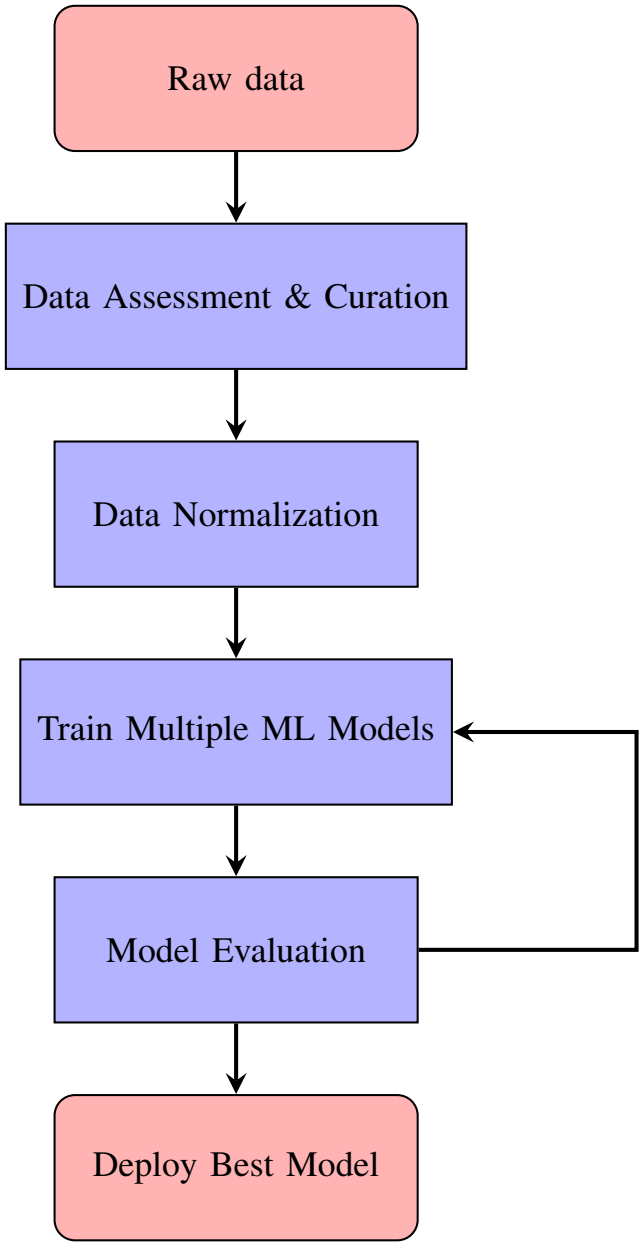


Fig. 1: Flowchart of the methodology for data processing and modeling. The evaluation results are used to refine the model training process.

The methods and ML models used are consistent with those in the literature, considering the type of features and target in this project. Moreover, during the model refinement stage a reassessment of past steps was carried in order to ensure no

gaps in the process. The programming language was Python, and the packages available therein, with notable mention for (`scikit-learn`).

IV. Dataset Analysis

A. Data Description

The data used in this project was collected from the swimming club CAPGE during the season of 2019/2020, where each athlete has several observations corresponding to training days, where each of the features was collected. Not all athletes logged the same number of training days, nor present an equal distribution between low, average and high levels of intense training. The names of the athletes were removed to ensure privacy and confidentiality, keeping only the gender as a variable. The team is comprised by seven athletes, three male and four female. Most of the features are related to feedback from the athletes on different aspects of their lives (i.e. sleep quality, appetite, and rate of perceived exertion after training), while others are measurable (i.e. workload, variation in heart rate before and after training, weight variation). A notable feature to mention is the RPE, that is still deemed as one of the most relevant metrics for workload planning and fatigue assessment. All these attributes are classified between 1 — 10, each value corresponding to increasingly 'worse' categories (e.g., 1 great appetite / normal, 10 no appetite at all).

The fatigue index is calculated from these features, using weights attributed by the coach based on his empirical experience. The resulting fatigue index is between 0 — 100, which was categorized in four classes as seen in Table II.

TABLE II: Classification of fatigue index into categories based on numerical ranges.

Range	Initial Classes	Final Classes
≥ 90	Risk	Risk/Caution
≥ 80	Caution	
≥ 40	Optimal	Optimal
< 40	Low/Minimal	Low/Minimal

There is a big gap between fatigue classes due to the nature of training and performing high effort workouts in specific times of the training cycle. The dataset was provided in Excel format (per athlete), from which we imported and combined the data as a pandas dataFrame to apply the different models.

B. Dataset curation

The initial assessment evidenced the need for balancing our data. To start, we've reduced the number of classes, by combining the two higher risk classes ('Caution' and 'Risk'). With this, the number of observations was closer between 'Low/Minimal' and 'Risk/Caution', leaving us with an excess of observations for 'Optimal', as seen in Figure 2.

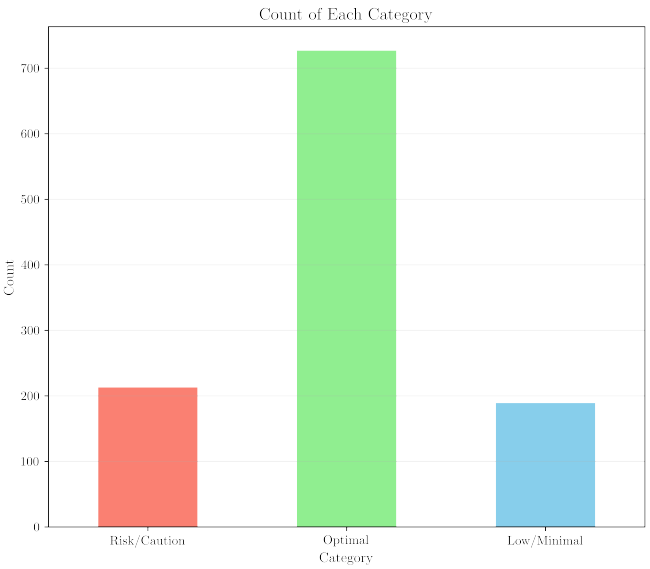


Fig. 2: Bar plot of the new classes, once 'Caution' and 'Risk' are combined into one.

At this stage, we opted to under sample our dataset to the number of observations of 'Risk/Caution', and over sample the observations in 'Low/Minimal', by imputing random samples from the pool of observations of 'Low/Minimal', ending up with 213 observations per class (regardless of gender). The use of SMOTE in this scenario would give continuous classes for our features, which wouldn't yield any physical meaning.

To assess how the different features vary among them and in relation to the target, we computed the correlation matrix as seen in Figure 3.

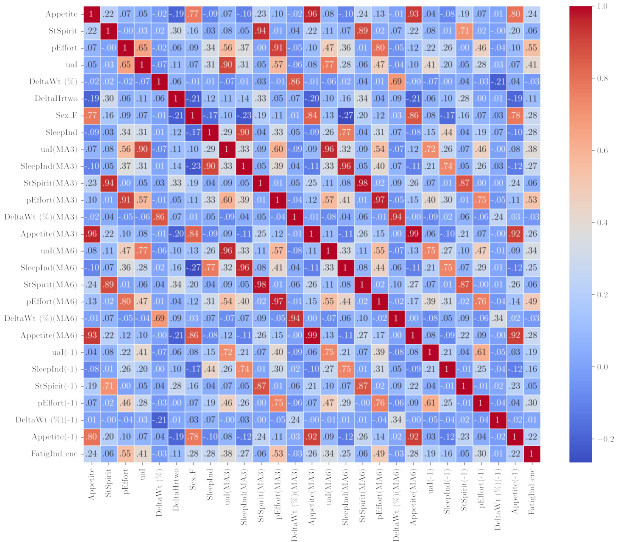
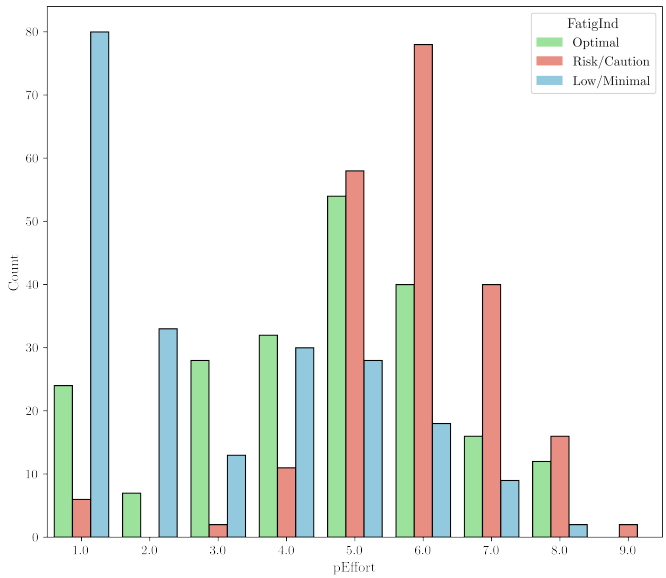


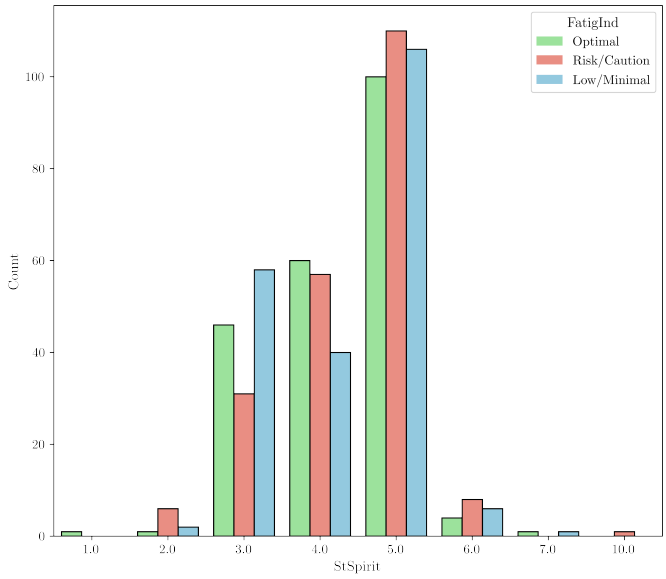
Fig. 3: Correlation matrix for all possible features considered for the models.

From the corr matrix it was possible to exclude several of the features, which was verified by how the classes are dis-

tributed across the scales for each feature. Figure 4 illustrates a proper and poor example of class distribution for the given features.



(a) Distribution of perceived effort across classes.



(b) Distribution of state spirit across classes.

Fig. 4: Class distributions across different metrics: (a) perceived effort, (b) state spirit.

Considering that the weights used in the coach's original estimation of fatigue were identical regardless of sex, we performed some simple models in order to decide if it would be necessary to split it. We could verify that gender didn't have a significant impact in model performance, so we opted to use it as a feature.

Figure 5 illustrates the periodicity of higher training loads and subsequent lower intensity periods. It is important to refer that once the two higher intensity classes were combined the loss of granularity of how fatigue changes throughout the season is evident.

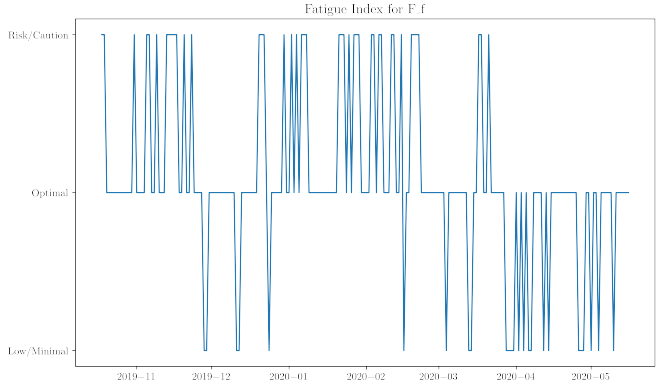


Fig. 5: Daily fatigue trends for athlete F_f.

In order to keep the time series nature of training, increase and decrease of training intensity, and varying fatigue with training we included exponentially weighted moving averages (EWMA) [1].

This way memory is introduced in our model, and allow to consider the decay in the weights of events further away from any given day. The mathematical expression used to calculate EWMA for each selected feature is,

$$\text{EWMA}_{\text{today}} = \text{Feature}_{\text{today}} \cdot \lambda_a + (1 - \lambda_a) \cdot \text{EWMA}_{\text{yesterday}}$$

where λ_a is a value between 0 and 1 that represents the degree of decay, with higher values discounting older observations at a faster rate. The λ_a is given by:

$$\lambda_a = \frac{2}{N+1}$$

At this point it was possible to select the most relevant features for modeling: (*pEffort*), (*uaI*), (*SleepInd*), (*Sex_F*), (*pEffort(MA6)*), (*SleepInd(MA6)*), (*uaI(MA6)*), (*Appetite(MA6)*).

V. Classification Models

In this section the results for the three machine learning models are presented and detailed in equal fashion, to facilitity the discussion and interpretation in the next section. As was mentioned in the methodology, the modeling approach was the same for all, and here the training dataset results are presented first, followed by the test dataset results.

VI. Logistic Regression

The logistic regression model was developed by setting different ranges for the hyperparameters, with C varying between 0.01 and 300, and allowing the selection of different cost functions ($L1$, $L2$, or none). In the end the best cost function to use was $L1$. Given the class imbalance we opted to determine the class weights for 'Risk/Caution' and 'Low/Minimal', ranging between 0.1 and 2 for both cases. The resulting estimation of highest accuracy can be seen in Figure 6.

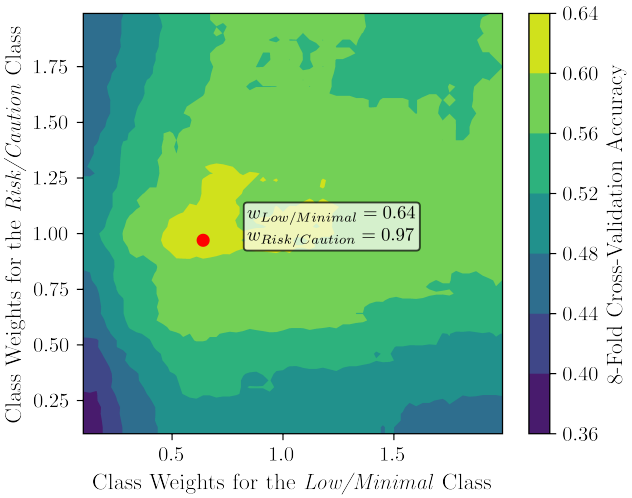


Fig. 6: Effect of the 'Risk/Caution' and 'Low/Minimal' classes weight on LogReg model accuracy.

From the figure above the weights selected were 0.64 and 0.97 for 'Low/Minimal' and 'Risk/Caution' respectively. These values were used throughout the training and test data onward. The remaining hyperparameters were set at C 2.13, cost function $L1$ and the solver $SAGA$ (Stochastic Average Gradient Augmented).

The learning curves from the training and the cross-validation converge after the largest training set size of 450, with no signs of over fitting.

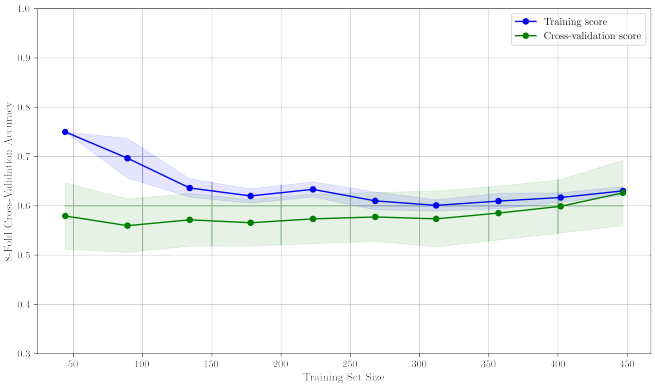


Fig. 7: LogReg model performance using learning curve representation across varying training data sizes.

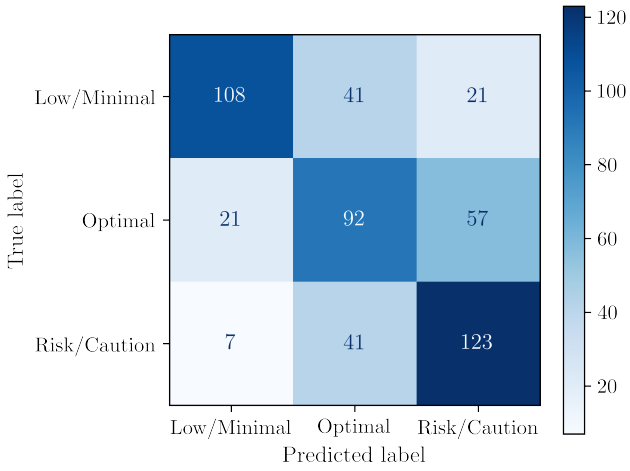


Fig. 8: Confusion matrix for the training data of LogReg model.

TABLE III: Classification report for LogReg model performance evaluation on training data.

Class	Precision	Recall	F1-Score	Support
Low/Minimal	0.79	0.64	0.71	170
Optimal	0.53	0.54	0.53	170
Risk/Caution	0.61	0.72	0.66	171
Accuracy			0.63	511
Macro avg	0.64	0.63	0.63	511
Weighted avg	0.64	0.63	0.63	511

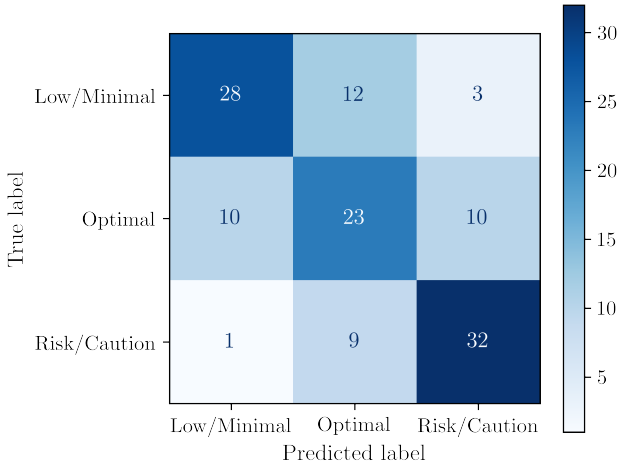


Fig. 9: Confusion matrix for the test data of LogReg model.

TABLE IV: Classification report for LogReg model performance evaluation on test data.

Class	Precision	Recall	F1-Score	Support
Low/Minimal	0.72	0.65	0.68	43
Optimal	0.52	0.52	0.53	43
Risk/Caution	0.71	0.76	0.74	42
Accuracy			0.65	128
Macro avg	0.65	0.65	0.65	128
Weighted avg	0.65	0.65	0.65	128

VII. Support Vector Machine

Given the increased complexity compared to LogReg, the support vector machine (SVM) model allows for manipulation of a larger number of hyperparameters, which consequently lead to longer computation times to achieve the best model. The hyperparameters used were Regularization Parameter (parameter that controls the penalty for misclassified training examples, i.e., cost function) (C),the Kernel Coefficient (γ), the kernel to be used, the highest degree possible (for a $poly$ kernel), and the value of the independent term (Coef_0), which controls the flexibility of the decision boundary. The ranges of possible values can be assessed in Table V.

TABLE V: SVM model hyperparameters search space.

Hyperparameter	Possible Values
C	Uniform(0, 100)
γ	{scale, auto, 0.1, 0.01, 0.001}
Kernel	{linear, rbf, poly, sigmoid}
Degree	{1, 2, 3}
Coef_0	Uniform(-5, 5)

As in the case for LogReg, an ideal class weight was estimated, however here only for 'Optimal' class, as it was the worst predicted class even at a training stage, with an equal split between 'Low/Minimal' and 'Risk/Caution'. The weight for the 'Optimal' class that retrieved the highest accuracy was estimated as shown in Figure 10. The remaining classes kept the initially attributed unitary weight. The best parameters found were C 6.93 $Coef_0$ -0.20, $degree$ 3, γ auto, and $Kernel$ RBF (Radial Basis Function).

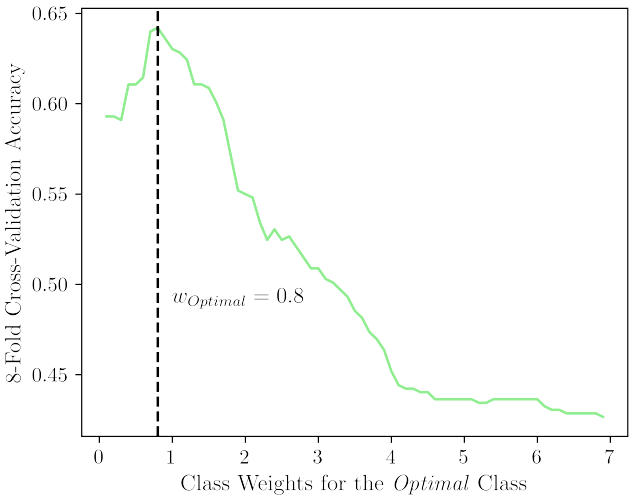


Fig. 10: Effect of the 'Optimal' class weight on SVM model accuracy.

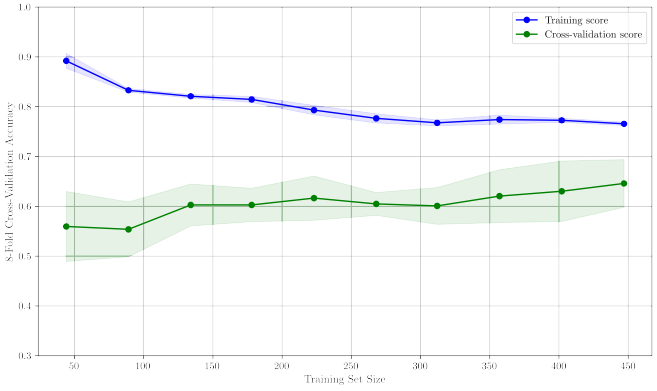


Fig. 11: SVM model performance using learning curve representation across varying training data sizes.

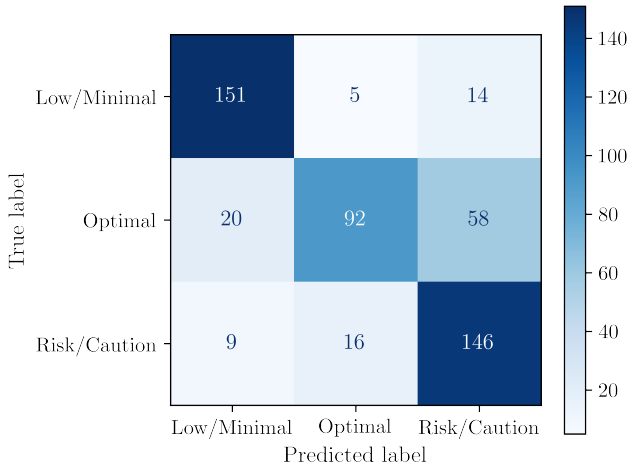


Fig. 12: Confusion matrix for the training data of LogReg model

TABLE VI: Classification report for SVM model performance evaluation on training data.

Class	Precision	Recall	F1-Score	Support
Low/Minimal	0.84	0.89	0.86	170
Optimal	0.81	0.54	0.65	170
Risk/Caution	0.67	0.85	0.75	171
Accuracy			0.76	511
Macro avg	0.77	0.76	0.75	511
Weighted avg	0.77	0.76	0.75	511

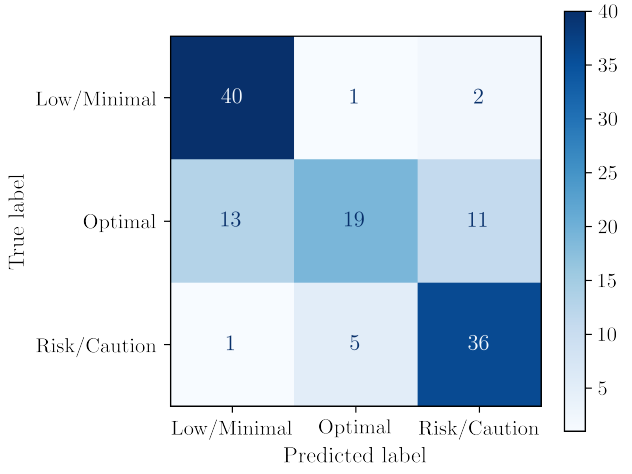


Fig. 13: Confusion matrix for the test data of SVM model

TABLE VII: Classification report for SVM model performance evaluation on test data.

Class	Precision	Recall	F1-Score	Support
Low/Minimal	0.74	0.93	0.82	43
Optimal	0.76	0.44	0.56	43
Risk/Caution	0.73	0.86	0.79	42
Accuracy			0.74	128
Macro avg	0.75	0.74	0.72	128
Weighted avg	0.75	0.74	0.72	128

VIII. DECISION TREE MODEL

The decision tree model was developed considering the hyperparameters available and setting ranges for the possible values to be estimated. The ranges were selected taking in consideration the need to minimize the risk of over fitting. The split criteria are among the most commonly used, and the remaining ranges considered for the hyperparameters are presented in Table VIII.

TABLE VIII: Decision tree model hyperparameters search space.

Hyperparameter	Possible Values
Split Criterion	{gini, entropy}
Max Depth	$[2, \ldots, 8]$
Min Samples to Split	$[5, \ldots, 20]$
Min Samples per Leaf	$[3, 4, \ldots, 10]$

The split criterion selected was $entropy$, $min_samples_split$ 11, $min_samples_leaf$ 7, and the max_depth 4.

Considering the entropy of a node (given by the equation below), the goal is to maximize the information gain by assessing each node, with the objective to make them more homogeneous.

- $H(\text{node}) = -\sum_{\text{Class }_j} p(\text{Class}_j|\text{node}) \log p(\text{Class}_j|\text{node})$

The optimized DTree resulted in the structure as seen in Figure 14.

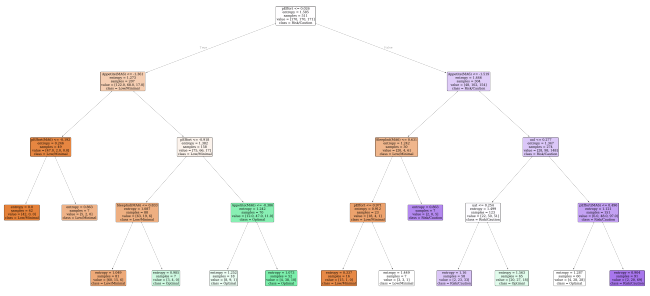


Fig. 14: Visualization of the decision tree model after optimization using randomized search.

The decision paths are characterized by the certainty in each node, highlighted by the color saturation for each of the classes ('Low/Minimal', 'Optimal', 'Risk/Caution'). The diminished

proportion of end nodes with higher color saturation for 'Optimal' when comparing with the other classes is noteworthy, and consistent with the observations for the remaining models.

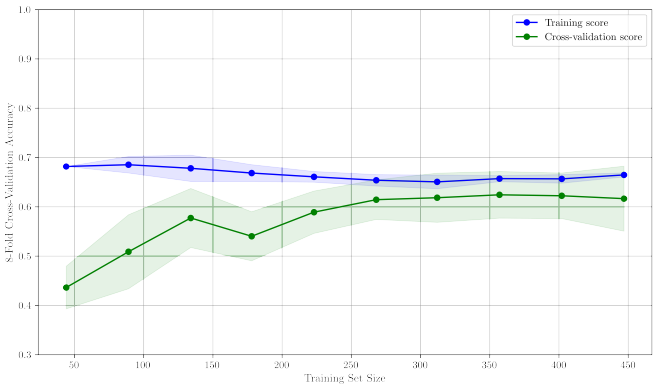


Fig. 15: Analysis of decision tree model performance using learning curve representation across varying training data sizes.

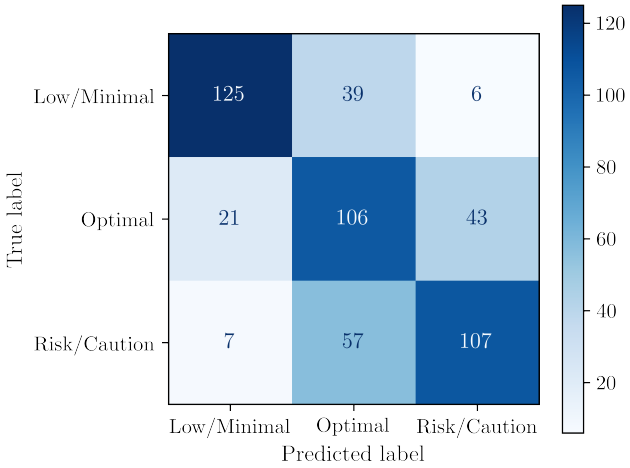


Fig. 16: Evaluation of decision tree model performance on training data performance via confusion matrix.

TABLE IX: Classification report for decision tree model performance evaluation on training data.

Class	Precision	Recall	F1-Score	Support
Low/Minimal	0.82	0.74	0.77	170
Optimal	0.52	0.62	0.57	170
Risk/Caution	0.69	0.63	0.65	171
Accuracy			0.66	511
Macro avg	0.68	0.66	0.67	511
Weighted avg	0.68	0.66	0.67	511

A. *resultados de teste*

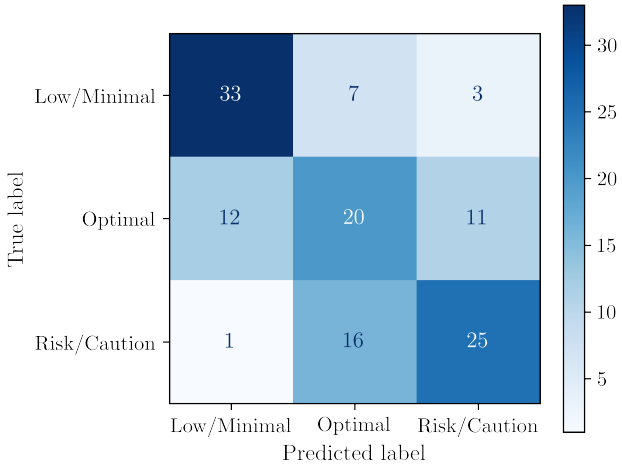


Fig. 17: Evaluation of decision tree model performance on test data performance via confusion matrix.

TABLE X: Classification report for decision tree model performance evaluation on test data.

Class	Precision	Recall	F1-Score	Support
Low/Minimal	0.72	0.77	0.74	43
Optimal	0.47	0.47	0.47	43
Risk/Caution	0.64	0.60	0.62	43
Accuracy			0.61	128
Macro avg	0.61	0.61	0.61	128
Weighted avg	0.61	0.61	0.61	128

IX. COMPARACAO DOS MODELOS E RESULTADOS

comparar aqui todos os resultados e isso

acuracaia, learning curves, etc , etc

- learning curves

- scalability

- performance

- cross validation

observa se q o melhor é tal e tal, mas aquele fez isto e ns q

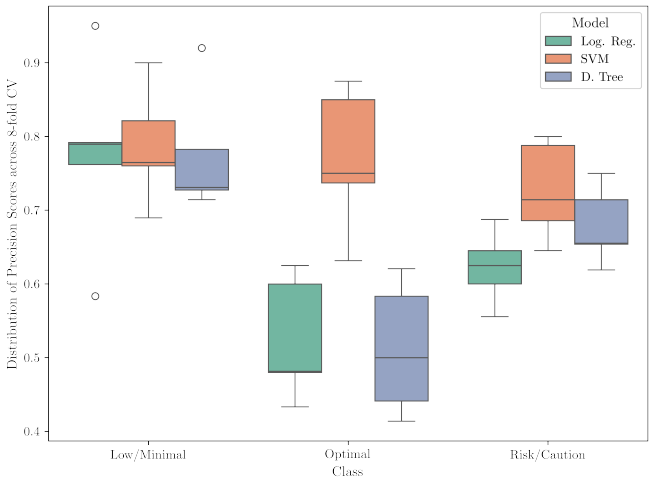


Fig. 18: CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION

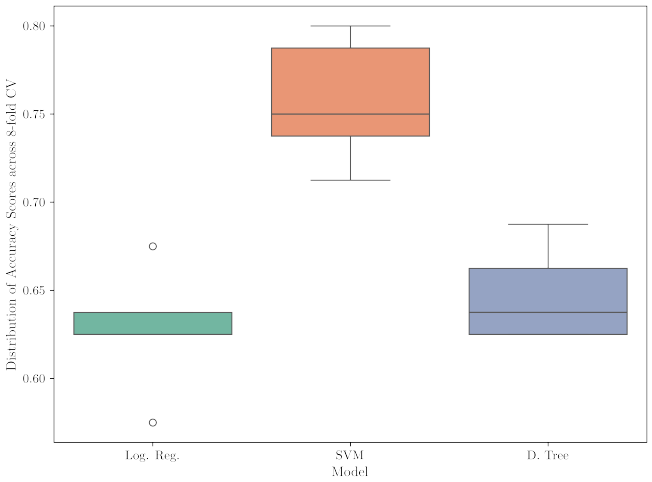


Fig. 20: CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION

lalal pca

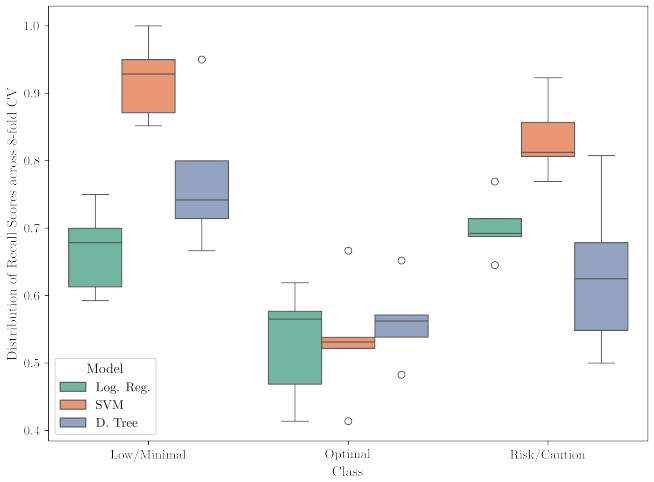


Fig. 19: CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION

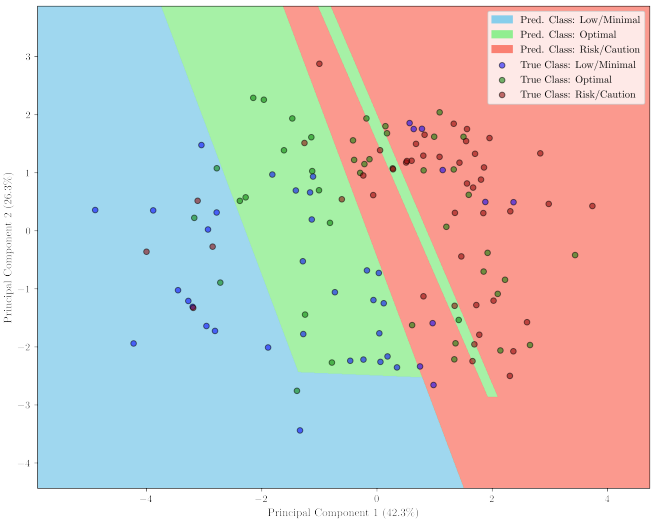


Fig. 21: CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION

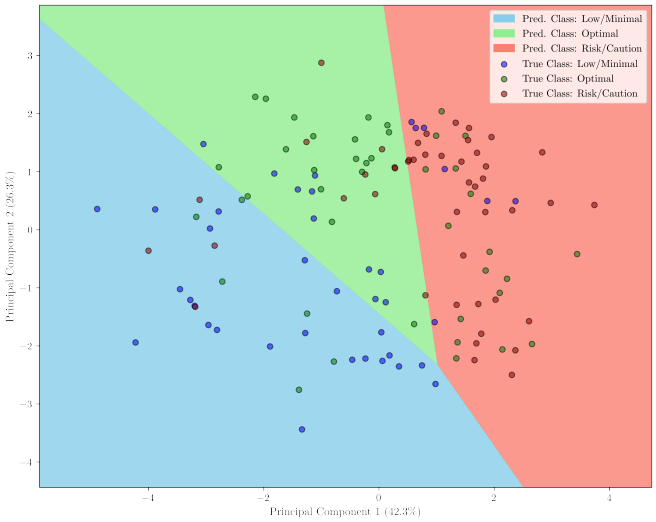


Fig. 22: CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION

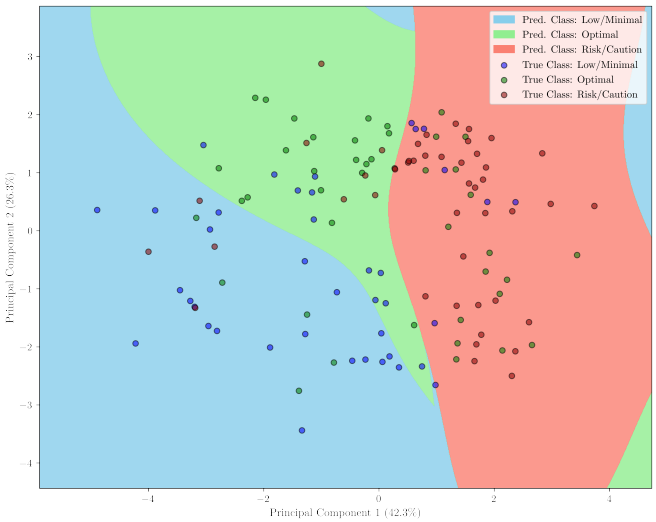


Fig. 23: CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION CAPTION

X. Conclusion

The conclusion goes here.

Acknowledgment

The authors would like to thank... professora petia e treinador e organizacao do clue

XI. NOTAS

Discussão dos resultados: - Importante considerar os kernels selecionados, e o que isso pode representar dado o tamanho reduzido do nosso dataset

-¿ Há coerência entre métricas (F1-score Test vs Train está à mesma distância quase sempre)

-¿ Confusion matrix destaca SVM, em que apenas sofre na atribuição da classe real do 'Optimal', mas tende a pender mais na fronteira Optimal/Risk

-¿ Boxplots, comparação entre modelos tendo em vista a disparidade de performance entre classes (particularmente para LogReg e DTree)

-¿ Facto de o nosso dataset ter dados repetidos, faz com que regra geral haja uma maior variabilidade em termos de scores (já que o dataset de teste poderá ter uma maior fração de valores que já foram avaliados no dataset de treino)

-¿ Mais do que tudo, plot das classes sobre os clusters com PCA mostra quão difícil é agregar os pontos com true optimal

-¿ Fronteiras complexas, particularmente no caso da DTree, indicando necessidade de aumento de dimensionalidade para encontrar fronteiras de separação (incorrendo o risco de over fitting)

-¿ Temos algumas zonas em branco ao longo do documento, temos de ver como resolver (não sei se será mais texto em alguma zona, mas gosto pouco de adicionar palha)

-¿ Movi o que tinhas nos modelos aqui para baixo para ver como começava a ficar distribuído

-¿ Novelty and contributions (3) Compare your solution with the works of other authors (published references) , try to propose a better solution, e.g. improve the performance of the ML model in solving the problem you work with.

SVM

aplicando novamente a otimizacao do modelo SVM , tendo em conta estes pesos, foram obtidos resultados mais satisfatorios

Não introduzi os parâmetros como tabela, porque varia de modelo para modelo e achei que não fosse ficar tão bem. Queres indicar as expressões para todos os kernel ou só discutir brevemente? É que se não começamos a ter pano para mangas, particularmente só com o SAGA....

Nesta parte não introduziria mais texto, deixava de maneira mais simples e ilustrativa como no caso do LogReg

QUAL FOI OS HYPERPAREMTES OBTIDOS

Os parametros usados foram ent:

slides da prof

- kernel: Radial Basis Function (rbf) (Gaussian kernel) $\exp(-\gamma||x - x'||^2)$ e pelos slides da prof $\gamma = 1/\sigma$

- gamma: auto $(1/n_{feautures})$

- $C \approx 6.93$

- weights (serve para dar a importancia da penalicao: C * weight torna-se a penalizacao):

- optimal 0.8

- low/minimal: 1

- risk: 1

DTREE

QUAL FOI OS HYPERPAREMTES OBTIDOS

os paraemtros usados foram ent

- max depth: 4 (The deeper the tree, the more splits it has and it captures more information about the data)

- min sample split: 11 (When we increase this parameter, the tree becomes more constrained as it has to consider more samples at each node.)

- min samples leaf: 7 (This parameter is similar to min samples splits, however, this describe the minimum number of samples of samples at the leafs, the base of the tree.)

splitter: best (to choose the best split at each node instead of 'random')

criterion: entropy (The function to measure the quality of a split. Shannon information gain)

Considerar usar dataset com informação combinada de atletas para reter carácter temporal (progressão da época, aumento de cargas e combinação com features que tenham referência temporal [carga de treino do dia anterior]

Modelos sugeridos: Random Forest / SVM

temos com objetivo generalizar para que qq treinador possa ter nocao do comporamento e do estado dos seus atletas, podendo ajustar os seus treinos e cargas físicas consoante as medidas de fadigas.

para alem disso tbm se quer ver quais as metricas mais importantes relacionadas coma a fadiga

De acordo com a conversa com a professora é importante perceber como é que se deve definir a memória da nossa t-SNE, e a importância que isso tem no período da fadiga.

ns q dados fornecidos por um treinador de natacao, tivemos de organizar os dados em folhas de excel, visto estarem por linhas e com graficos e formulas de acordo com o treinador, bla bla, teve-se fazer oq? sabes melhor q eu pq foste tu q fizeste

os valores da fadiga foram convertidos para categoricos, pq e mais interessante classificar a fadiga, tendo em conta intervalos dados pelo treinador, do q numero que tornam mais dificl a interpretacao dos mesmos

de seguida agruparam-se os dados todos num novo ficheiro excel, para posterior analise atraves do python, de forma mais facilitata

tem se entao dados diarios, para X atletas, durante ns q tempo, ao longo da epoca tal, totalizando x observacoes (linhas)

para cada observacao tem se variaveis como ... bla bla e bla, contudo, nem todas serao usadas, devido a insights dados pelo treinador aka expert

as variaveis que irao ser utilizadas no estudo são tal tal, que representa tal, tal tal, ... ns q ns q mais.

[1] better way chronic load

[2] Predictive Modelling of Training Loads and Injury in Australian Football Carey 2018

[3] another review (Prediction models for musculoskeletal injuries in professional sporting activities: A systematic review)

[4] Murray et al. (2016

REFERENCES

[1] S. Williams, S. West, M. J. Cross, and K. A. Stokes, "Better way to determine the acute:chronic workload ratio?" *British Journal of Sports Medicine*, vol. 51, no. 3, pp. 209–210, 2017. [Online]. Available: https://bjsm.bmj.com/content/51/3/209

[2] D. L. Carey, K. Ong, R. Whiteley, K. M. Crossley, J. Crow, and M. E. Morris, "Predictive modelling of training loads and injury in australian football," *International Journal of Computer Science in Sport*, vol. 17, no. 1, p. 49–66, Jul. 2018. [Online]. Available: http://dx.doi.org/10.2478/ijcss-2018-0002

[3] D. Seow, I. Graham, and A. Massey, "Prediction models for musculoskeletal injuries in professional sporting activities: A systematic review," *TRANSLATIONAL SPORTS MEDICINE*, vol. 3, no. 6, pp. 505–517, 2020. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/tsm2.181

[4] N. Murray, T. Gabbett, A. Townshend, and P. Blanch, "Calculating acute: Chronic workload ratios using exponentially weighted moving averages provides a more sensitive indicator of injury likelihood than rolling averages," *British journal of sports medicine*, vol. 51, 12 2016.