

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

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 oversampling 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:chronic 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. 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 1.

TABLE I: Sample Table with Ranges and Classifications

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

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.

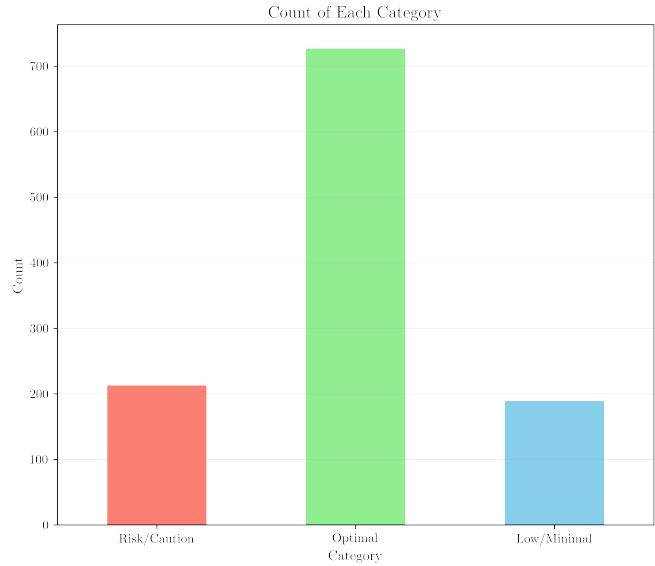


Fig. 1: Histogram of the new classes, once 'Caution' and 'Risk' are combined in one.

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'. The use of SMOTE in this scenario would provide 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 2.

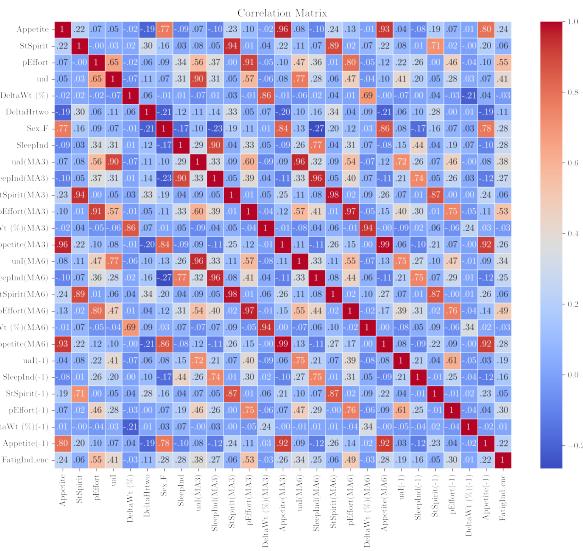
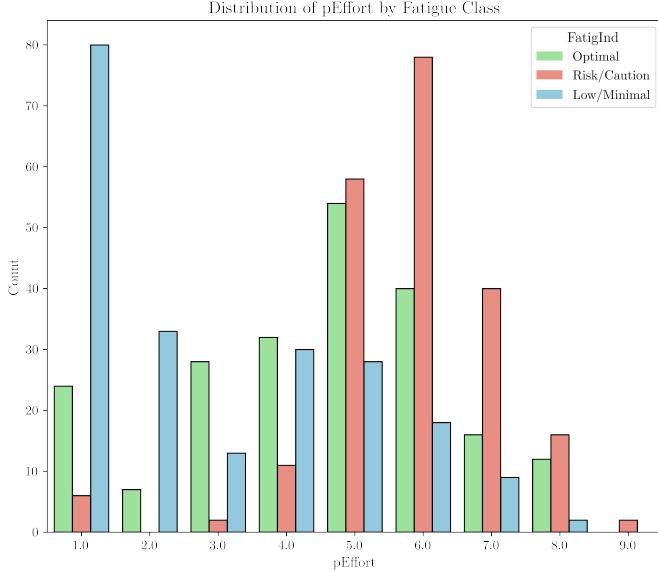
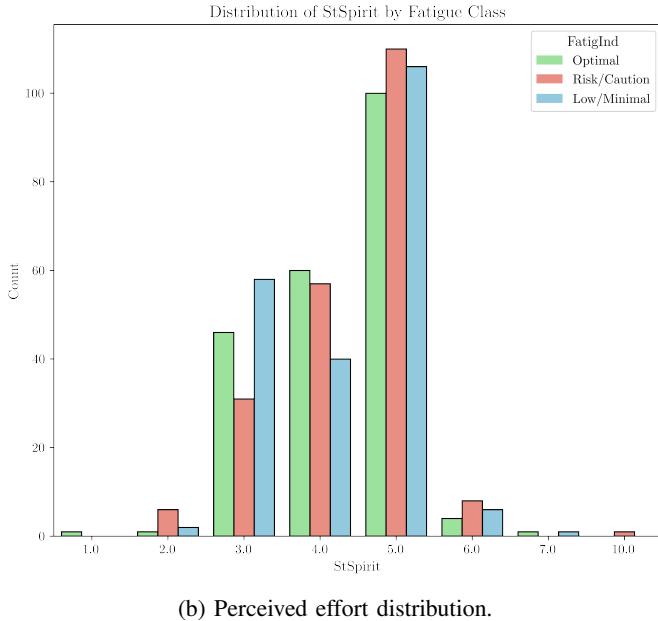


Fig. 2: hgfdgushfg

From this we could exclude several of the features, which are verified by how the classes are distributed across the scales for each feature. Figure illustrates a proper and poor example of class distribution for a given feature.



(a) Perceived effort distribution.



(b) Perceived effort distribution.

Fig. 3: Class distributions across various metrics: (a) Perceived effort, (b) Another metric.

Considering that the weights used in the dataset were identical regardless of sex, we performed some simple models in order to decide if it would be necessary to split our dataset. We could verify that gender didn't have a significant impact in model performance, so we opted to use it as a feature. In order to keep the timeseries nature of training, increase and decrease of training intensity, and varying fatigue with training we included exponentially weighted moving averages (EWMA). This way we include memory 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:

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

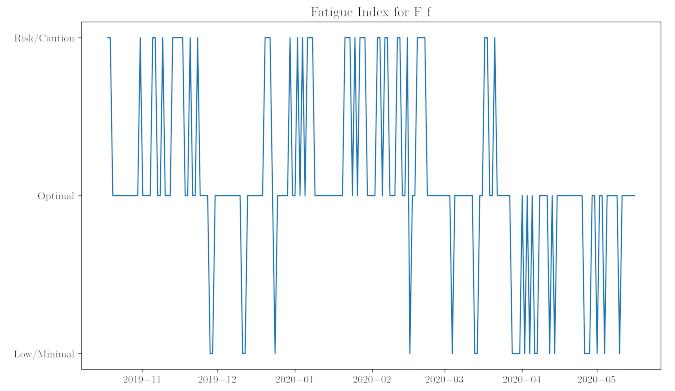


Fig. 4: Histogram of the new classes, once 'Caution' and 'Risk' are combined in one.

tentamos o smote mas ele criava variaveis continuas mesmo quanto eram categoricas entre outros problemas e por nao precisarmos de uma grande qnt relativa de obs geradas, ent usou se o metodo de "copiar obervacoes (procurar nome melhor)"

ONDE SE ENTRA O GRAFICO DAS CLASSES EQUILIBRADAS?

C. analisa das features

fazer descricao das feautures realmente usadas ?

ns q temos q standartizar os dados para se tornar mais facil a sua analise, para q esteja tudo na mesma medida bla bla

OU PASSAR ISSO TUDO PARA O CLASSIFICATION MODELS?

IV. CLASSIFICATION MODELS

para classificar a fadiga, vamos usar varios modelos de machine learning, aplicados através da linguagem de programacao python, pelo sua extensivas livrarias com funcoes dedicadas a este tipo de analise. nomeadamente reg logistica (LogisticRegression), support vector machine (SVC) e arvore de decisao (DecisionTreeClassifier), e experimentar variacoes dos mesmos, tendendo encontrar o que melhor modela os nossos dados, utilizando a funcao de otimizacao de modelos RandomizedSearchCV, com um cross validation de 8 para diminuir o risco de overfitting

uma componente mt importante deste estudo é o avaliar cada modelo, para isso serão usadas metricas tais como, para analisar modelo a modelo qual é o melhor para modelas/classificar os vários níveis de fadiga

mostrar algumas formulas e graficos de exemplo etc etc
para alem disso, importante referir q cada modelo sera treinado com 80% dos dados e bla bla train_test_split.

V. REGRESSAO LOGISTICA

AUSHDIUAHIAHFSIH

A. sem termo de castigo ou assim

ns q resultados bla bla

B. com termo de castigo ou assim

ns q mais resultados

meter tabelas com resultados e tudo mais, mas so no fim metemos uma geral a comparar tudo

meter:

- matriz de confusao
- precision score
- f1 score
- accuracy score
- recall score
- cross validation

VI. SUPPORT VECTOR MACHINE

para a otimizacao do modelo, foram testadas varias combinacoes de parametros diferentes, nomeadamente o Regularization Parameter ($1/\lambda$), o Kernel Coefficient (γ), o tipo de kernel, o grau maximo caso para o kernel do tipo ‘poly’ (degree) e o valor do termo independente ($Coef_0$), podendo-se obsevar entao um resumo? destas combinacoes na tabela II.

TABLE II: SVM Hyperparameter Search Space ?

Parameter	Distribution/Values
$1/\lambda$	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)

contudo, ao aplicar o melhor modelo obtido tendo em conta as combinacoes destes parametros, os resultados nao foram satisfatorios, tendo em conta q a classe optimal estava mt má

para isso foi decidido aplicar pesos às classes, fixando o peso de todas as classes com excecao da optimal, sendo este considerado agora como um hyperparametro

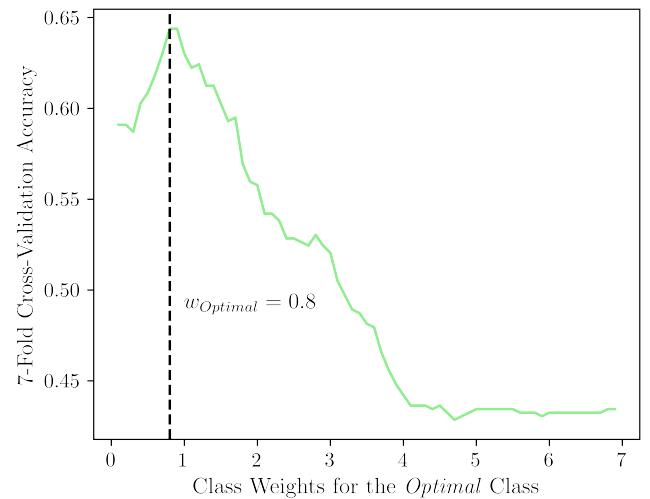


Fig. 5: Histogram of the new classes, once ‘Caution’ and ‘Risk’ are combined in one.

através da figura 5, verifica-se que o peso ótimo para esta classe é 0.8, quando as restantes classes têm peso 1.

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

QUAL FOI OS HYPERPARAMETES OBTIDOS

Best parameters found : 'C' : np.float64(6.9254306285859935), 'coef0' :

alguns ele n usa, tp o degree é ignorado, o coef0 se calhar tbm, ver melhor isso

A. resultados de treino

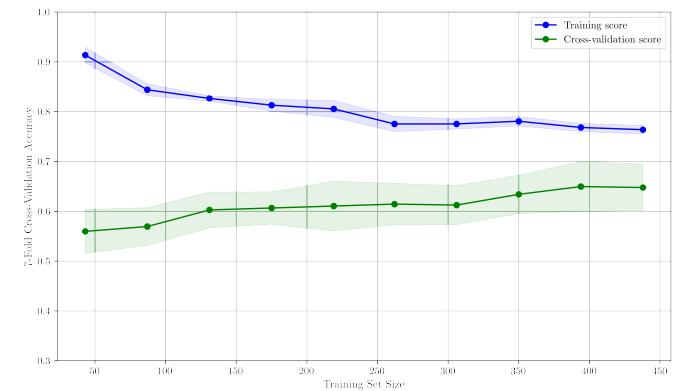


Fig. 6: Histogram of the new classes, once ‘Caution’ and ‘Risk’ are combined in one.

TABLE IV: Classification Report for Model Evaluation (teste)

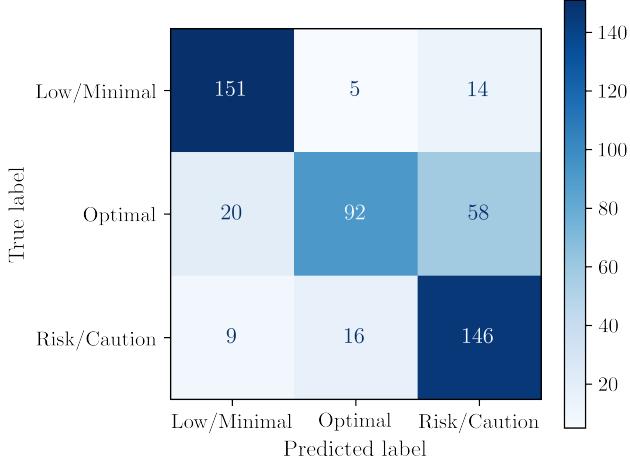


Fig. 7: Histogram of the new classes, once 'Caution' and 'Risk' are combined in one.

TABLE III: Classification Report for Model Evaluation (treino)

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

B. resultados de teste

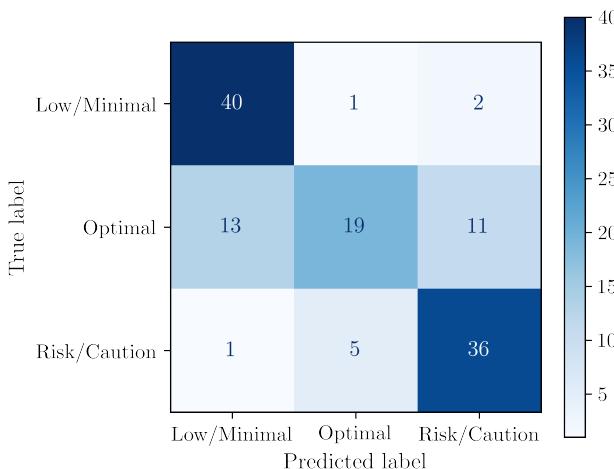


Fig. 8: Histogram of the new classes, once 'Caution' and 'Risk' are combined in one.

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

VII. DECISION TREE MODEL

os paraemtros a otimizar e varias e bla bla Critério de divisão, Profundidade máxima da árvore, Número mínimo de amostras para dividir um nó e Número mínimo de amostras em um nó folha

TABLE V: DTREE Hyperparameter Search Space ?

Parameter	Distribution/Values
Split Criterion	{gini, entropy}
Max Depth	[2, ..., 8]
Min Samples to Split	[5, ..., 20]
Min Samples per Leaf	[3, 4, ..., 10]

QUAL FOI OS HYPERPARAMETES OBTIDOS

BestParameters : 'splitter' : best', 'min_samples_split' : np.int64(11)

otimizada a arvore, foi obtida a seguinte estrutura

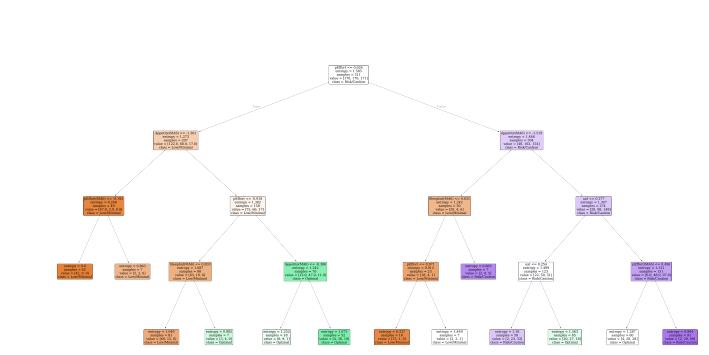


Fig. 9: Histogram of the new classes, once 'Caution' and 'Risk' are combined in one.

verifica-se que foram o depth é de 4, as cores representam a classe (laranja tal, verde tal, ...)

A. resultados de treino

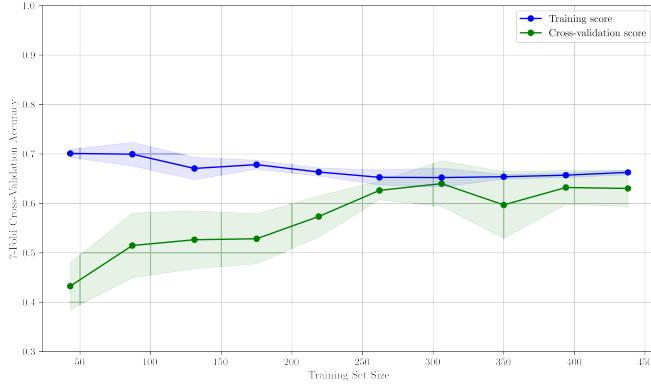


Fig. 10: Histogram of the new classes, once 'Caution' and 'Risk' are combined in one.

B. resultados de teste

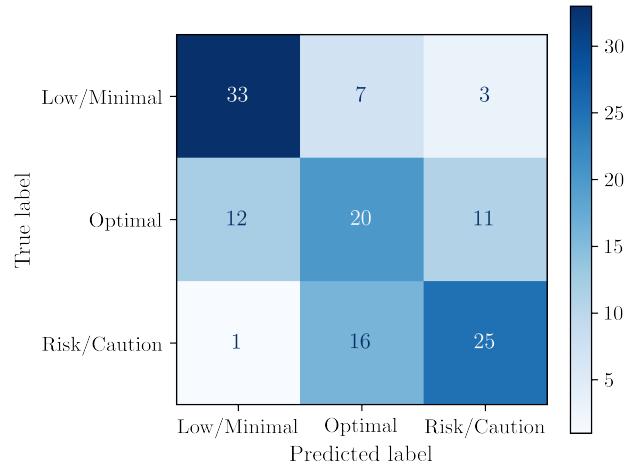


Fig. 12: Histogram of the new classes, once 'Caution' and 'Risk' are combined in one.

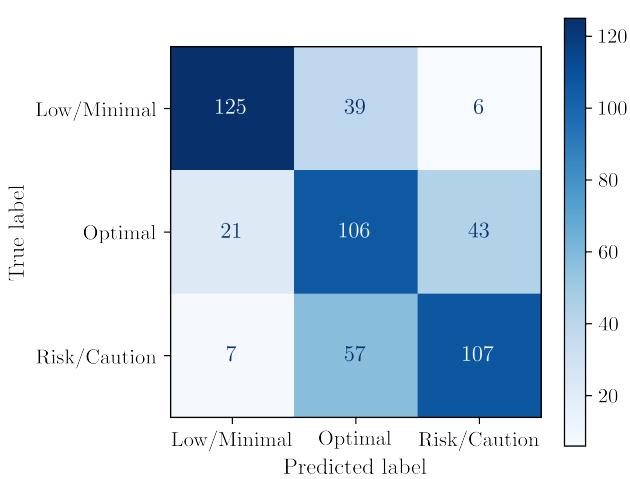


Fig. 11: Histogram of the new classes, once 'Caution' and 'Risk' are combined in one.

TABLE VI: Classification Report for Model Evaluation (treino)

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

TABLE VII: Classification Report for Model Evaluation (teste)

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

VIII. COMPARACAO DOS MODELOS

comparar aqui todos os resultados e isso

acuracaia, learning curves, etc , etc

- learning curves

- scalability

- performance

- cross validation

IX. RESTULADOS

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

X. CONCLUSION

The conclusion goes here.

ACKNOWLEDGMENT

XI. NOTAS

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 comportamento 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 importânciia 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 sao tal tal, que representa tal, tal tal, ... ns q ns q mais.

REFERENCES

REFERENCES

- [1] A. Author, "Title of the paper," Journal Name, vol. 12, no. 3, pp. 45–67, 2024.